[Course Overview](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a)

[Course Overview](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a)

[Hi, everyone. My name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=5.46) [Welcome to my course, Working with Classes and Interfaces in Java. I'm a](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=7.65) [managing partner of JWHH, LLC and have had the good fortune to have been a](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=11.84) [professional software developer for over 35 years.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=15.46) [Hey,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=18.5) [did you know that Java continues to be one of the most](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=18.78) [sought‑after programming language skills? Java is used to](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=21.23) [develop everything from smartphone apps,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=24.14) [to websites,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=26.23) [to server‑side processes, to the many smart device](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=27.25) [features that make up the Internet of Things.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=29.67) [In addition,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=32.31) [a recent survey listed Java as one of the five most important](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=33.16) [skills for data scientists and big data processing.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=35.97) [To work effectively in any of these environments,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=39.16) [you'll need to understand how to work with Java's core object‑oriented features,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=41.41) [classes, and interfaces. And that's exactly what you'll learn in this course.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=44.31) [Some of the major topics we cover include declaring and using classes,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=48.59) [constructors and initializers, class inheritance,](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=52.65) [declaring and implementing interfaces, nested types, and anonymous classes.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=56.02) [By the end of this course, you'll be ready to begin working in any of the](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=60.44) [many environments that rely on Java. Before beginning this course, you](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=63.27) [should already be familiar with the basics of the Java programming](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=67.21) [language. I hope you'll join me as we continue building our Java](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=69.22) [programming skills with the course, Working with Classes and Interfaces in Java, at Pluralsight.](https://app.pluralsight.com/course-player?clipId=fa8e4a3b-4730-440c-8437-eb40ba819f6a&startTime=72.62)

[Understanding Java Classes and Objects](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c)

[Introduction](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c)

[Welcome to the course, Working with Classes and Interfaces in Java.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=0.64) [This is our first module, Understanding Java Classes and Objects.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=4.44) [My name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=8.23) [So now in this module,](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=11.24) [we're going to focus on the fundamentals of working with classes and objects.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=12.5) [So to do that, the first we'll look at is how we](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=16.14) [declare classes within our applications.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=18.87) [We'll then see the different kinds of members a class can](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=21.84) [have and how we add those members to our classes. We'll then](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=23.86) [see how to work with objects,](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=27.25) [which is simply a case of creating instances of classes.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=28.71) [We'll then see how we protect the details of our classes by understanding](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=32.24) [encapsulation and how access modifiers make that possible.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=35.79) [And then we'll finish up by looking at how we protect the data within our](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=39.84) [classes by looking at field accessors and mutators.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=42.66) [Now, just as a quick reminder, this course builds on the Pluralsight course,](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=47.22) [Getting Started with Programming in Java.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=51.35) [So if you haven't really watched that course,](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=53.67) [I strongly encouraged you to go back and watch Getting Started](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=55.4) [with Programming in Java and then return back to this course,](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=57.84) [because we're assuming you already understand information that's covered in that course.](https://app.pluralsight.com/course-player?clipId=6a411b7e-684a-407f-a398-d63c1244f80c&startTime=60.9)

[Declaring Classes](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350)

[Java is an object‑oriented language.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=0.44) [What objects allow us to do is encapsulate data and the](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=3.39) [operations on that data into a single entity,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=6.72) [then expose semantics for working with that entity.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=9.98) [And with objects, the storage and manipulation details can be hidden,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=13.09) [and that's a really important concept because it allows the object to expose the](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=17.85) [operations that can be performed without burdening the user of the object with](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=22.5) [the details of how those operations are performed.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=26.96) [And when used correctly,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=29.86) [this can simplify building complex applications. Because all](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=31.54) [the complexity is packaged up in the object,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=35.17) [the user of the object does not have to get involved in those complexities.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=37.63) [Now before we can have an object,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=43.04) [we first have to have a class because a class is a](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=44.75) [template for creating objects.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=47.9) [So we'll start out by declaring a class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=50.29) [We do that using the class keyword followed by the class name.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=52.44) [So if I want to have a class named Flight,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=56.54) [I'll use the keyword class followed by the word Flight.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=58.74) [When we create class names,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=61.86) [we follow the same basic rules is with variable names.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=63.64) [We use just letters and numbers, but we always capitalize the name of a class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=66.04) [If a class is composed of multiple words, each word within](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=70.94) [that class name is also capitalized.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=73.92) [Now the body of the class is contained within brackets.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=76.54) [So we'll have an opening bracket and a closing bracket,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=79.41) [and all members of the class will be contained within those brackets.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=81.95) [Now when we create our class, we're of course going to put it into a source file,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=86.1) [and that source file name is normally required to](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=89.84) [have the same name as the class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=92.02) [So I have a class named Flight.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=94.44) [It's going going to a file named Flight.java.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=95.9) [Now classes are made up of both state and executable code.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=100.54) [We use members to represent these.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=104.29) [So one kind of member are fields. Fields store the object's state.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=106.42) [So those are the data within the class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=111.81) [So now if we look at our class Flight,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=115.54) [some state it might have is maybe the number of passengers.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=117.74) [So we have a field here named passengers whose type is int.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=120.92) [We might also have state related to the number of seats.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=124.54) [So we have another int field here, this one named seats.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=127.95) [So together, passengers and seats represent the state of this class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=131.74) [Now classes can also have methods.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=137.64) [And as you recall from the previous course, methods are executable code.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=139.71) [Now within a class,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=143.53) [there are executable code that can manipulate the state of the object,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=144.98) [as well as perform operations related to the class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=149.14) [So our Flight class might have a method,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=153.58) [add1Passenger, that's responsible to add one more passenger to the flight.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=155.82) [So here inside of add1Passenger,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=161.24) [the first thing we'll do is check to make sure that we](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=163.33) [haven't already filled the seats.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=165.73) [So as long as the passengers are less than the number of seats,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=167.48) [then we can go ahead and add one more passenger.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=170.32) [So this add1Passenger method was able to view the state of the class, in other](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=173.78) [words it could see its fields, as well as manipulate the state, in other words](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=178.2) [it changed the value of the passengers field.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=182.07) [So now we have one more kind of member, which is a constructor.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=186.14) [Constructors are also executable code,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=189.54) [but they're executable code that automatically runs when an](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=191.99) [object is created. And what we normally use them for is](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=195.26) [setting the object's initial state.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=198.44) [So here in our Flight class, we'll add a constructor.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=201.79) [Notice a constructor looks a lot like a method,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=204.77) [but it doesn't have a return type, and the constructor](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=207.19) [name is the same name as the class.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=209.8) [Now again,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=213.34) [the code within this constructor will run automatically](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=213.71) [when an object is created and, again, will normally set initial state.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=216.56) [So here within our Flight class, we'll start out with an initial state of](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=220.33) [the flight having 150 seats available and no passengers.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=223.82) [Okay,](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=228.24) [so now that we understand how to declare a class, in our next clip, we'll jump into some code, and we'll create a class from scratch.](https://app.pluralsight.com/course-player?clipId=6ef125e2-4ca8-4543-b54d-ffe119097350&startTime=228.46)

[Declaring the MathEquation Class](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c)

[So here we are now with some source code opened up in our Java IDE.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=0.74) [The IDE we're using is IntelliJ by the company JetBrains.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=4.54) [IntelliJ is actually one of the most popular of the Java IDEs.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=8.66) [Now the project I have open here is a simplified version of the app that we](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=13.04) [created in the course Getting Started with Programming in Java,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=16.8) [and this app currently does a fairly simple task.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=20.74) [It actually works on two numeric values that we call a leftVal and a rightVal,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=23.72) [then uses a one‑character code called an opCode to indicate an operation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=28.54) [And each of the opCodes are simply the first letter of the operation name.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=32.47) [So the opCode d means divide, a means add,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=36.74) [s means subtract, and m means multiply.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=40.37) [So we have a leftVal of 100 and a rightVal of 50 with an](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=44.24) [opCode of d. That means divide 100 by 50,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=47.73) [which would give us a result of 2.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=51.08) [Now the way this application is currently built,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=53.74) [it's using a very procedural approach,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=55.91) [meaning that all the data and all the methods are really kind of independent,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=58.28) [and it's up to the application to lace them all together.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=62.41) [So when our application first starts up, we go here into our main method.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=65.89) [Our main method calls the method performCalculations.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=69.64) [PerformCalculations uses a bunch of parallel arrays.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=72.33) [Parallel arrays simply means that each element in each array](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=76.14) [corresponds to the same element in the other array.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=80.25) [So if we look here at our leftVals array,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=83.69) [its first element 100 goes with the first element of the rightVals array,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=85.28) [50.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=89.47) [That goes with the first element of the opCodes array d.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=90.11) [So again, that would divide 100 by 50 and so with a result in the results array.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=94.04) [And the same sort of thing is true for the second,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=99.04) [third, and fourth elements of the array.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=101.43) [Now the work of actually doing the operation is in](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=104.64) [this method here called execute.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=107.2) [Let me scroll down so we can actually see that method.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=109.24) [Now if we look here at the execute method,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=111.88) [notice that it receives three parameters: the opCode,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=114.28) [the leftVal, and the rightVal.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=118.21) [Then it has a switch statement that operates on the opCode,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=119.94) [does the appropriate operation, stores that into its local variable named result,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=122.82) [and returns that back from our execute method.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=127.78) [So what we want to do now is move away from this](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=131.54) [procedural approach to take advantage of classes that](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=133.72) [kind of tie all these pieces together.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=136.64) [Because again, as this application is currently written,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=138.32) [all the data values in the method really have nothing to do with each other](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=141.03) [unless the application specifically ties them together itself.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=145.4) [So the first thing we'll need to do is create a new class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=149.24) [Well the easiest way to do that is to head over here to our project window.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=151.88) [Within a project window, we'll go to our package name.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=155.52) [We'll then right‑click on the package name.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=158.84) [Go up to here where it says new.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=160.91) [Then I'll choose Java Class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=164.24) [So now we need to give our class a name.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=166.94) [Let's name the class MathEquation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=168.81) [To create the class, I'll go ahead and hit Enter.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=172.64) [And once I do that, a couple of things happen.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=174.88) [First, if we look here in the project window,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=178.04) [notice we have a new source file, MathEquation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=180.04) [Then over here in our source code window,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=183.68) [we have our class, MathEquation, stubbed out for us.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=185.52) [So now we can start adding members to the class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=188.84) [So the first thing we'll do is add our fields.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=191.59) [And again, we'll need fields for each of the parts of the equation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=193.73) [For example, we'll need a field for the leftVal.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=197.17) [Then once we have our leftVal, we can add fields for our rightVal,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=201.64) [our opCode, and our result.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=205.24) [So now our MathEquation class has four fields.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=208.34) [We have our leftVal and our rightVal Fields,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=211.03) [which are both double, our opCode field,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=213.38) [which is a char, and then another field that's a double,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=216.03) [which is a result.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=219.18) [So those four fields represent the state of our MathEquation class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=220.54) [So now let's add a method that will execute the equation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=225.54) [We'll call the method execute, and we'll give it a return type of void.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=228.35) [Now notice that our execute method within our](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=232.99) [MathEquation class doesn't take any parameters,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=235.24) [and that's distinctly different from the execute method in our Main class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=238.06) [Now that's because here in our MathEquation class,](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=242.44) [all the values we need are actually part of the class itself.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=244.29) [We have all the values in the opCode. So the execute method will use](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=247.54) [those fields rather than accepting parameters.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=250.86) [So now our execute method will need the same switch statement](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=254.14) [we had over in the execute method for our Main class. So let's](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=256.86) [head back over to our Main class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=259.48) [Let's highlight the switch statement.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=262.57) [We'll go ahead and copy that. We'll head back to our MathEquation](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=265.64) [class, and then we'll paste in the switch statement.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=268.71) [So now the execute method in our MathEquation class has the](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=272.04) [switch statement that we need to perform the appropriate](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=274.79) [operation for each individual opCode.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=277.34) [Now let me scroll up here a bit so we can see more of our MathEquation class.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=280.24) [So now our MathEquation class has everything we need to perform an](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=284.74) [equation. We have both our left and right values along with our](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=287.93) [opCode. We have our execute method that knows how to interpret the](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=291.18) [opCode and perform the operation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=294.69) [We have a result field that will store the result of the equation.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=297.04) [So now we have our class declared.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=301.14) [We're just about ready to start using it. So in our next clip, let's see how we can use a class once it's declared.](https://app.pluralsight.com/course-player?clipId=6f4bb29f-d89f-463b-83b7-70d62d9c671c&startTime=302.84)

[Using Classes](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328)

[Once we have our class declared, we're ready to start using it.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=0.64) [So to use our Flight class,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=4.54) [we'll of course need to start out with a variable of type flight.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=5.69) [Now something that's very important to understand,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=9.74) [declaring a variable of type flight doesn't actually create a flight object.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=11.77) [Instead it creates a variable that can hold what's](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=16.56) [then a reference to a flight object.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=19.16) [To actually create the flight object, we have to use the new keyword.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=21.72) [Now when we use the new keyword create our Flight class,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=25.84) [we can refer to what is doing in a couple of different ways.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=28.7) [We can say it's creating a new flight object, or another way to](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=31.74) [say the same thing is that we can say that we're creating a new](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=35.09) [instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=38.02) [Both those terms mean exactly the same thing.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=40.14) [And what we're doing is kicking off a process that has a few steps in it.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=43.19) [First of all, we allocate out the memory to hold our Flight class.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=47.44) [So this memory will have our passengers field,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=51.54) [as well as our seats field within it.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=53.67) [We'll also run the constructor code,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=55.84) [and our constructor code sets passengers to 0 and seats to 150.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=57.94) [And then, we'll return back a reference.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=62.84) [That reference is then stored in the variable.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=65.27) [So our variable, nycToLv,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=67.95) [rather than directly holding an instance of the Flight class, instead](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=70.98) [holds a reference to an instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=75.18) [Now as I had the code written here,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=78.64) [we're doing all this work across two lines. We could,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=80.38) [of course, do it in a single line.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=82.61) [But the result is exactly the same.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=85.08) [We have our variable, in this case it's slcToSf.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=87.54) [We create our new instance.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=91.24) [We run the constructor code.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=92.84) [And we take the reference of that instance and store it in the variable.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=94.54) [Now because classes are always accessed through references,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=99.14) [they're known as reference types.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=102.83) [And as reference types, there are some important implications on their behavior.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=104.53) [So let's go ahead and create another instance of our Flight class.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=108.94) [So again, we'll have our variable here.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=112.04) [In this case, it's flight1.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=113.36) [We'll create the new instance, run the constructor code,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=115.24) [and then store the reference of that instance in our variable flight1.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=118.16) [We'll go ahead and create a second instance that we'll assign to a variable,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=122.14) [flight2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=124.86) [So again, we have our variable, the newly created instance,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=125.94) [and a reference to that instance stored in flight2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=128.79) [Now if we want to interact with the members of one of](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=132.39) [these instances, we use dot notation.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=134.06) [So it's the variable, dot, and the member name.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=136.38) [So in this case,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=139.64) [we're calling the add1Passenger method using the variable flight2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=140.42) [So what Java will do is it'll look at the reference stored in flight2,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=144.94) [follow that to the object instance, and then call the method on that instance.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=148.27) [So when we call add1Passenger, passengers is incremented from 0 to 1.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=153.94) [So if we now go ahead and print out the value of flight2.passengers,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=159.24) [that of course will print out the value 1.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=162.14) [But now what happens if we make an assignment from one](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=166.24) [of these variables to the other?](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=169.15) [So we're assigning flight1 to flight2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=170.78) [Well, if this was a primitive type, something like a double or an int,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=174.04) [the entire value in flight1 will be copied over to flight2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=177.66) [But these are not primitive types.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=182.14) [These are reference types.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=183.81) [And working with reference types,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=185.67) [the only thing that gets assigned is the reference.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=187.05) [So what that means is our variable flight2 will have its reference changed to](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=190.64) [point to the same object instance that flight1 points to.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=194.99) [So if we print out flight2.passengers now,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=199.44) [we'll print out the value of passengers in that](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=202.75) [instance that is currently referencing, which is now 0.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=204.67) [But now what happens if we start interacting with flight1?](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=208.34) [So we say flight1.add1Passenger.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=211.54) [We're going to follow the reference stored in flight1.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=214.64) [When we call add1Passenger, passengers will increment from 0 to 1.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=217.44) [If we call flight1.add1passenger again,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=221.63) [passengers in that instance will increment from 1 to 2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=224.73) [But now what happens when I print out flight2.passengers in this case?](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=229.24) [I haven't made any changes using flight2 at this point,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=233.03) [but let's see what the system is going to do.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=236.77) [It's going to look at the reference stored in flight2,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=238.71) [follow it out to that object instance, and print out that value,](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=240.5) [and that value is 2.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=244.39) [So you see what's happening here is that reference types allow us to have](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=247.04) [multiple variables that point to the same object instance.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=250.68) [And what that means is changes made through one of those](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=254.92) [variables are still reflected in other variables that](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=256.94) [reference that same object instance.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=260.25) [And that's a really powerful concept when it comes to working with classes.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=262.78) [It's important that we understand it so we don't get unexpected side effects.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=266.84) [All right, so now in our next clip, let's jump back into our Java code, and let's start using our MathEquation class.](https://app.pluralsight.com/course-player?clipId=782f1623-053e-45d8-bcb1-f2b2f4cbc328&startTime=271.64)

[Creating an Array of Classes](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0)

[Here we are back in IntelliJ, and we're looking at our CalcEngine project.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=0.74) [This is the project we we're working on when we added our](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=4.17) [MathEquation class earlier in this module.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=6.72) [And what we want to do now is update the application](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=9) [to use our MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=12.24) [So to do that, we'll head over to our Main class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=14.54) [And here in our Main class is where we have this](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=17.64) [performCalculations method that we looked at earlier.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=19.52) [And as you recall,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=22.59) [this method does its work by using a series of parallel arrays.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=23.64) [And what we want to do is change the application so we no](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=27.97) [longer need these individual parallel arrays.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=31.3) [Instead we'll have just one array,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=33.31) [and that'll be an array of our MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=35.48) [So the first thing we'll need to do is declare an array of type MathEquation,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=39.04) [and let's call it equations.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=42.76) [Now you'll notice that when we declare the array,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=45.17) [it's very much like declaring an array of a primitive type.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=48.03) [We, of course, give the variable a name.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=51.24) [And when we specify its type, we identify the type itself,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=53.64) [which is MathEquation.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=56.88) [But then we indicate that it's an array of putting the](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=58.44) [square brackets after the type name.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=61.04) [So then we'll go and create a new MathEquation array of size 4.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=63.34) [Now notice here when we create this array,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=68.04) [we're creating an array of MathEquation references.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=69.8) [So we're not creating four instances of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=73.84) [Instead, we're creating four references of type MathEquation.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=77.81) [Each of the elements within this array will need to explicitly](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=81.62) [create an instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=85.56) [So we'll start out by setting the zeroth element to a new](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=88.31) [instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=91.87) [So now that zeroth element of our equations array references](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=95.24) [a new instance of our MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=98.88) [So that means we can now set that instance's field values.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=102.04) [Now if we look up here at our parallel arrays,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=105.64) [notice that the first calculation we were doing was a left value of 100,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=107.55) [a right value of 50, and an opCode of d.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=111.86) [So let's set each of those fields on equation subzero to the appropriate values.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=115.84) [So now with that code in place,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=121.64) [each of the fields within our new MathEquation](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=123.27) [instance have the appropriate values.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=125.58) [Now, of course, that just takes care of one member of a four‑element array.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=128.54) [We're going to have to repeat all this work for each of the other elements.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=132.64) [So rather doing all that code explicitly right here,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=136.34) [maybe what we should do instead is use a method to create the](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=139.44) [MathEquation instance and set each of the fields.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=143.22) [So let's go ahead and remove this code here where I set the individual fields,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=145.89) [as well as create the new instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=149.99) [So now with that code removed, I'll call a method we'll create named create.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=154.34) [Then I'll pass on the values we want for each of the fields.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=159.84) [So now that we have a call to the method passing in the appropriate values,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=163.99) [we now need to create the method.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=167.45) [But it turns out I don't have to create the method manually.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=169.44) [You'll notice that when I place my cursor here on the method name that](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=172.34) [IntelliJ places this light bulb off to the left.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=175.79) [That tells me the IntelliJ has some ideas for how it can help.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=178.67) [So to access those options here on Windows, I'll press Alt+Enter.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=182.74) [And you'll notice when I do that, one of the options is to create the method.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=187.34) [So I'll go ahead and choose that option.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=191.14) [You'll notice that it stubbed out this create method.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=193.24) [It indicates it has a return type of MathEquation, which is what I want.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=196.34) [It also provides a list of parameters and has given](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=200.44) [those parameters some default names.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=202.59) [But we'll want to change those parameter names.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=204.74) [So to get over to those parameters, I'll press the Enter key a few times.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=207.14) [For the first parameter,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=210.16) [I'll change its name to leftVal. I'll press Enter to accept](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=211.86) [it and move over to my next parameter. I'll make this one](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=216.31) [rightVal. Using the Enter key,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=218.86) [I'll go over to our last parameter, and I'll make this one opCode.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=221.87) [Then I'll go ahead and accept that parameter.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=226.27) [So now that we've got our method stubbed out, we can start](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=228.44) [adding the code that we actually want.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=230.44) [So what we'll do is create a new instance of our MathEquation class and assign](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=232.49) [it to a local variable named equation of type MathEquation.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=236.83) [So that gives me a new instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=241.54) [So now I can use each of the parameters to set the appropriate fields](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=244.52) [on that new instance using our equation variable.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=247.78) [So now once we've set the fields, I can simply return equation,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=252.04) [which will return back the reference to our newly](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=255.43) [created instance of MathEquation.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=257.7) [So now with that, our create method takes care of the details of creating a](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=261.24) [new instance of our MathEquation class and then returning back a reference](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=265.48) [to that newly created MathEquation instance.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=269.58) [So that means that we can now use its create method to initialize](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=272.64) [each of the elements in our equations array. So let me scroll back](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=275.74) [up to where we set our equations array.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=279.32) [So now we're back up here in our performCalculations method. Ao](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=282.54) [we'll go ahead and use that create method to set elements 1, 2,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=285.54) [and 3 of our equations array.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=288.81) [So now with that,](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=292.14) [we have our equations array set up. Each of the elements in the array are](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=293) [referencing new instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=296.96) [And each of the fields within those instances will have the values](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=300.14) [that we want to use to perform our calculations.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=303.54) [So now we're ready to start using this array. We'll see how to do that in our next clip.](https://app.pluralsight.com/course-player?clipId=cbe925aa-76f3-4131-bd4a-1ede5a465fa0&startTime=306.54)

[Using the MathEquation Class](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a)

[Here we are back in IntelliJ,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=0.64) [and what want to do now is continue the work of migrating our](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=2.13) [application to using our MathEquation class rather than having to](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=5.2) [manage a bunch of individual arrays directly.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=8.55) [So now we already have the code in place that sets up our MathEquation array.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=11.44) [So that means that we can take away all the code related that](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=15.44) [are working with those individual arrays that the application](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=18.73) [had to explicitly manage on its own.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=21.51) [So that means first of all, we can get rid of the arrays themselves,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=23.84) [so I'll remove those.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=27.36) [So now with those arrays gone,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=29.78) [we can also get rid of these loops here at the bottom.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=31.01) [And notice that one of these loops here at the bottom actually calls the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=33.57) [execute method that accepts a number of parameters,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=37.04) [and these parameters were all values that came from those arrays.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=39.52) [So that means first of all, we can go ahead and get rid of these loops.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=43.14) [So now once those loops are gone,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=47.74) [we can also get rid of the execute method that's located here in our Main class](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=49.32) [that relied on receiving each of those values as parameters.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=52.75) [So let me go ahead and scroll down a bit.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=56.14) [So now we're down here in our Main class's execute method.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=58.94) [We don't need this anymore, so let me go ahead and get rid of this.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=62.14) [And now with that method gone,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=66.04) [let's scroll back up to where we're doing the work with our MathEquation class.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=67.32) [So now we're back up here in our performCalculations method. So what](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=72.64) [we want to do now is loop through this equations array and do the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=75.57) [work contained in each of the elements.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=79.2) [So let's start out with a for loop,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=81.94) [and what we'll do is use the variation of the for loop and do a foreach loop.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=84.91) [Remember that the foreach loop can iterate to the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=89.34) [elements in the array automatically.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=91.53) [So to do that, we'll declare a variable named equation of type MathEquation.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=93.94) [Then we'll place a colon in the array we want to iterate through,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=99.56) [which is our equations array.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=102.61) [So what this for loop will do is walk through each element in the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=105.54) [equations array and each pass through the loop the equation variable](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=108.88) [will represent the current element of the array.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=113.23) [So first it'll represent the zeroth element,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=116.54) [than the 1 element, and then all the way through to the end of the array.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=118.78) [So we can simply use that equation variable to perform the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=122.15) [work we want to do on each of those elements.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=125.37) [Well the first thing we'll want to do is execute the work](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=127.94) [that's contained in the MathEquation instance.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=130.42) [So we'll call equation.execute.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=133.34) [Now remember,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=136.74) [the way we wrote the execute method is that we'll look at the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=137.55) [leftVal, the rightVal, and the opCode fields,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=139.9) [do the appropriate work, and then store that result in the result field.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=142.9) [So that means we can simply display equation.result.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=147.49) [And with that,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=152.64) [our application is all set. Rather than the application having to](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=153.32) [manage a bunch of values individually, manage how all the work is](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=156.98) [done and the details of how that work is done, instead the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=160.76) [application simply relies on our MathEquation class. So it sets up an](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=163.62) [array of type MathEquation, it sets all the appropriate values for](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=168.02) [each of those equations,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=172.03) [and then it can simply loop through and say to the MathEquation class,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=173.54) [hey, go do an execute.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=176.48) [You handle the details of how that's done, and just tell](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=177.79) [me what the result is. So what we've done now is we've](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=180.6) [got a good division of labor.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=183.41) [The MathEquation class focuses on how to manage an equation.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=185.26) [The application simply takes advantage of the](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=189.33) [capabilities provided by our MathEquation class.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=191.24) [In addition,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=194.99) [if we need to do any other math equations anywhere else in the application,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=195.95) [we can now leverage that existing MathEquation class.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=199.34) [So now just to confirm that everything works, I'll](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=203.24) [go ahead and run the application.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=205.1) [Once the application runs, we can see the results in our run window.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=207.84) [So if we look up here in our code, the first equation was a d opCode,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=211.78) [which means division. So we divide 100 by 50.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=215.84) [That gives us the result of 2.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=218.81) [Our next equation is an add operation, so we'll add 25 to 92.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=220.82) [That gives us 117. Then we have a subtract operation, so](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=225.59) [we subtract 17 from 225. That gives us 208. And then our](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=229.12) [last operation is a multiply.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=233.82) [So wel multiply 11 times 3, and that gives us 33.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=235.67) [All right, so now in our next clip,](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=239.94) [let's take a look at this idea of encapsulation and how we use something known as access modifiers to achieve it.](https://app.pluralsight.com/course-player?clipId=a814fdf0-e626-4edc-a459-e91a9d5fde3a&startTime=241.8)

[Encapsulation and Access Modifiers](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25)

[Now as we've mentioned.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=0.71) [as we declare our classes, in general,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=1.43) [we went the details of how that class is implemented to be hidden.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=4.16) [This idea of hiding certain details about the class](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=8.04) [is what's known as encapsulation.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=11.17) [So you're encapsulating the details within the class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=13.54) [Now in order to achieve encapsulation, Java gives us access modifiers.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=16.18) [What access modifiers allow us to do is control the](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=21.74) [visibility of classes and their members.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=24.94) [So let's take a look now at three of the basic access modifiers.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=28.84) [Now one option for access modifiers is to not](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=33.14) [specify any access modifier at all.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=35.63) [And in this case,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=38.54) [the item would be visible only in the package where the class is declared.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=39.63) [And for that reason, this is sometimes known as package private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=44.21) [Now classes themselves can be package private,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=48.24) [as well as the members of a class can also be package private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=50.91) [Then we have the public access modifier.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=55.32) [If something is public, that means it's visible everywhere.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=57.92) [And classes can be public and the members of classes can also be public.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=60.99) [And then at the opposite end of the spectrum, we have private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=66.04) [If something is private, it's only visible within the class where it's declared.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=69.44) [Now, in general, classes cannot be private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=74.24) [Now there is an exception to that.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=77.54) [There's something in those nested classes that we'll](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=79.54) [talk about later on in this course, and nested classes can be private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=81.35) [But in general, classes cannot be marked as private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=85.81) [But now the members of a class can be marked as private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=89.44) [So let's again look at our Flight class and see how access](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=94.04) [modifiers affect the uses of that class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=96.9) [So we have our class itself, as well as our two fields,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=99.84) [passengers and seats, we have a constructor,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=102.76) [and we have our add1Passenger method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=105.65) [Now there are a number of things we need to consider](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=108.24) [about the usability of this class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=109.9) [One of the key things we need to consider is where can](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=112.09) [variables of this type be declared?](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=114.57) [So if I mark the class itself as public,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=117.14) [that means I can declare variables of type flight anywhere I want to.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=119.44) [But interestingly,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=123.44) [marking the class as public does not mean I can](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=124.49) [necessarily create instances of the class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=127.28) [If I want to be sure I can create class instances from anywhere,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=130.14) [I need to also mark the constructor as public.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=133.18) [So by marking the constructor as public,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=136.14) [I can declare variables of type flight and create new](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=138.07) [instances of that type flight from anywhere.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=140.95) [So now when it comes to our fields, let's go ahead and mark those as private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=144.24) [That means that there are only accessible from within the class itself.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=147.84) [So if I have code that tries to use that flight reference](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=151.78) [and print out the number of passengers,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=154.33) [I'm actually going to get a compile error because passengers is private,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=156.34) [meaning it's not accessible from outside the class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=160.12) [But now for our add1Passenger method, let's go ahead and mark that as public,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=163.41) [and because it's public, it can be accessed from anywhere.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=167.34) [So what that means is the add1Passenger method gives us a controlled](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=170.44) [way for user's to increment the number of passengers without giving](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=174.36) [them access to our passengers field itself.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=177.9) [So again looking at our Flight class,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=181.88) [let's take a closer look at our add1Passenger method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=183.68) [Now currently the way this method is implemented,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=187.34) [we check to make sure there are still seats available.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=189.77) [As long as there are seats available, we increment the number of passengers by 1.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=191.97) [But notice if there are not seats available, we silently fail.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=196.34) [We don't increment the number of passengers,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=200.55) [and we give no indication that there's no room left.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=202.94) [So we should really improve this method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=205.89) [Now this idea of trying to add passengers and there being no room](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=208.46) [left is something that might come up fairly often.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=212.14) [So let's go ahead and add a method that will handle when](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=215.11) [we try to add too many passengers.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=217.54) [And for simplicity,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=219.25) [we'll just have it print out the fact that there are too many passengers.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=220.44) [Now notice that our handleTooMany method is private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=224.24) [And that means that we don't want anyone from outside](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=227.94) [the class to access this method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=230.12) [But of course, methods within this class can access it.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=232.64) [So in our add1Passenger method, if there aren't any seats left,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=235.84) [then in that case we'll go ahead and call our handleTooMany method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=239.06) [So now let's look at how this affects the usability of our class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=243.74) [So we have our Flight class, which is public.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=247.04) [It has our two private fields.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=248.8) [It has a public instructor and our public add1Passenger method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=250.2) [And now it also has a private handleTooMany method.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=254.72) [So we can, of course, still use the class itself.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=258.02) [But if we try to call handleTooMany,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=261.34) [we're actually going to get an error because handleTooMany is private.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=263.46) [It's not accessible from outside the class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=266.8) [But now what about add1Passenger?](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=269.84) [Well add1Passenger is public, so we can still call add1Passenger.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=272.14) [The fact that add1Passenger calls a private method does not](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=276.6) [impact the accessibility of add1Passenger itself.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=280.81) [And this is a key part of how we implement our classes.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=284.64) [We have a finite number of members that we make public and that](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=287.74) [controls the interaction points with our class.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=290.81) [And then we deal with the details of our work using private members,](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=293.37) [which hides those members away from being used outside the class itself.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=296.9) [All right, so now in our next clip, let's take a look at some of the special references provided by Java.](https://app.pluralsight.com/course-player?clipId=78c7c07f-f23e-46d2-8583-2ea2ae1d0d25&startTime=301.94)

[Special References: this and null](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0)

[So let's take a look now at some of the special references that Java provides.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=0.64) [Now one of the special references is what we call the this reference.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=4.41) [The this reference is a reference to the current object instance.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=8.59) [One of the things we often use this for is reducing ambiguity,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=13.34) [making it clear that we're referring to a member of the current object instance.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=17.08) [Also,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=21.64) [this is useful for allowing an object to pass a reference](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=22.12) [to itself as a parameter to a method.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=25.64) [Then we have the special reference null.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=28.9) [Null represents an uncreated object.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=30.95) [So it allows us to have a reference that,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=33.94) [rather than referring to an object instance,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=36.27) [doesn't refer to any instance at all.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=38.5) [And null can be assigned to any reference variable.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=40.84) [So let's first take a closer look at this.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=45.64) [So we again have our Flight class.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=48.01) [Remember, our Flight class has the two fields, passengers and seats.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=50.29) [Now let's say we want to add a method here, hasRoom.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=54.42) [What hasRoom does is return back true or false,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=57.54) [indicating whether there's room on this flight to combine](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=60.56) [the passengers from another flight.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=63.71) [So the first thing hasRoom does is figure out the total](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=66.24) [number of passengers between the two flights,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=68.9) [then returns back a true or false value,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=71.3) [indicating whether there are enough seats to hold all those passengers.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=73.47) [Now let's look here where we calculate the total.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=77.94) [Our passengers field actually shows up twice.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=80.12) [We first have f2.passengers.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=83.84) [So this refers to the number of passengers on the flight](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=86.64) [that was passed in as a parameter.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=89.52) [But our other use of the passenger field is unqualified.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=91.84) [And in this scenario,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=95.34) [the passengers field not being qualified actually](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=96.8) [refers to the current object instance.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=99.17) [In other words, this code is doing exactly we want it to do.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=102.14) [But there may be scenarios where we want to make that fact more clear.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=105.63) [So what we can do is, rather than just saying passengers here,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=108.78) [we can say this.passengers,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=112.43) [making it clear that we're referring to the current object instance.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=114.92) [Now in this scenario, the use of this is not required,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=118.84) [although it's helpful for reducing ambiguity.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=121.96) [But as we'll see a little bit later in this module,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=124.84) [there are scenarios where the this reference is required in](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=126.86) [order for the code to do what we want it to do.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=129.54) [So now let's take a closer look at null.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=133.54) [So let's say we have some code that creates two instances of the Flight](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=135.84) [class and assigns them to the variables lax1 and lax2.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=139.06) [Then we'll have some logic that goes off and adds some number](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=143.74) [of passengers to each of those flights.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=146.58) [And then we'll have another variable here, lax3.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=148.98) [But we don't want lax3 to be initially set to an instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=152.04) [So what we can do is actually initialize it to the value null.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=156.94) [So that means that lax3 doesn't actually refer to](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=161.19) [an instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=164.82) [It doesn't actually refer to any instance of any class at all.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=165.91) [Lax3 is simply null.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=169.44) [So let's add some code here that determines whether there's room to](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=172.24) [combine the passengers that are on lax1 and lax2.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=175.87) [So it uses our hasRoom method.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=179.74) [And if there is, room we'll call some method called createNewWithBoth.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=181.93) [And what we'll say this method does is it creates a brand new instance of the](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=186.7) [Flight class and combines the passengers from lax1 and lax2 and returns back a](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=190.2) [reference so that new instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=196.09) [So if there was room lax3 now references that new instance of the Flight](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=199.44) [class. So then we'll have our code go off and do some other work. When](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=204.7) [they work completes, we need to determine whether lax3 was set to an](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=208.66) [instance the Flight class or not.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=213.45) [Well what we can do is check to see if lax3 is not equal to null.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=215.14) [And if lax3 is not equal to null,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=219.84) [that tells us that lax3 now references an the instance of the Flight class,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=222.15) [which means those flights must have been combined.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=227.84) [So as you can see,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=231.14) [null is useful for initializing variables to refer to no object instance at all,](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=231.9) [and we can use it to test whether a particular variable refers to an object](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=236.83) [instance or not. All right, so now in our next clip, let's take a look at field encapsulation.](https://app.pluralsight.com/course-player?clipId=3c4673e6-0a26-4829-97b6-1296be0623e0&startTime=240.83)

[Field Accessors and Mutators](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a)

[As we've talked about, as we design our classes,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=0.71) [we want to keep the details of how that class is](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=3.5) [implemented hidden from the users of the class,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=5.84) [and our fields are generally considered to be an](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=8.98) [implementation detail because the fields are details of how](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=11.72) [we actually manage the state of our classes.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=15.71) [So in most cases,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=19.34) [we do not want our fields to be directly accessible from outside the class.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=20.29) [So instead of allowing users of our class to directly access to fields,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=25.24) [we instead want to control access to the fields through methods.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=28.45) [This will allow us to evolve the implementation of our class and](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=32.34) [possibly even change the way we store the state within our class](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=35.38) [without breaking any code that uses our class.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=38.42) [Now a common pattern we use for protecting our fields is](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=42.84) [what's called the accessor/mutator pattern.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=46.26) [And in this pattern, we use a pair of methods to control access to the fields.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=49.08) [So we have one method, which is known as the accessor.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=53.44) [The accessor method retrieves the value of the field.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=56.25) [This is also commonly known as a getter method.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=59.79) [When we create the method, we'll normally name it get followed by the field name.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=62.94) [And then we also have a mutator method.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=66.94) [The mutator method modifies the field value.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=69.69) [This is also commonly known as a setter.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=72.94) [When we create these methods, we'll normally name it set,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=75.24) [followed by the field name.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=78.09) [So again, looking at our Flight class, let's focus in on our seats field.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=80.74) [So now we've marked this field as private because we don't want](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=85.74) [users of the class directly accessing the field.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=88.15) [So we'll use the methods to control interaction with it.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=91.24) [So we'll first have our getter or what's called our accessor.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=94.1) [So since our field is named seats,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=97.34) [we've named the method getSeats. Notice the method's return](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=99.03) [type is the same type as the field itself,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=102.37) [and all we're going to do in our getter here is return back the value of seats.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=104.74) [Now we'll also need our mutator or what we call our setter.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=109.34) [So we've named this method setSeats.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=112.24) [Its return type is void, meaning it doesn't return any value.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=114.74) [But notice is accepts a value as a parameter.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=117.9) [In this case, we've named that parameter seats.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=120.38) [So what we want to do is have the value that's passed in to](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=123.14) [setSeats assigned to our seats field.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=125.59) [But now it turns out,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=129.24) [as this code is currently written, there's a bit of a problem.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=130.11) [Now looking here at seats on the right side of the equal sign,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=132.39) [this refers to our seats parameter, and that's what we want.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=136.26) [We want to take the value that's passed into our setSeats](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=140.2) [method and then assign it to our seats field.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=142.31) [The problem is on the left‑hand side of the equals sign,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=145.34) [as this code is written, that seats also refers to the parameter,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=147.71) [and that's not what we want.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=152.24) [We want to assign the value to the seats field itself. And the](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=154.04) [issue here is that when a parameter name within a method has the](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=157.77) [same name as a field within a class,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=162.06) [the compiler assumes that we want to use the parameter.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=164.84) [So we need to be more explicit about the fact we want to use the field,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=168.33) [and we know how to take care of that.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=171.84) [Remember we have our this reference. Our this reference allows us to refer to](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=173.34) [something within the current class instance. So by using this,](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=177.47) [we can now take the value that's passed in as a parameter](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=181.81) [and assign that value to our seats field.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=184.69) [So to use our methods, it's really straightforward.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=188.64) [We'll go ahead and create an instance of our class.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=191.64) [We want to set its value. We'll simply call setSeats, passing in the value.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=193.9) [So this will set this flight to have 200 seats.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=198.2) [If we want to get the value back, we call our get](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=201.29) [method, which will return back the value, which will display 200.](https://app.pluralsight.com/course-player?clipId=e0dc55af-4085-40f1-9a4e-87e0115ac62a&startTime=203.31)

[Summary](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a)

[To wrap up,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=0.74) [here are some of the key things you want to remember from this module.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=1.51) [Remember we started out by looking at classes.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=4.01) [We said that a class is a template for creating objects.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=6.53) [Then an object is simply an instance of a class.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=10.36) [So it's a class that describes the type that we're creating.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=14.64) [The object is what we get when we actually create an instance of that type.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=18.08) [Now remember that classes are reference types. That means that when](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=22.32) [you declare a variable whose type is a class,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=25.75) [it doesn't actually create an instance of the class.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=28.72) [The variable can simply hold a reference to the class.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=31.14) [To actually create an instance of the class, we used the new keyword.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=34.01) [So when we call new, we create that class instance,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=38.14) [and then we assign that reference into a variable.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=41.14) [Now remember that multiple variables can reference the same instance,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=44.27) [and this ability for multiple variables to reference the same](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=47.94) [instance of an object is really powerful,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=50.99) [but it's also really important that we understand that's the case](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=53.4) [so we can work effectively with our classes.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=56.16) [Now as we saw, classes have three types of members.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=58.91) [Classes have fields, and fields store the object state.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=63.14) [Then we saw that classes have methods, and methods are executable code,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=67.4) [and remember that class methods have access to the class's state,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=71.04) [so they can actually modify and manipulate the state within the class.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=75.34) [We can also use methods to simply perform operations](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=78.99) [that are related to that class itself.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=81.84) [And then remember that classes can also have constructors,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=84.24) [and constructors are executable code that is executable code that runs](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=86.86) [automatically during the creation of a class instance.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=90.89) [And generally what we use constructors for is to set that class's initial state.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=94.74) [Then we saw that classes have some special references.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=99.94) [We have the this reference,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=102.52) [which allows us to refer to the current object instance.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=104.03) [And then we have the null reference,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=107.43) [and the null reference represented an uncreated object.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=109.56) [As we saw with null,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=112.59) [we can use it to initialize a variable to indicate that it doesn't](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=114.23) [reference any object instance, and we can also use null to test to](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=117.48) [see whether a variable refers to an object instance. And then we](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=121.77) [looked at the access modifiers.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=126.46) [What access modifiers do is control visibility. They can control class](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=128.44) [visibility. They can also control class member visibility. And access](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=132.75) [modifiers enable us to have encapsulation.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=137.62) [So with access modifiers,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=140.64) [we can have portions of our class that are accessible from](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=142.05) [outside the class and other parts of our class that are not](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=144.52) [accessible from outside the class.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=147.82) [And as we saw, we normally don't want our fields to be directly accessible.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=150.34) [So instead, we use methods to provide access to our field values.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=154.77) [So we have the accessor methods, what we often call the getter methods.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=159.44) [We use these to retrieve the field values.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=162.79) [Then we also have mutator methods, what we sometimes call setter](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=165.84) [methods, that we use to modify the field values.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=168.63) [All right,](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=172.74) [that wraps up this module. In our next module, we're going](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=173.43) [to take a closer look at constructors. We're also going to look at something known as an initializer.](https://app.pluralsight.com/course-player?clipId=9c061978-ce31-4488-beb8-5fdb5d0c9c0a&startTime=176.52)

[Implementing Class Constructors and Initializers](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0)

[Introduction](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0)

[Welcome to our next module, Implementing Class Constructors and Initializers.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=0.64) [This is part of the Pluralsight course, Working with](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=4.69) [Classes and Interfaces in Java, and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=6.49) [Throughout this module we'll be looking at how we can control the](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=11.34) [initial state of a class when it's first created.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=14.56) [Now, in order to do that, the first thing we need to understand is,](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=17.44) [what is the default initial state of the fields within a class](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=20.02) [when that class this first created? From there, we'll look at](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=23.15) [some of those field initializers,](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=26.32) [which allows us to assign our own values to a field.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=27.83) [We'll then look at class constructors. Now we touched on](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=30.94) [constructors quickly in the previous module,](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=33.41) [and in this module we'll going to look at it much more closely.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=36) [And one of the interesting things about constructors, is a class can](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=38.94) [have multiple constructors and even has the option of chaining](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=41.54) [constructors together. So that when one constructor is used, that](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=44.91) [constructor automatically calls another constructor and runs the code](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=48.11) [in that constructor as well.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=51.27) [We'll take a look at constructor visibility.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=53.09) [We'll talk about why,](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=54.89) [in certain cases, you may want to limit the visibility of](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=55.75) [some of the constructors within your class.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=58.67) [We'll then look at initialization blocks.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=60.94) [Initialization blocks allow us to have a code that](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=62.79) [automatically runs no matter which constructor is used.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=64.81) [And then, finally,](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=68.34) [with so many options for how we establish the initial state of a class,](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=69.28) [we want to be sure that we understand how those different](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=73.44) [options relate to one another, and in what order each of the options are performed.](https://app.pluralsight.com/course-player?clipId=82575952-9acc-42c8-8230-d12e077457e0&startTime=75.68)

[Class Initial State](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a)

[Whenever we create a new instance of a class,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=0.64) [that newly created object is expected to be in some useful state.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=3.23) [So to help us with that,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=7.24) [Java will give you to the fields within the class the default value.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=8.17) [But these default values are often not enough.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=11.94) [In many cases,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=14.45) [we need to take specific action to properly establish the](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=15.64) [initial state of the fields within our class.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=19.23) [Now there's a couple ways we can do that.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=21.74) [One option is to simply set the field values directly.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=23.64) [Another option is to have code that executes automatically](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=26.94) [as part of creating the class instance.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=30.06) [And as you recall, the state of a class is represented by its fields.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=33.44) [So when a new class instance is created,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=36.94) [each of those fields will be given a default value.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=38.85) [Now the specific value will depend on the type of the field itself,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=42.19) [but in general, each field is set to whatever its zero value is.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=46.34) [Now in the case of numeric field types, things like byte and short,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=51.24) [int and long, as well as float and double,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=54.54) [they are literally set to zero.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=57.08) [But remember that all types are not numeric.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=59.74) [For example, we have the char type,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=62) [where the char type contains a character value.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=63.99) [So the default value for any field that's a char type is the](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=67.58) [character that's represented by all zeros.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=70.65) [Then for our boolean fields, boolean fields default to a value of false.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=74.21) [But oftentimes our classes are going to continue with a](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=79.14) [number of reference type fields as well.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=81.57) [In the case of reference types, each field is set to a null value by default.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=84.24) [Which of course means that field is not set to](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=89.44) [reference an actual object instance.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=92.38) [Now, as we've mentioned,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=95.94) [we're not limited to accepting the default state that Java provides.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=97.1) [We can actually establish our own starting state of our classes.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=101.08) [And in general there are three ways to do that.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=104.63) [One way is to use what's known as a field initializer,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=107.34) [and a field initializer allows us to give a field a value](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=110.48) [as part of the field declaration itself.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=113.26) [Now, in addition to that,](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=115.69) [we have constructors and we also have initialization blocks. And both](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=116.95) [of these give us the option of having code that runs automatically as](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=120.6) [part of the object instance creation.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=124.16) [Now, throughout this module, we'll look at all these options in more detail. So in our next clip, let's start out with a closer look at field initializers.](https://app.pluralsight.com/course-player?clipId=37141630-5479-4135-87cf-0259446e531a&startTime=126.28)

[Field Initializers](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19)

[Now oftentimes, we want our fields to have a value other than the](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=0.64) [default value, and that's where field initializers come in, because](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=4.13) [field initializers allow us to specify a field's initial value as part](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=7.77) [of the field's declaration itself.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=11.88) [So for example,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=14.14) [I have a class here named Earth, and I want to have a field](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=14.84) [circumferenceInMiles. Now at this point, circumferenceInMiles has](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=18.71) [a default value of 0, and I'd like to set it to the appropriate](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=22.92) [value for circumferenceInMiles.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=26.75) [And I can do that right here where I declare the field by](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=28.71) [simply assigning the value to the field.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=31.02) [So now when a new instance of our class is created,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=33.94) [circumferenceInMiles will have that value of 24,901.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=36.09) [But now when we set our field values, we're not limited to simple assignments.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=41.14) [You can also include equations.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=45.54) [So if I want to have a field here circumferenceInKms, well,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=47.84) [I know I can convert miles to kilometers by multiplying by 1.6.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=51.24) [So here I'm taking the literal value of 24,901 and multiplying that times 1.6.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=55.84) [Now although that'll give me the correct answer, it's](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=62.64) [not really clear what I'm doing here.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=64.99) [So it would be nice if I could reference that other](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=67.04) [field, and I can indeed do that.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=68.97) [When I set the value of one field,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=71.74) [I can use other fields in determining its value.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=73.31) [So here where we set circumferenceInKms, rather using that 24,901](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=76.34) [literal, we'll use our field circumferenceInMiles.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=80.8) [Now when we do this calculation and multiply circumferenceInMiles by 1.6,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=84.74) [the result will be a double, which will include a fractional portion.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=89.24) [But now currently, we're simply casting it to a long,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=93.14) [which would drop any fractional portion.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=95.89) [It would be nice if we could round the result.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=98.64) [Well, it turns out we can do that because we're also allowed to](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=101.1) [include method calls when we make the assignment.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=103.73) [So here when I set my circumferenceInKms, rather than doing the cast](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=106.84) [to long, I'm going to call the Math function round, which will then](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=110.3) [round the results of that equation.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=114.03) [So as you can see,](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=116.44) [field initializers give us an easy way to set the initial values of our fields.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=117.24) [But in some cases they're not enough.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=122.14) [In some cases, we need to actually execute code. So in our next clip, let's take a look at constructors.](https://app.pluralsight.com/course-player?clipId=6502613f-77ac-463d-8879-882eab759e19&startTime=123.96)

[Constructors](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1)

[Now as you recall, we talked about constructors briefly in our previous module.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=0.69) [A constructor contains code that runs whenever we](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=5.09) [create a new instance of our class.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=8.19) [Now remember the constructors have to have the same name as the class](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=10.53) [that contains them, and constructors don't have a return type because](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=13.64) [a constructor doesn't return a value.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=17.33) [It simply contains code that runs when we create a new instance of our class.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=19.56) [So let's take a look at our Flight class that we worked on in a previous](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=24.97) [module where the Flight class contains two fields,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=27.96) [passengers and seats, and we also gave it a constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=31.69) [Now the constructor has set values for those two fields.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=35.69) [But now we've actually learned a lot since we originally wrote this code.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=39.14) [For example, here we set passengers equal to 0.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=42.74) [We know that that code is not necessary because an int](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=46.24) [field automatically has its value set to 0.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=49.13) [And then here we set our seats to 150,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=52.48) [but we don't need that code either because we know a field](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=54.6) [initializer will allow us to assign the value to the field as](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=57.58) [part of the field declaration itself.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=61.21) [So it turns out that we didn't really need to do any of the work](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=63.34) [in that constructor that we had been doing.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=66.29) [Now it turns out that each class does need to have at least one constructor,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=69.74) [but we don't have to explicitly write that constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=74.54) [If we don't provide a constructor of our own, Java will provide one for us.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=77.11) [The one Java provides will simply be an empty constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=81.46) [So let's take a look at another class here.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=85.82) [Let's say we have our passenger class,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=87.34) [and the passenger class represents a passenger on a flight.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=89.15) [Now our passenger class will have a checkedBags field to keep track](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=93.04) [of the number of bags that passenger has checked.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=96.43) [And then we'll also include a freeBags field because some passengers](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=98.09) [will be allowed to bring some bags with them for free.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=102.13) [We'll assume we have getters and setters for both of those fields.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=104.99) [Then we'll also another field, our perBagFee field,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=107.67) [which will indicate the cost for each bag the passenger](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=111.38) [brings beyond the number of freeBags.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=113.51) [Now as our passenger class is currently written,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=116.49) [it does not explicitly contain a constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=118.53) [But we can still create instances of our passenger class,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=120.84) [and the reason that we can do that is that Java will](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=124.62) [automatically inject a constructor for us.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=127.93) [Again, that constructor won't do anything,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=130.99) [but it does fulfill that requirement of having at least one constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=132.61) [And then, of course, once our passenger instance is created,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=137.14) [we're free to interact with that instance so we can do](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=139.6) [things like set the number of checked bags.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=142.02) [Now although a class has to have at least one constructor,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=145.54) [it's not limited to having just a single constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=148.13) [It can have as many constructors as it needs.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=150.82) [The thing is is each constructor has to have a unique parameter list.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=153.84) [So that means the constructors either need to differ in the](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=157.84) [number of parameters they contain or they need to differ in](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=159.95) [the data types of those parameters.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=163.05) [Now this information about the number of parameters and their](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=165.24) [data types is often what we call the signature.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=167.66) [So we commonly will say that you need to have](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=170.84) [constructors with differing signatures.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=172.83) [And when we say that,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=175.49) [all we mean is that each constructor has to have its own unique parameter list.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=176.16) [So now back here in our passenger class.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=181.74) [Now again,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=183.4) [our passenger class is relying on that constructor that](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=184.22) [Java has provided for us automatically.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=186.75) [So now let's say we want to add another constructor that allows us to](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=189.24) [specify the number of freeBags this passenger is allowed to have as](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=192.02) [part of the object instance creation.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=195.53) [And inside this constructor,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=197.39) [we're simply going to set our freeBags field to contain the](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=199.2) [value we pass in to this constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=201.82) [So now with this constructor in place, when we create a new passenger,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=204.24) [we can pass in a value.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=207.43) [So in this case, we're passing in the number 2.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=209.44) [So when we create this new instance of the passenger class,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=212.04) [the freeBags field in that instance will be set to 2.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=214.36) [Now once we create this passenger instance,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=217.94) [we can interact with it like any other instance of our passenger class.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=219.83) [But adding this constructor will actually break our existing code.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=223.17) [We'd no longer be able to create an instance of our passenger](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=226.94) [class, passing in no parameters at all.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=229.16) [And that's because the code that now fails relies on this](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=232) [constructor that Java generated for us automatically.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=234.98) [Java only automatically creates a constructor if your](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=238.44) [class contains no constructors at all.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=241.19) [Once we provide one constructor,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=244.04) [we're responsible to provide all of the constructors.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=245.84) [Now keep in mind we're not required to replace that constructor](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=249.14) [that Java was generating for us automatically.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=252.77) [If we want to require that whenever a passenger class instance is created,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=255.6) [we have to pass in the number of freeBags, we're allowed to do that.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=259.24) [We can keep just this one constructor we have here. But if we still would allow](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=262.88) [instances of our class to be created that pass in no arguments at all, then](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=266.82) [we're responsible to add that constructor as well.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=270.76) [So now we've explicitly written a constructor that doesn't accept any arguments.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=274.34) [Now the constructor happens to be empty.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=278.34) [It doesn't have to be, but it's allowed to be.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=279.93) [But that would then allow the creation of our passenger](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=282.64) [class without passing any arguments.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=284.8) [And just as a matter of terminology,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=287.44) [when we have a constructor like this that doesn't accept any](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=289.64) [arguments, we call that constructor the default constructor.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=291.95) [Okay, so now in our next clip,](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=295.82) [let's continue our discussion of constructors, and we'll take a look at how we can chain constructors together.](https://app.pluralsight.com/course-player?clipId=fa969546-6a31-48ad-8e53-7f87fe8771f1&startTime=298.13)

[Constructor Chaining](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e)

[With classes able to have multiple constructors,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=0.64) [it makes sense that one constructor might want to leverage the](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=2.85) [code that's contained in another constructor,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=5.91) [and we can do that.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=8.22) [One constructor is able to call another constructor.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=9.7) [And oftentimes we call this process of one constructor calling](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=13.42) [another chaining the constructors together.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=15.94) [Now when one constructor calls another constructor,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=18.88) [the call to the other constructor must be the first line of the new constructor.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=20.78) [And the way we make that call is by using our this](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=25.82) [keyword followed by a parameter list,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=28.09) [which means that one constructor can actually pass in](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=30.73) [values to the other constructor.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=32.84) [So let's again look at our passenger class.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=36.34) [As you recall,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=39.04) [one of the constructors in our passenger class accepts a value for freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=39.67) [And in that constructor, we set our freeBags field.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=43.68) [So let's add another constructor here that accepts two integer values,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=46.54) [one for freeBags and one for checkedBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=50.13) [So now inside this constructor,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=53.34) [we're going to want to set the appropriate fields.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=54.7) [So we'll set our freeBags field, and we'll set our checkedBags field.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=56.74) [Now although this code is perfectly valid,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=60.54) [the first line here actually duplicates the code that we already have in](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=62.41) [the constructor that accepts only the value for freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=66.6) [So rather than duplicating that code, we can simply call that other constructor.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=70.24) [So now what'll happen if the user creates an instance of the passenger class,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=75.04) [passing in two integer values,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=78.75) [those values will be received by the constructor as](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=81.08) [parameters for freeBags and checkedBags,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=83.05) [the first thing this constructor does is call the](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=85.64) [constructor that accepts the value for freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=88.01) [We pass in the value for freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=90.3) [We run the code inside that constructor, which then sets our freeBags field.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=93.14) [And then when that code completes, we return it back to the original constructor,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=97.57) [running the code inside there to set our checkedBags field.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=100.9) [So by chaining the constructors together,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=104.59) [it allows each constructor to focus on doing a particular job well.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=106.55) [Also,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=111.04) [this idea of keeping the code centralized in each individual](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=111.59) [constructor makes it easier to evolve our code over time.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=114.19) [So again, looking at our passenger class,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=118.64) [let's look just at the constructors that we have here.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=120.3) [So we have one constructor again that accepts a number of freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=122.94) [The other constructor accepts freeBags and checkedBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=126.23) [And let's add one more constructor that accepts a value for the perBagFee.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=128.72) [And the perBagFee is the amount of money the customer has](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=134.34) [to pay for any bags they check beyond the number of free](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=137.16) [bags that they're allowed.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=140.52) [So let's say as part of designing our passenger class,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=142.34) [we're given business rules to tell us that the perBagFee is](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=144.58) [dependent upon the status of the passenger.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=148.26) [So elite passengers pay a lower bag fee than standard passengers do.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=151.54) [And a passenger's status is indicated by the number](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=156.38) [of free bags that they're allowed.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=159.03) [So what we can do is go up to our constructor that](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=160.91) [receives a value just for freeBags,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=162.66) [and let's add that appropriate business logic there.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=164.78) [So now, inside this constructor,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=167.54) [we have code that checks the number of free bags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=168.81) [And a passenger is considered an elite passenger if](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=171.94) [they're allowed more than one free bag.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=174.34) [So those customers pay a discounted bag fee.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=176.74) [So if a customer is allowed more than one free bag,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=179.64) [their perBagFee is only $25,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=182.04) [but all other customers have to pay a perBagFee of $50.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=185.64) [So what our code here will do is run that logic to determine the](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=189.54) [appropriate perBagFee and then pass that value into the](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=192.61) [constructor that accepts the perBagFee.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=195.64) [So now if we create a new instance of our passenger class,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=198.74) [passing in two integer values,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=201.09) [we'll again call the constructor that receives](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=202.99) [parameters for freeBags and checkedBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=204.82) [We then call the constructor that accepts just the value for freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=207.14) [We pass in the value for freeBags.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=210.35) [That constructor then does the business logic to determine the perBagFee,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=213.28) [then calls the constructor that receives a value for perBagFee.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=217.97) [Then inside that constructor, we set our field for perBagFee.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=220.7) [When that constructor completes,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=224.64) [we run the code inside the constructor that received the value](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=226.38) [for freeBags that sets our freeBags field.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=228.58) [And when that code completes,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=231.6) [we go back to our original constructor and set our checkedBags field.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=232.66) [So as you can see, by chaining the constructors together,](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=236.94) [it allows us to easily evolve the code related to creating the instance of](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=239.74) [our passenger class because each constructor was focused on a certain part](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=243.3) [of the work that had to be done based on the information we had about the](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=246.91) [passenger instance we were creating.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=249.8) [All right, so now in our next clip, let's take a look at the issue of constructor visibility.](https://app.pluralsight.com/course-player?clipId=c965ddd0-260a-44fc-8c2e-0e1173f9d30e&startTime=251.7)

[Constructor Visibility](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606)

[As we add constructors to our classes,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=0.69) [we should be thinking about what visibility each constructor should have,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=3.07) [because some constructors should probably not be public.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=6.34) [And the reason we make some constructors nonpublic is because we want to](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=10.04) [limit what code can have access to certain phases of the instance creation](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=13.34) [process. So let's again look at our Passenger class, and let's look at the](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=18.22) [list of constructors that we now have.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=22.77) [We have a constructor that accepts no parameters, we call that our default](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=24.74) [constructor, we have a constructor that just accepts a value for freeBags, a](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=28.17) [constructor that accepts a value for freeBags and checkedBags, and a](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=32.63) [constructor that accepts a value for the perBagFee.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=35.76) [Now remember that our perBagFee is meant to be part of the business logic](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=38.78) [for our Passenger class, so we probably don't want code that we don't](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=42.47) [control being able to use this constructor, because by having this](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=45.98) [constructor as public, we run the risk of code that we don't control,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=49.47) [creating an instance of a passenger, we'll call him cheapJoe, and this code](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=52.5) [determines that the perBagFee should just be 1 penny. And it's unlikely](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=56.32) [that that's consistent with the business logic we want to use with our](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=61.03) [Passenger class.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=63.81) [So we really want to restrict what code can](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=65.19) [determine the perBagFee for a passenger.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=67.49) [So rather than making this constructor public,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=70.94) [let's go ahead and make it private.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=73.24) [So now by making that constructor private, only code](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=74.96) [within the Passenger class can access it.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=77.97) [So the attempt to create an instance of passenger from outside the](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=80.74) [Passenger class using this constructor will no longer compile because that](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=83.88) [constructor is not accessible outside of the class.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=87.93) [But now the constructors within the class can still access it, so we](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=91.24) [want to create another passenger instance here,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=95.13) [geetha, and we pass in a single integer value,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=97.04) [which means we'll create this passenger instance with a](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=100.01) [constructor that accepts the value for freeBags.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=102.55) [Remember that this constructor has the logic that determines the](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=104.92) [perBagFee based on the number of freeBags,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=107.85) [and this constructor can still chain to that](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=110.48) [constructor that accepts the perBagFee.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=112.34) [Now, let's look at another use case. We'll create another Passenger here,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=115.14) [santiago, and this uses a constructor that accepts two integer values,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=118.19) [which means we're passing in a value for freeBags and a value for checkedBags.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=122.74) [This constructor calls the constructor that accepts](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=126.67) [the value for freeBags, and again,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=128.69) [this constructor has the logic for determining the perBagFee and](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=130.39) [calls the constructor that receives the perBagFee.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=133.89) [So as you can see, making certain constructors nonpublic limits what code](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=137.14) [has access to specific parts of our instance creation process. All right,](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=141.75) [so to help us get a better understanding of what it's like to work with](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=146.12) [constructors, in our next clip, let's jump back into our code, we'll add some constructors to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=c06b923e-146e-494e-b569-b6b3e1f15606&startTime=148.96)

[Adding Constructors to MathEquation](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f)

[Here we are back in IntelliJ,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=0.64) [and what we want to do now is add the appropriate](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=1.75) [constructors to the MathEquation class that we added to](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=4.25) [this project in the previous module.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=7.17) [As you recall, the way the code works is we have this method here,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=9.28) [performCalculations.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=11.7) [It creates an array of type MathEquation and then goes through](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=12.61) [and creates a series of MathEquation instances that we can](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=16.75) [then run and display the results.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=19.36) [Now remember the way we currently create each instance is down](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=21.15) [here in this create method within the Main class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=24.48) [So the Main class is responsible to go ahead and create](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=27.24) [an instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=29.73) [And each time it creates an instance,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=31.31) [it's using the default constructor that Java provided for us.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=32.84) [Then we go through and set the leftVal, rightVal,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=36.11) [and opCode fields and then return back that reference](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=38.75) [to the new MathEquation instance.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=41.79) [Now one quick thing to note,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=44.24) [I have made one small change to this code since our previous module.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=45.54) [As we initially wrote this code,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=49.94) [we allowed the Main class to directly access each](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=51.7) [of the fields within the class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=54.67) [And as we've talked about, that's not really a good best practice.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=56.05) [So I've actually updated the MathEquation class to have the](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=59.94) [appropriate getters and setters for each of our fields.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=62.58) [You can see here where I'm using the setters to set each field value.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=65.44) [All right, so how can we use constructors to improve this code?](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=69.14) [Well right now,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=72.14) [the entire burden of setting up the MathEquation instance to](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=73.09) [be in an appropriate usable state is being placed on the code](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=76.63) [that uses our MathEquation class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=80.29) [But with the appropriate constructors in place,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=82.59) [we can put that work inside the MathEquation class itself.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=84.64) [So let's head over here to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=88.24) [And then here in a MathEquation class,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=91.14) [we want to think about what are the appropriate constructors to add?](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=92.33) [Well, I think one appropriate constructor would accept just the opCode.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=96.14) [That would be useful in a situation where we know what](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=100.07) [operation we want to perform, an add, subtract,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=102.23) [multiply, or divide, but don't yet have the values.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=105.03) [So let's go ahead and add a public constructor that](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=108.38) [accepts the opCode as a parameter.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=110.01) [Then here inside this constructor, we'll go ahead and set our opCode field.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=113.34) [So now we're all set with this constructor.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=117.54) [And we also know we need another constructor for our current use case.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=119.65) [We need a constructor that accepts the opCode, leftVal, and rightVal.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=123.44) [So now in this constructor,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=128.74) [we'll start by setting are leftVal and rightVal fields.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=129.63) [And then for our opCode field, rather than setting the value here,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=134.04) [let's chain to the constructor that accepts the opCode,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=137.38) [passing in our opCode.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=140.1) [And now with that, we can create an instance of our MathEquation class,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=142.08) [passing in the opCode, leftVal, and the rightVal or just passing in the opCode.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=145.69) [Now before we do anything else,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=151.18) [let's go ahead and run our code and verify that everything still works.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=152.32) [And you notice that when we do that,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=156.04) [we get an error message, and it says that no suitable](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=157.56) [constructor is found, and the code it's showing us is our](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=160.23) [create method over here in our Main class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=163.31) [Well remember that this method relies on our default constructor, and](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=165.82) [that default constructor is being provided for us by Java because](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=169.67) [previously our class didn't have any constructors.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=172.78) [But once we add one constructor, we're responsible](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=175.84) [to provide all the constructors.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=178.48) [So let's head back to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=180.65) [Let's go ahead and add default constructor that doesn't do anything.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=183.8) [So now with that, our class explicitly contains a default](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=187.94) [constructor. So let's go ahead and run the code again. And](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=190.55) [you'll notice that now it runs properly.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=194.28) [So with that code all working, let me go ahead and collapse the run window, and](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=197.04) [we'll head back over to our Main class. So now we're back over here in our Main](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=199.84) [class. Let's take a look at this create method.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=204.01) [If you think about it,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=206.52) [we don't really need this create method anymore because all the](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=207.22) [work the create method is doing is now encapsulated in one of the](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=210.06) [constructors for our MathEquation class.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=213.39) [So I'll go ahead and highlight this method, and](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=215.84) [then I'll go ahead and delete it.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=218.33) [So now up here where were previously calling the create](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=220.34) [method, we'll instead use our constructor.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=222.41) [Let's start out with the zeroth element of our array where we divide 100 by 50.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=225.48) [Rather than calling create, we'll simply say new](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=230.21) [MathEquation, passing in the parameter values.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=232.7) [So now as we're creating this new instance of MathEquation, we want to](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=236.84) [be sure we have the parameters in the right order.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=239.85) [The Main class's create method accepted the opCode d as](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=242.36) [the last parameter. In our constructor, that should be the first parameter.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=246.2) [So let me go ahead and move our d value to the appropriate position.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=250.66) [So now we're properly creating an instance of our MathEquation class, so let's](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=254.68) [do the same thing for each of the other three instances.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=258.24) [And so with that, we should be all set.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=262.44) [We create each instance of our MathEquation class,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=264.17) [passing in the appropriate opCode, leftVal,](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=266.66) [and rightVal.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=269.22) [So when we go ahead and run this, we'll verify that](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=270.44) [everything works as it should.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=272.16) [And as you can see, we now display the appropriate results.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=274.44) [And the key thing we've done here is encapsulate the details of establishing](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=277.58) [the initial state of the class within the class itself.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=281.91) [We're no longer burdening the code that uses our](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=285.14) [class with those kind of details.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=287.07) [All right, so now in our next clip, let's take a look at something known as initialization blocks.](https://app.pluralsight.com/course-player?clipId=72d42344-c0eb-4f57-838b-063a5327954f&startTime=289.84)

[Initialization Blocks](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81)

[Now another option for incorporating code that automatically runs whenever a](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=0.64) [class instance is created is known as initialization blocks.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=4.39) [Initialization blocks allow us to have code that's](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=8.54) [shared across all constructors.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=11.21) [Basically it's a block of code the automatically runs no matter what](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=13.2) [constructor is used to create a new instance of the class.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=17.28) [Now initialization blocks do not accept any parameters.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=20.74) [They are simply a block of code.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=24.24) [And the way we specify them is having opening and closing brackets](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=26.06) [containing the code outside of any method or constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=29.78) [Now a class can have zero or more initialization blocks.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=34.1) [But understand, if you provide more than one initialization block,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=37.94) [all of the blocks will always run.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=41.5) [Basically, when we create in the instance of the class,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=44.24) [the block that's closest to the top of the source code file runs first,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=46.05) [the one below that runs next,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=49.94) [and so on until all of the initialization blocks have been run.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=51.52) [So let's take a look at where initialization blocks can be helpful.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=56.26) [So let's again look at our Flight class.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=59.24) [As you recall,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=61.07) [our Flight class has a passengers field. It also has a seats](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=62.02) [field that we've initialized to be 150.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=65.13) [So let's look at adding a couple more fields to our Flight class](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=68.6) [that we can use to identify an individual flight.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=71.22) [So we'll have a flightNumber field,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=75.14) [which we'll use in cases where a flight has a very specific flight number,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=76.79) [flight 100, flight 450, that sort of thing.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=80.44) [Other flights we'll identify with a more general value known as the flightClass.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=83.84) [Is this a class A flight, is this a class B flight, that sort of thing.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=88.64) [Now in addition to identifying the flights more specifically,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=92.84) [we also want to keep track whether a particular](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=96.09) [seat on that flight is available.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=98.15) [So what we'll do here is have a boolean array,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=100.34) [isSeatAvailable, that will tell us if a particular seat is still available.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=102.44) [Now remember that a field that's a boolean type defaults to false.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=107.79) [So when we declare this array isSeatAvailable,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=111.94) [all the elements in it are going to start out as false,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=114.16) [and that's not what we want.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=117.17) [We want to start out with the value as true,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=118.21) [indicating that all the seats are available.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=120.31) [Now one way we could initialize all the elements of the array](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=123.14) [to start out with a true value would be to go ahead and](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=126.31) [provide our default constructor and, inside that default constructor,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=128.6) [loop through and set each element of the array to have a value of true.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=132.15) [And that would certainly work well enough.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=136.54) [We would have important implications on other constructors we add to the class.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=138.51) [So let's look at some of the other constructors we](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=142.7) [might want to add to this class.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=144.04) [Well, remember that when we create some flights,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=145.84) [those flights are going to be identified by a flight number.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=147.48) [So we'll have a constructor that receives a flight number. And inside](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=150.64) [that constructor, we'll set our flight number field.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=153.43) [But at this point, the code in the default constructor will not run.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=155.93) [The only way to run the code into the default constructor is to](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=159.89) [actually chain to that default constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=162.73) [So that means we want to add another constructor that allows us to create a](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=165.78) [flight identifying its flightClass. Inside that constructor,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=168.6) [we'll of course have to set the flightClass field, but we'll](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=171.94) [again have to chain to the default constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=174.81) [We'd have to do the same sort of thing for any other constructor we add](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=178.14) [to the Flight class. If we want to be sure that the isSeatAvailable](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=181.38) [array is properly initialized, we're always going to have to explicitly](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=184.94) [chain to that default constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=188.69) [So let's see how initialization blocks can help us with that.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=191.58) [Let's head back here to the top of our Flight class and down](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=194.5) [here where we have our default constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=197.49) [Let's get rid of that default constructor, and instead](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=199.32) [we'll have an initialization block.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=201.77) [And you'll notice the initialization block has that same code that](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=204.14) [we had previously. We're simply looping through the isSeatAvailable](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=207.32) [array, setting all its members to true.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=210.34) [But notice our initialization block doesn't have a name or anything like that.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=212.94) [It simply has the code contained within opening and closing brackets.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=216.84) [And it's this syntax that creates the initialization block. And the code in](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=220.64) [the initialization block will now run automatically no matter what](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=224.8) [constructor we use to create an instance of our class.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=228.52) [So if we take a look at adding those other constructors again, we have](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=231.94) [the one constructor that accepts a flightNumber.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=234.96) [The only thing we have to do in here now is set that flightNumber field. The](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=237.79) [code contained within the initialization block will automatically run. Same](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=242.1) [thing for the constructor that accepts a flightClass.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=246.91) [We simply do the core work, we just set our flightClass field,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=249.34) [the code in the initialization block will again](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=252.85) [automatically run. Now something to keep in mind.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=255.32) [Initialization blocks do not replace the default constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=258.69) [If we want to allow instances of our class to be created without](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=263.24) [passing either a flightNumber or flightClass,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=267.18) [then we need to go ahead and include a default constructor.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=269.69) [And then if we have code that creates an instance of our](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=272.94) [Flight class using this default constructor,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=275.25) [the code contained with the initialization block will still run](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=277.67) [just like it does in the other constructors.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=280.63) [Now throughout this module,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=284.34) [we've learned a number of mechanisms for establishing](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=285.53) [the initial state of our classes.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=287.87) [So let's just take a quick look at what order that](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=290.04) [initialization work actually occurs.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=292.63) [So now when we create a new instance of a class, the first](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=295.14) [thing that occurs is the field initializers.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=297.38) [So basically,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=300.14) [all the fields that specify their values using a field initializer have](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=300.95) [their values set before any of our other work occurs.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=304.96) [Once the field initializers are done,](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=308.94) [then the code in the initialization blocks run. So that code is all run](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=310.77) [before any constructor code is run. Only after the initialization blocks](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=315.34) [are complete do we run the code in the specified constructor. And this](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=319.45) [order is followed each time we create a new instance of a class. So this](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=323.63) [gives us an orderly and predictable way to establish the initial state of our class instances.](https://app.pluralsight.com/course-player?clipId=5bcfe7fd-7330-41b2-b2c2-a238b8d58c81&startTime=327.97)

[Summary](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a)

[To wrap up, here are some of the key things we want to remember from](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=0.64) [this module. Remember that whenever we create a new instance of a](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=3.65) [class, that class instance is going to be in some initial state. That](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=6.38) [initial state is expected to be useful.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=10.3) [We wouldn't want a new instance of our class to be](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=12.63) [loaded up with garbage values.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=14.64) [So to help us with that, Java does provide default values for our fields.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=17.24) [For example,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=21.24) [numeric fields start out with a default value of 0, whereas fields that are](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=21.66) [reference types start out with a default value of null.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=25.25) [But we can also provide specific values for our fields using field initializers,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=28.34) [and that allows us to assign a value to the field as](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=32.88) [part of the field's declaration.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=35.47) [Now it can be a simple assignment of our literal value,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=37.44) [or it can be something more involved.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=39.93) [Field initializers can include equations, other fields, even method calls.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=41.74) [Then,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=47.59) [after field initializers we looked at constructors, and](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=47.76) [constructors give us code that can run during object creation. And](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=50.66) [constructors can accept 0 or more parameters.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=54.49) [Remember,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=57.99) [a constructor that accepts no parameters is called the default constructor.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=58.25) [But keep in mind a class can have multiple constructors,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=62.34) [and the specific constructor that's used is based on the parameters that](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=65.47) [are passed when the new class instance is created. Now remember that if a](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=68.58) [class has multiple constructors, one constructor can call another](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=72.82) [constructor. Now if we want to do that,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=76.65) [the call to the other constructor has to be the](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=78.74) [first line of the new constructor,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=80.81) [and when we make that call we can pass parameters, which means that one](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=82.67) [constructor can pass data values to another constructor. And remember that](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=86.2) [constructors don't all have to be public, our classes can have nonpublic](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=90.21) [constructors, and the reason we do that is to limit which code can perform](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=94.8) [specific types of instance creation.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=98.86) [Some phases of creating an instance of our class should not](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=101.24) [be accessible outside of the class itself.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=103.84) [And then we finished up with initialization blocks.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=107.54) [Initialization blocks also give us code that's run during object](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=110.29) [instance creation. So remember that initialization blocks are not](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=113.74) [tied to any specific constructor.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=117.21) [The code within initialization blocks runs no matter which constructor is](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=119.54) [used to create the class instance. And also keep in mind,](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=123.08) [the initialization blocks cannot receive parameters.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=126.05) [There's simply a block of code that runs when you create a new instance of the](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=128.75) [class. Alright, that wraps up this module. In our next module, let's take a look at something known as static class members.](https://app.pluralsight.com/course-player?clipId=bbd31603-b7d9-4f3c-bcc7-4af8785a980a&startTime=131.87)

[Using Static Members](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c)

[Introduction](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c)

[Welcome to our next module, Using Static Members.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=0.74) [This is part of the Pluralsight course,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=3.56) [Working with Classes and Interfaces in Java,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=4.97) [and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=7.26) [There are times when we are designing our classes that we need to have](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=10.27) [values and actions that are associated with the class itself,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=13.24) [and that's where static members come in.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=16.84) [So in this module, we'll start out with a quick overview of static members.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=18.88) [We'll then look at how we can have static fields,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=23.44) [static methods, and we'll also look at something new with static imports,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=25.76) [which allow us to use a shorthand when accessing static methods.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=29.46) [And then we'll finish up with a look at how it can do type](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=32.83) [initialization with static initialization blocks.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=35.1) [Now when we associate members of our classes,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=39.34) [most commonly those are instance members.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=41.72) [Meaning that each time we create an instance of the class,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=44.41) [that instance has its own copy of the values,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=46.82) [and the methods on that instance interact with those copies.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=49.64) [Static members are different.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=52.89) [Static members are shared class‑wide.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=54.25) [Meaning that static members are not associated with an individual instance.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=56.9) [They're associated with the class itself.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=60.67) [Now the way we declare static members is by using the static keyword.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=63.44) [The declarations are very much like the way we declare instance members,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=67) [we just annotate them with a static keyword.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=70.57) [And because these members are associated with the class itself,](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=73.59) [we're going to access them using the class name rather than relying on an](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=76.29) [instance variable. Hurry to turn on our next clip, let's start looking at the details of how we work with static members.](https://app.pluralsight.com/course-player?clipId=b26d1184-624c-4276-8d39-e0ce070d2a7c&startTime=79.94)

[Static Members](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53)

[So now the first type of static member we'll look at are static fields.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=0.64) [A static field is a value that's not associated with](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=4.34) [a specific instance of a class.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=7.51) [This value is associated with a class itself,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=9.44) [and what that means is each instance of the class accesses the same value.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=12.4) [There's not a separate copy for each instance, there's](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=16.84) [just one value for the whole class.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=19.13) [So let's look again at our Flight class. Now we know our](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=23.04) [Flight class already has a couple fields,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=25.57) [passengers and seats.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=27.9) [Now although we normally call these just fields, in](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=30.09) [reality they're instance fields, again,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=32.28) [meaning that each instance of the Flight class will](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=35.04) [have its own copy of these fields.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=37.45) [But now if I declare another field here,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=40.14) [allPassengers, and indicate that it's static,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=41.99) [that means that there's one value for the whole class.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=44.96) [Each instance of the class will access that exact same value.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=47.88) [So let's see how this would impact our add1Passenger method.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=52.74) [Now as you recall, the way this method is implemented,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=55.94) [so we use our two instance fields, passengers and seats,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=58.56) [and as long as there are still seats available, we go and increment](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=61.78) [passengers. So that would increment the number of passengers](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=64.52) [associated with a specific instance of flight.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=67.64) [But now let's go ahead and add 1 to allPassengers as well.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=70.94) [Now remember that allPassengers is static, meaning that](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=74.64) [it's not tied to a specific instance.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=77.07) [So what allPassengers would allow us to do is count the total number of](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=79.94) [passengers across multiple instances of the Flight class.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=83.45) [Now in addition to static fields, we also have static methods, and a](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=88.44) [static method performs an action that's not tied to a specific](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=92.22) [instance. In other words, it's an action that's tied to the class. And](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=95.69) [static methods can only access other static members. Static methods](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=99.87) [cannot access instance members.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=104.78) [So again, looking at our Flight class,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=108.44) [we have our instance fields, passengers, and seats. We have our static](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=110.35) [field, allPassengers. Well allPassengers is private,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=114.22) [so it's not accessible outside the class.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=117.89) [So if we want to make its value available, we'll need to wrap it in a getter.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=120.74) [So notice our getAllPassengers method is also marked as static,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=124.54) [and so it can return that value for allPassengers.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=128.5) [But now if we're using allPassengers to keep track of the](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=132.34) [number of passengers across multiple flights, we might](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=135.25) [periodically want to reset it back to 0.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=137.98) [So we have another static method here, we set](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=140.94) [allPassengers, that will set allPassengers = 0.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=142.87) [So now with that, our Flight class has a static field,](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=147.14) [allPassengers, and two static methods, getAllPassengers and](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=149.48) [resetAllPassengers. So now in our next clip, let's take a look at how we'll use these static members.](https://app.pluralsight.com/course-player?clipId=839f059d-9214-4a42-8502-ffc12587fd53&startTime=153.49)

[Using Static Members](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166)

[So let's see what it's like to work with our Flight](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=0.74) [class now we have some static members.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=2.19) [So let's start out by creating an instance of our Flight class,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=4.74) [and we'll assign it to a variable, laxToSlc.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=8.31) [So as we know, that'll give us that variable.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=11.99) [It'll hold a reference to an instance of our Flight class.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=14.15) [And note that that instance has its own copy of the fields,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=17.4) [passengers, and seats.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=20.85) [And then if we go off and create another instance of our Flight class,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=22.21) [we'll assign that to a variable dfwToNyc.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=25.48) [Again,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=27.98) [that creates our variable and it holds a reference to a](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=29.17) [separate instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=31.77) [And this separate instance has its own copy of the fields,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=33.93) [passengers, and seats. Now remember that our Flight](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=37.94) [class now has a static member, allPassengers.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=40.78) [And as a static member,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=43.27) [that field is associated with the class itself being that it's](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=45.38) [separate from any instance of the class.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=49) [Now just like instance members, a static integer field will default to zero.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=52.1) [But let's go ahead and call resetAllPassengers anyway because](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=56.84) [it's possible that some code earlier has already been](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=60.09) [affecting the value of allPassengers.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=63.38) [So if we want to keep track of the total passengers on just these two flights,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=65.64) [we'll call resetAllPassengers just to make sure allPassengers is set to zero.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=69.14) [Now notice here when we call resetAllPassengers,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=74.14) [we're calling it using the class name because,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=76.63) [again,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=79.25) [we setAllPassengers as a static method so it's accessed using the class name.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=79.49) [So let's go ahead and call add1Passenger on our laxToSlc variable.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=85.04) [Now of course, that will follow that reference to the appropriate instance.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=89.64) [When we call add1Passenger, it'll take our passengers](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=93.44) [instance field and increment that from 0 to 1. Remember that](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=95.67) [we also now increment allPassengers.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=99.84) [So that'll also increment from 0 to 1.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=102.64) [We'll call add1Passenger again on laxToSlc.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=106.04) [So our instance field passengers goes from 1 to 2, and our](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=109.74) [static field allPassengers also goes from 1 to 2.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=113.54) [Now let's go ahead and call add1Passenger on dfwToNyc.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=117.74) [So again, we'll follow the reference to the appropriate instance,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=122.14) [and that will increment the copy of passengers](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=125.44) [contained in that instance. But again,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=127.29) [we're also going increment allPassengers. AllPassengers is a static](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=129.78) [field again that's shared across all instances.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=133.65) [So allPassengers will increment from 2 to 3.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=136.84) [So if we print out the value for getPassengers on laxToSlc, that would,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=140.54) [of course, return 2. Do the same thing for dfwToNyc, that,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=145.44) [of course, will return 1.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=150.49) [And then if we print out the value for](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=152.07) [flight.getAllPassengers, that would print out the value 3.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=153.8) [So notice that each instance is tracking its own values because](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=158.64) [it has its own copy of the instance values.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=162.4) [But all of the instances affect a single static value.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=164.98) [All right,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=169.54) [so to help us get a better understanding of all this, in our next clip,](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=170.22) [let's jump back into our CalcEngine project. We'll start adding static members to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=1e976327-dbd9-40b7-92cf-5c2c1cdb1166&startTime=173.56)

[Enhancing MathEquation with Static Members](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b)

[Here we are back in IntelliJ looking at our CalcEngine project,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=0.74) [and what we want to do now is see how we can use static members to](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=4.04) [add a new feature to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=7.13) [And the feature we want to add is to provide the ability to get back](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=10.54) [the average result for a series of calculations.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=14.17) [And we want to calculate this average result in a way that's not tied](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=17.5) [to any single instance of the MathEquation class.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=21.14) [We want an average result across all the instances of MathEquation.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=24.44) [So to do that, we'll need to use static members. Now](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=29.14) [in order to determine the average,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=32.79) [one of the things we'll need to know is how many](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=34.38) [times we've calculated our equations.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=36.67) [So let's go ahead and add a static field here named](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=39.14) [numberOfCalculations of type int.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=41.45) [Now once we have that field, we'll also need a field to hold the](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=45.24) [sum of all the results of these calculations.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=48.47) [So let's add another static field named sumOfResults of type double.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=51.44) [So now with these two fields in place, we have everything we need to start](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=57.14) [calculating that average. Now remember that because these fields are static,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=60.61) [there's only one copy of these fields for the whole class.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=64.32) [It doesn't matter how many instances of the class we create.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=67.89) [Also, keep in mind that static fields,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=71.49) [just like instance fields, automatically have default values set.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=74.01) [So both of these fields will start out as zero, which is exactly what we want.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=77.96) [All right,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=82.54) [so now that we have these fields declared, let's head down to](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=82.8) [our MathEquation class's execute method.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=85.05) [So now we're down here in our execute method.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=88.94) [And as you recall,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=91.1) [the way our MathEquation class works is that we provide an opCode and two](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=92.14) [values, and then we call execute to actually determine the result.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=95.81) [So inside this method is a great place for us to keep count of](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=100.04) [the number of calculations we had to execute.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=103.17) [So here after the switch statement where we](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=105.74) [determine the results for the equation,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=107.28) [let's go ahead and increment the number of calculations by 1.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=109.36) [And then once we've done that,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=113.79) [we can take the result for this MathEquation instance and add](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=115.07) [that result to our field someOfResults.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=118.33) [So now with those two statements in place, that gives us everything](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=122.34) [we need to calculate our average result across all the different](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=124.88) [instances of our MathEquation class.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=128.59) [So all we need to do now is provide a method to get back that value.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=131.24) [So let me just scroll down here a little bit.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=135.77) [So here we are now just after the execute method.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=138.59) [So let's go ahead and add a static method named getAverageResult of type double.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=141.14) [And then inside this method,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=146.74) [we can go ahead and calculate the average result and return that](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=147.86) [value back. So to calculate the average, we'll take our sumOfResults](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=150.56) [and divide it by our number of calculations. And that easily we now](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=154.8) [have the ability to determine the average result across all our](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=159.85) [MathEquation instances.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=163.16) [So now what we need to do is display that value out.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=165.34) [So to do that, let's head back over here to our Main class.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=167.99) [Now remember here in our Main class,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=172.39) [we have this method performCalculations. PerformCalculations](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=173.77) [has an array of MathEquation instances.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=177.4) [We then loop through that array,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=180.29) [calling execute on each of those instances and to](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=181.94) [display the corresponding result.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=184.66) [So that means that once this loop completes, we should be](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=186.9) [able to callGetAverage result and display out the average](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=189.22) [for all those calculations.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=192.52) [So just after our for loop, we'll call System.out.println. And then to](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=194.54) [display our average, we'll call getAverageResult.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=200.12) [And remember, because it's a static method,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=202.94) [we're going to access it through the MathEquation class](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=205.16) [name. And that's all there is to it.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=207.63) [So let's go ahead and run our application and](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=210.86) [verify that it behaves as we expect.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=212.57) [So now once the program runs, let's take a look here at our output.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=216.14) [You can see the results of each of the MathEquation instances.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=219.64) [And if we were add each of those results up,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=222.75) [it would give us a total of 360. And we have four results there,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=224.73) [so if we divide 360 by 4, that gives us an average result of 90,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=228.7) [and that's exactly what we displayed.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=232.77) [So our code is doing exactly what we want it to do.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=235.44) [Each MathEquation instance took care of the details of that individual equation,](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=238.34) [and then the addition of our static fields allowed us the](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=242.76) [track information across those instances.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=244.78) [All right, so now in our next clip, let's take a look at something known as static imports.](https://app.pluralsight.com/course-player?clipId=a45a2435-baf4-4d1a-89b0-ee75f2c7571b&startTime=248.24)

[Static Imports](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f)

[Let's take a look now at static import statements.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=0.74) [As you recall in our previous course, Getting Started with Programming in Java,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=3.86) [we talked about standard import statements.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=7.44) [As you recall,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=10.14) [an import statement allows us to have a shorthand for accessing](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=10.74) [a type because in Java type names are fully qualified by the](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=14.11) [package that contained the type.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=18.67) [And having to type the full package qualified name of every type](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=21.09) [out every time would be really cumbersome.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=24.35) [So import statements allow us to specify at the top of the source file,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=27.54) [the package qualified name of the type.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=31.21) [And then from that point forward, we can simply use the simple name of the type,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=33.7) [which in general is just the class name.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=37.15) [Well static import statements take this idea one step farther.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=39.84) [Static import statements allow us to access static](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=43.81) [methods with a simplified notation.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=46.01) [What static import statements allow us to do is specify the fully](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=48.33) [qualified name of the method at the top of the source file.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=52.55) [And then from that point forward, we can use the method named by itself.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=56.39) [We no longer have to qualify it by the class.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=59.87) [So let's look at some code again that uses our Flight class.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=62.87) [As you recall, our Flight class has a static method, resetAllPassengers.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=66.63) [Since it's a static method, we access it through the class name,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=71.5) [so it's flight.resetAllPassengers.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=74.53) [Then we have some code that creates an instance of our Flight class,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=77.68) [call add1Passenger on that instance twice.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=80.32) [We create another instance of our Flight class,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=83.34) [call add1Passenger on that instance. And then we want to get the](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=85.6) [total number of passengers across those two flights, so we use](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=88.47) [our static method, getAllPassengers,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=91.7) [and again we qualify that method by the class name.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=93.96) [Now let's see how using a static import changes this code.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=97.54) [Now as you recall,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=101.14) [the way a standard import statement works, so we simply say](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=102.08) [import and the package qualified name of the type.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=104.84) [Well, in the case of a static import,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=108.34) [we're going add the word static after the word import.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=109.96) [And in the case of a static import, we're referencing a static member.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=113.58) [So we're going to give the name of the method that we want to use.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=117.24) [So now we have a static import for our Flight class's resetAllPassengers](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=121.34) [method. So if we look at this code here where we currently say](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=125.43) [Flight.resetAllPassengers, once we have the static import in place,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=128.72) [this line can be simplified to just resetAllPassengers.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=133.7) [We can use the method name without having to qualify it by the class.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=137.74) [Now we can do the same sort of thing where we call Flight.getAllPassengers.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=142.04) [We can add a static import for getAllPassengers, and this line of code now](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=146.54) [can be simplified again simply using the method name. And static imports are](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=150.57) [useful in scenarios where you have some static methods you just have to](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=155.49) [access very frequently. Now in scenarios where a class has a large number of](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=159) [static methods that you want to use rather than having to list them](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=163.7) [individually as we've done here,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=166.53) [we can actually switch to a star notation. So we can boil](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=168.45) [this down to simply Flight.\*. So basically,](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=171.63) [we're saying we want to import all the static members of the Flight class.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=175.64) [The code to use those members is just as it was before.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=179.39) [We simply used the member names without having to qualify them by the class.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=183.04) [The value of the star notation is that we only need one static](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=186.94) [import to have access to all those static members.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=190.15) [All right, so now in our next clip, let's take a look at static initialization blocks.](https://app.pluralsight.com/course-player?clipId=b3ae8749-368f-47bc-ab3a-26edc61ce59f&startTime=194.04)

[Static Initializers](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f)

[Static initialization blocks allow us to perform one‑time type initialization.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=0.64) [The key is the code in a static initialization block is](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=6.18) [executed before a type's first use.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=9.03) [So it gives us a way to centralize the work we want](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=12.44) [to do to prepare a type for use.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=14.83) [Now static initialization blocks are limited to accessing only static members.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=16.87) [Now the way we create static initialization blocks is very similar to](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=22.24) [the way we created instance initialization blocks.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=25.54) [It's going to be code enclosed in brackets to mark it](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=28.54) [as a static initialization block, the brackets are preceded by the word static,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=31.38) [and this code is going to be outside of any method or constructor.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=35.84) [So to see how to use a static initialization block,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=40.54) [let's again look at our Flight class.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=43.11) [Now recall our Flight class has these two instance fields, passengers and seats.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=45.05) [We also have a static field, allPassengers.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=49.44) [Well let's go ahead and another static field, maxPassengersPerFlight.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=52.44) [And what we use this field for is that there are some scenarios where the](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=56.64) [flight administration authority may be putting limits on how many passengers](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=59.63) [are allowed on a given flight. And in some cases, that limit might be lower](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=63.39) [than the number of seats a flight actually has.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=67.79) [And this limit is going to be the same for all flight instances.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=70.64) [But in order to find out the limits,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=74.42) [we need to go out and connect up to a service and](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=76.05) [retrieve the value back from that service.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=77.94) [So we'd like to do that lookup work once before we start using our Flight class.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=80.29) [So this is a great chance for us to use a static initialization block.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=84.74) [So here within these brackets preceded by the word static](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=88.64) [is our static initialization block.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=91.27) [So inside of our initialization block, we'll go ahead and create an instance](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=93.34) [of the admin service and then connect up to the service.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=96.47) [Once we connect to the service,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=100.01) [we're going to go ahead and check and see are there any restrictions in effect?](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=101.51) [Now if there are restrictions in effect,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=105.14) [we're going to go out to the admin service and call](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=106.86) [its getMaxFlightPassengers method,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=108.65) [and that will tell us the limit that they've set for the](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=111.34) [number of passengers on a given flight.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=113.67) [But if there aren't any restrictions in effect,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=116.04) [we're using the constant Integer.MAX\_VALUE. And that](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=117.58) [constant contains the largest possible integer.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=120.98) [In other words,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=124.01) [that constant gives us a value that indicates there's no](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=124.43) [administrative limit on the number of passengers.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=127.55) [Now once we're done with the administrative service, we'll go ahead](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=130.44) [and close it. Now again the code within this block will run once](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=132.79) [before the first time we use the class.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=136.42) [So that means the work to initialize our field maxPassengersPerFlight](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=138.84) [is done before we start using our Flight class.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=143.11) [So let's see how we can use that field in the rest of our code.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=146.34) [As you recall, our Flight class has an add1Passenger method.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=149.24) [Now currently the way this method is implemented is we simply](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=152.9) [check the number of passengers against the number of seats. As](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=155.35) [long as there are seats available,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=158.7) [we'll go ahead and increment the number of passengers by 1, as well](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=160.32) [as increment are static field allPassengers by 1.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=163.35) [So now we need to evolve this code because seats aren't the only possible limit.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=166.64) [So what we'll do is in addition to checking against the](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=171.24) [number of seats, we'll also make sure the passengers is less](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=173.45) [than the maxPassengersPerFlight.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=176.83) [So now with this code in place,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=179.14) [we only add another passenger if there are still seats available and](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=180.43) [we haven't exceeded any administrative limits.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=184.74) [And because we set our maxPassengersPerFlight field as](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=187.64) [part of our static initialization block,](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=190.46) [the work of retrieving that value only had to be done one time, but was available to all instances of our Flight class.](https://app.pluralsight.com/course-player?clipId=0db0fa6d-c7ef-4b3a-86d4-7124bd17c14f&startTime=192.4)

[Summary](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966)

[To wrap up, here are some of the key things you want to remember](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=0.7) [from this module. Remember throughout this module,](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=3.29) [we've been looking at static members.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=5.88) [Now static members are members of our class that are shared class‑wide. Now](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=7.92) [the way we indicate that the member is static is simply by including the](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=12.6) [keyword, static, as part of that member's declaration.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=15.88) [Now the first kind of static member we looked at were static fields.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=20.04) [Now the static field is a value that's not associated with any instance.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=23.44) [In other words, all instances of the class access the same value,](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=27.61) [so it's a value that shared across all class instances.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=31.99) [Now we also have static methods.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=35.94) [Static methods allow us to perform an action that's not tied to an instance.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=37.84) [In other words, the action is tied to the class itself.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=41.9) [Remember that static methods can only access static members.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=44.78) [So static methods can only access other static methods or static fields.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=48.54) [Then we looked at the static import statement. Remember, the static](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=54.14) [import statement gave us a shorthand for accessing static methods.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=57.37) [Normally our static methods have to be qualified by the type name. By](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=61.64) [using a static import, we're able to use that method by using the](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=65.45) [method name by itself, not having to qualify it by the class name.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=68.64) [And then we finished up with static initialization blocks.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=72.94) [Remember, static initialization blocks allow us to perform one‑time](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=76.14) [type initialization. Because the code we put inside these blocks](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=79.62) [executes before a type's first use.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=83.02) [So stack initialization blocks give us a great way to do the work](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=86.19) [in our type, before we start using that type.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=88.84) [All right, that wraps up this module. In our next module,](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=92.14) [we're going to take a closer look at methods and take a look at some of the richer capabilities that methods provide.](https://app.pluralsight.com/course-player?clipId=5cc1a5e6-46d9-4d85-aa64-e81da3860966&startTime=95.43)

[A Closer Look at Methods](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720)

[Introduction](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720)

[Welcome to our next module, A Closer Look at Methods.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=0.69) [This is part of the Pluralsight course,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=3.59) [Working with Classes and Interfaces in Java,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=5.13) [and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=7.37) [In our previous course, Getting Started with Programming in Java,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=10) [we introduced methods,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=13) [and as part of that discussion, we looked at how we can use](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=14.36) [parameters to pass data values into our methods.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=16.74) [Well, in this module,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=20.09) [we're going to take a closer look at Java's capabilities](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=21.31) [for accepting values and parameters.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=23.71) [We're going to take a look at some of the richer features that it provides.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=25.73) [Now to start out, we'll take a look at passing objects as parameters.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=28.49) [In other words,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=32.14) [just what happens when you pass an object as a parameter to a method?](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=32.85) [Then, of course,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=37.1) [the next question is, just what happens if that method starts making changes](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=37.53) [to an object that it receives as a parameter? Then from there we'll take a](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=41.59) [look at something known as method overloading, and method overloading allows](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=45.29) [us to have multiple versions of the same method. And if we can have multiple](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=48.79) [versions of the same method,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=52.21) [there must be some set of rules for how Java decides which one](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=53.48) [to call. So we'll take a look at how Java goes about resolving](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=56.8) [which overloaded method to use.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=60.04) [And then finally, we'll finish up with a look at variable number of parameters.](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=62.29) [In other words,](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=66.34) [how can we create a method that can accept different number of parameters at different times?](https://app.pluralsight.com/course-player?clipId=fe32dfc1-a98a-4b76-9c48-c3887c2ac720&startTime=66.87)

[Passing Objects as Parameters](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287)

[Now as you recall from our previous course,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=0.64) [when we talked about passing primitive types as parameters,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=2.74) [we said that primitive types were passed by value,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=5.33) [meaning that a copy of the value was passed into the parameter.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=9.05) [Well, when it comes to passing objects as parameters,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=13.54) [objects are passed by reference.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=15.92) [What that means is, we don't pass a copy of the entire object into the method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=18.74) [Instead what we do is,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=23.34) [we pass a copy of the reference to the object into the method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=24.7) [So that means that the method will have its own copy of the reference.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=29.14) [Now since the method has its own copy of the reference,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=33.04) [the method is free to make changes to that reference,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=36.02) [and any changes it makes to the reference will be visible inside of the method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=39.04) [But again, because the method is working on its own copy of the reference,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=43.54) [any changes it makes to the reference will not be visible outside of the method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=47.44) [So now, to see what this looks like, we're again going to use our Flight class.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=53.34) [As you recall, our Flight class has a number of fields.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=56.94) [One of those fields is the flightNumber field.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=59.54) [One of the constructors for our Flight class accepts a flightNumber,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=62.24) [and of course, that constructor then sets our flightNumber field.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=64.95) [So let's use our Flight class to see how object parameters behave.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=68.84) [So i have a method here, swapFlight, and it receives two parameters,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=73.44) [i and j, both of type Flight.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=77.36) [And the code inside of swapFlight wants to swap the references for i and j.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=80.24) [So by the end of this method,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=85.84) [i should point to the Flight instance that j originally pointed to,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=87.54) [and j should point to the flight instance that i originally pointed to.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=90.82) [So to call this code,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=95.44) [let's go ahead and create two Flight instances and assign those to](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=96.65) [local variables outside of our swapFlight method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=99.65) [So when we do that,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=103.04) [we'll have a val1 reference that points to an instance of our](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=103.9) [Flight class that has a flightNumber of 10,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=106.63) [and our val2 reference will point to an instance of our](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=109.34) [Flight class that has a flightNumber of 20.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=111.59) [So if we now call swapFlight, passing in val1 and val2.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=114.48) [One side of swapFlight, the space for those two parameters needs to be allocated,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=118.78) [and then the reference contained in val1 will be copied down to our parameter i.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=123.64) [So the reference in i will now point to the same](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=128.34) [Flight instance that val1 points to.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=131.12) [Same sort of thing for val2.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=134.04) [The reference in val2 will be copied down to our j parameter,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=135.81) [so our j parameter will now point to the Flight instance that val2 points to.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=139.56) [So now once we get down inside the code here in our swapFlight method,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=144.78) [well, we'll allocate out a local variable named k,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=148.44) [we'll initialize it to the value of i.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=150.91) [So k will now reference the same object instance that i references.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=153.84) [So then when we say i = j,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=159.18) [what we want to do now is actually make a change to the i reference.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=161.49) [So the reference in i will now point to the same](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=165.88) [flight instance that j points to.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=168.78) [So we've successfully changed that reference.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=171.48) [So then when we get to our next line of code,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=174.84) [j = k, we're again going to operate on our reference j,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=176.77) [and we're going to give the value of k.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=180.03) [So j will point to the same Flight instance that k points to.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=182.69) [So now at the end of this code,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=187.34) [we've successfully swapped our references i and j.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=188.98) [I now points with one that j originally pointed to,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=192.94) [and j now points with one that i originally pointed to.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=195.24) [But again,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=199.24) [all this work was done on the local copies of the](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=199.73) [references that we had inside the method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=202.91) [And when this method exits,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=205.79) [all that gets cleaned up with no changes to the](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=207.24) [original references of val1 and val2.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=210.37) [So if we want to point out the flight numbers for those 2 Flight classes,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=214.04) [if we follow the reference for val1,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=217.5) [it points to the same Flight instance it always did,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=219.49) [so it'd print out the flightNumber of 10.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=222.34) [If we follow our val2 reference,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=224.47) [it also points to the same flight instance it always did.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=226.47) [So it'd still print out its original value of a flightNumber of 20.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=229.73) [So even though the method was able to make changes to the references,](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=234.4) [it was only making changes to its own copies.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=237.8) [Those changes were not visible outside of the method.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=240.54) [But it turns out there are some changes that can be visible outside of the method, and we'll see what those are in our next clip.](https://app.pluralsight.com/course-player?clipId=032ed53c-969b-42e6-8c85-3f46d364c287&startTime=243.59)

[Changes to Objects Passed as Parameters](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7)

[As we mentioned, when we pass objects as parameters to a method,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=0.74) [those objects are passed by reference,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=4.2) [meaning that the method receives a copy of the](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=7.1) [reference back to the original object.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=9.05) [Remember that objects have members,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=11.94) [and since the method has a reference back to that original object,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=14.15) [the method can make changes to those members.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=17.1) [Now any changes the method makes to the members will of](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=19.82) [course be visible within the method.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=22.9) [But it turns out those changes will also remain visible even outside the method.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=25.36) [So those changes will be visible all the way back to the original calling code.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=30.64) [So let's see why this is the case.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=36.74) [Let's have another method here, swapNumbers.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=39.14) [SwapNumbers again receives two parameters, i and j, both of type Flight.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=41.75) [Now here inside of swapNumbers,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=47.44) [we're not going to try and swap the references themselves.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=48.57) [Instead what we want to do is swap the flight numbers](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=52.07) [contained within those flights.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=54.61) [So again we'll have our code here where we allocate out](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=57.44) [two instances of our Flight class.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=59.77) [So val1 will point to an instance of Flight that has a flightNumber of 10,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=62.07) [and val2 will point to an instance of Flight that has a flightNumber of 20.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=66.16) [So we'll call swapNumbers, passing in val1 and val2.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=70.44) [And of course swapNumbers will allocate out space for the parameters,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=74.49) [so we have our parameters i and j,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=77.42) [The reference in val1 will be copied down to our i parameter,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=79.52) [so i will now reference the same object that val1 references.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=83.85) [The reference in val2 will be copied down to j,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=87.83) [so j will now reference the same object that val2 references.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=90.8) [So now when we get down here inside of swapNumbers,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=95.54) [the first thing we do is allocate out a local variable named k.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=98.14) [It's type is int.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=101.76) [So we get the flightNumber from the Flight that i points to,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=102.49) [and we assign that to k.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=106.62) [So now k has that value 10.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=108.89) [So now the next thing we want to do is set the flightNumber and the Flight](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=111.68) [reference by i to the flightNumber of the Flight reference by j.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=114.88) [So now, in this case,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=119.64) [we're not making a change directly to i. What we're going to do is](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=120.8) [follow the reference that's contained in i. We want to modify the](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=123.8) [object that i points to. So what we're going to do then is take the](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=127.67) [Flight instance that's referenced by i and set its flightNumber to](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=131.6) [20. Then the next thing we do is set the flightNumber of the Flight](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=135) [reference by j.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=139.15) [So again, in this case, we're not trying to change j.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=140.78) [Instead,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=143.65) [we're going to follow the reference that's](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=143.93) [contained in j, and when we get there,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=145.25) [we're going to set its flightNumber to now have a value of 10.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=147.99) [So now notice here we haven't tried to change the references; instead,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=151.84) [we're changing members within the objects that these references point to.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=156.16) [So now once our method is complete,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=161.04) [we know that everything that was allocated within that](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=162.92) [message gets cleaned up, so all of our references and local](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=164.79) [variables here are cleaned up.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=167.71) [But now even though those references are cleaned up,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=169.54) [the objects that they pointed to are still around because those are](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=171.79) [the same objects pointed to by val1 and val2.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=175.75) [So if we now go and print out the flightNumbers, we'll](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=179.28) [first follow the reference in val1,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=181.89) [and it points to an object instance that has a flightNumber of 20.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=183.96) [So even though that instance originally had a value of 10, if](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=187.84) [we print out its flightNumber now, it is 20.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=190.66) [The change that was made inside of swapNumbers is still there.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=193.64) [Same thing for val2.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=197.47) [If we follow its reference,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=198.72) [it points to a Flight instance that has a flightNumber of 10, even](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=200.12) [though it originally had a flightNumber of 20.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=203.62) [So the key thing to take away from this is that](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=206.64) [because objects are passed by reference,](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=209.05) [the method has a reference back to the original object.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=212.01) [So any changes it makes to that object are lasting changes and will](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=215.94) [still be in effect even after the method exits.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=219.97) [All right, so now in our next clip, let's take a look at something known as method overloading.](https://app.pluralsight.com/course-player?clipId=c33a41d2-7a98-4e54-9fba-e5af12e308d7&startTime=223.94)

[Overloading](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902)

[Java supports a really important concept known as overloading.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=0.74) [What overloading allows us to do is,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=4.74) [within a single class we can have multiple versions of a particular](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=6.8) [method or multiple versions of our constructor.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=10.2) [And we've actually seen overloading in practice earlier in this](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=13.29) [course when we talked about constructors.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=17.07) [Remember our Passenger class.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=19.24) [Our Passenger class had multiple constructors.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=21.4) [Had a constructor that didn't accept any parameters,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=24.74) [one that accepted the number of freeBags,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=27.26) [one accepted the perBagFee,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=29.56) [and one that accepted the number of freeBags and a number of checkedBags.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=32.23) [So in this scenario, our Passenger class had an overloaded constructor.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=36.04) [Now a key concept in overloading is what's known as the signature.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=41.5) [Every constructor and every method must have a unique signature,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=46.44) [and there are multiple parts that make up the signature.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=50.94) [Now one of those parts is the number of parameters.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=54.34) [So again, looking at our Passenger class,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=57.74) [our first instructor had 0 parameters, we had another](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=60.14) [constructor with a single parameter, and then yet another](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=62.95) [constructor that had two parameters.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=65.69) [So by having a different number of parameters, that allowed them to](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=68.14) [have unique signatures. But notice that number of parameters must](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=71.23) [not be enough because we have two constructors here that both have a](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=74.79) [single parameter. So there must be something more than just the](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=79.16) [number of parameters.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=82.37) [And it turns out another key part of the signature is](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=84.04) [the data type of each parameter,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=86.76) [so developing that unique signature uses both the number of](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=89.34) [parameters and the type of each parameter.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=92.08) [So again, looking at our Passenger class,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=94.56) [we have these two versions of the constructor that both have a single](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=96.84) [parameter. But notice that one's parameter type is int; the other's](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=99.77) [parameter type is double. So even though they have the same number of](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=104.31) [parameters, because the parameter types are different,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=107.86) [the signatures are considered unique.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=110.66) [Now when we're talking about constructors, the number of parameters](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=113.44) [and the type of each parameter is enough to make them unique because](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=117.05) [constructors have to have a specific name.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=120.16) [The constructor's name is always the same as the class](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=122.84) [name. But when it comes to methods,](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=125.15) [the method name is also a key part of the signature.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=127.34) [So when we're talking about methods, it's that combination of the](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=130.94) [method name, how many parameters it has, and the type of each](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=134.05) [parameter that creates a unique signature. So in our next clip, let's take a look at some examples of method overloading.](https://app.pluralsight.com/course-player?clipId=bed890a0-ba10-42a5-8873-05e1e26d4902&startTime=137.62)

[Method Overloading Examples](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963)

[As we look at some examples of method overloading, we're, again, going](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=0.7) [to use our Flight class. As you recall, our Flight class has fields for](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=3.57) [the number of passengers and the number of seats. We have a method,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=7.34) [add1Passenger, that internally will check to see if there's enough seats](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=10.7) [to add the passenger. As long as there are, it will increment the number](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=14.82) [of passengers.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=17.75) [Well,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=18.94) [this code here that checks to see if there are enough seats](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=19.12) [available to add the passenger is something we might want to do more](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=21.25) [often. So let's go and add another method here, hasSeating, that](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=24.28) [contains the logic to do that check.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=28.07) [So as long as the number of passengers is less than the number of seats,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=30.54) [hasSeating will return true. So let's go and update our add1Passenger method](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=33.64) [to use our hasSeating method. And while we're in here in our Flight class,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=37.84) [let's add another field, totalCheckedBags,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=41.93) [and that will allow us to keep track of how many checked bags we](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=45.74) [currently have on the flight, and we can do that as we add each](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=48.95) [passenger. And since we're adding a new feature,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=52.18) [keeping track of the total number of checked bags, we might want to add](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=55.06) [some overloads for our add1Passenger method. So we'll go and keep the](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=58.37) [original version of our add1Passenger method that takes care of](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=63.16) [incrementing the number of passengers, but we'll go and add an overload](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=66.3) [that accepts the number of bags.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=69.73) [So in this case,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=71.79) [we're adding an overload that provides additional](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=72.59) [functionality, accepting the number of checked bags.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=74.98) [And in this overload,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=78.64) [what we'll do is we'll call the original implementation of](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=79.64) [add1Passenger, and we'll increment the total number of checked bags.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=82.14) [Now as this code is written, it will compile just fine,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=87.04) [but it actually has a logic error in it.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=90.34) [Because when we call our new overload of add1Passenger,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=93.14) [the first thing we're going to do is call that](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=96.06) [original implementation of add1Passenger.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=97.8) [When we transfer control to that implementation,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=100.15) [the first thing it's going to do is check to make](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=102.62) [sure that we have seating available.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=104.44) [Well,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=106.64) [if there's no seating available, we're not going to increment the number](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=106.81) [of passengers. But as our new overload is written,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=110.03) [we would still increment the total number of checked bags.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=112.91) [So in order for our new overload to work correctly, we need to](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=116.74) [wrap all this code in a call to hasSeating.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=119.5) [So that way we only increment the number of passengers and the](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=123.34) [total of checked bags as long as there's seating available for](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=126.03) [the passenger that we're adding.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=128.87) [So here in our new implementation of add1Passenger, notice](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=130.94) [that when we call the other overload of add1Passenger, we](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=134.15) [don't use any special syntax.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=137.54) [We call it just like we were calling that](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=139.7) [add1Passenger method from anywhere else.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=141.46) [So that's an important thing to understand.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=144.04) [When one overload calls to another overload of the same method,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=145.71) [there's no special syntax. You call that overload just like](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=149.61) [you would call it from anywhere else.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=152.93) [Also notice that the call to the other overload does not have](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=155.24) [to be the first line in this implementation.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=158.84) [Remember when we were talking about constructors, if one constructor called](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=161.84) [another version of the constructor, that call to the other constructor had](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=164.74) [to be the first line in the new constructor.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=168.21) [No such limitation exists when one overload of a method](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=170.64) [calls another overload of that same method.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=173.86) [So let's take a look at some other overloads of our add1Passenger method](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=176.8) [we might want to add. And something that's important to understand, just](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=179.66) [because you're adding an overload of a method,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=183.07) [it doesn't have to be adding a brand‑new feature. Some](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=185.2) [overloads are provided just for convenience.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=188.05) [So let's say we have an add1Passenger overload here that](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=191.04) [accepts a reference to our Passenger class.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=193.52) [Remember that one of the things the Passenger class can be asked is](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=196.44) [how many checked bags does that passenger have?](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=199.18) [So our overload that accepts the reference to the Passenger class is simply](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=202.34) [going to call getCheckedBags on that reference and then call the overload](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=206.4) [that we just added that accepts the number of bags.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=209.95) [So by adding this new overload that accepts a reference to the passenger,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=213.04) [we're not adding any new features. We're just making it](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=216.32) [easier for other code to use our Flight class.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=218.8) [Now let's go and add another overload.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=222.44) [This overload will actually accept the number of checked](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=224.15) [bags, as well as the number of carry‑ons.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=226.51) [So in this overload,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=229.74) [what we're going to do is check to make sure the](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=230.73) [number of carry‑ons is two or less, and as long as it is,](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=233.04) [we'll call the overload that accepts the number of bags.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=236.64) [Let's add one more overload.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=239.98) [This one accepts a reference to a passenger, as well as the number of carry‑ons.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=241.61) [And all this overload is going to do is get the number of checked bags](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=245.64) [from the passenger and then call the overload we just created that](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=248.73) [accepts the number of bags and number of carry‑ons.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=251.88) [So as you can see, as we add these overloads, some](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=254.84) [of them are adding new features.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=257.22) [others are just being provided for convenience.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=258.85) [So now in our next clip, let's take a look at some code that uses these new add1Passenger overloads.](https://app.pluralsight.com/course-player?clipId=89a9ebe4-6b67-47a6-ba56-f8f3c1fbb963&startTime=261.69)

[Matching Method Calls to Overloads](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d)

[As you recall,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=0.74) [our Flight class now has five overloads of our add1Passenger method.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=1.51) [So let's take a look at some code that uses these various overloads.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=6.56) [So we'll start out by creating a new instance of our Flight class,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=9.89) [and the first thing we'll do is call add1Passenger, passing in no parameters.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=13.04) [So, of course, because we're passing no parameters,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=17.44) [the compiler is going to use the implementation that accepts no parameters.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=19.77) [So we're using the implementation to simply increment the number of](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=23.44) [passengers. Then we'll call add1Passenger, passing in an integer.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=26.53) [We're passing in a single integer,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=31.14) [so we'll use the overload that accepts a single parameter of type integer.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=32.86) [And then we know internally this implementation will increment the](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=37.24) [total number of checked bags and call the original implementation](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=39.99) [to increment the number of passengers.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=43.53) [So now let's go and create a new instance of our Passenger class.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=46.04) [This newly‑created passenger has a single checked bag.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=49.01) [We'll call add1Passenger, passing in a reference to that passenger,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=52.84) [and then, of course,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=56.84) [we'll call the overload that accepts a reference to our Passenger class.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=57.41) [Internally,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=61.38) [this method gets the number of checked bags for that passenger, calls the](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=62.18) [add1Passenger overload that accepts the number of checked bags,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=65.86) [which internally will call the original add1Passenger](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=68.81) [implementation to increment the number of passengers.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=71.51) [Alright,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=74.94) [so let's take a look at a couple more examples of using our overloads. So](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=75.23) [let's go ahead and create another reference to a passenger.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=78.2) [This passenger has two checked bags,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=80.82) [so we'll call add1Passenger, passing in the reference to the passenger and](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=83.04) [an integer. So that will call our overload that accepts the passenger](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=86.15) [reference and an integer for the number of carry‑ons.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=89.62) [Again, internally,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=92.94) [we call getCheckedBags on the passenger reference, call the](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=94.21) [overload that accepts the number of checked bags,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=97.1) [as well as the number of carry‑ons. Then internally, that will call the overload](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=99.45) [that accepts the number of checked bags, then it finally calls our original](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=103.71) [implementation to increment the number of passengers.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=107.13) [So now let's take a look at one example that's a little bit](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=110.4) [less obvious. Let's declare a variable here of type short](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=112.52) [named threeBags with a value of 3.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=116.03) [Then we call add1Passenger, passing in threeBags and the literal integer 2.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=118.84) [So now, of course,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=124.64) [what the compiler's going to look for at first is an overload of](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=125.17) [add1Passenger that accepts a short and an int. Of course,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=128.21) [the problem is there is no overload that accepts a short and an int.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=132.62) [So when the compiler can't find an exact match,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=137.04) [what it will do is start looking for opportunities](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=139.82) [to do an automatic conversion.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=142.5) [As you recall from our previous course, Getting Started with Programming in Java,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=144.74) [there are a number of scenarios where the compiler](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=148.56) [can do automatic type conversions.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=150.79) [One of those automatic conversions is that a short can be](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=153.24) [automatically converted into an int. So the compiler will look for](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=156.5) [an opportunity to take advantage of that.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=160.18) [So we'll look for an overload that accepts two ints, and we have one of](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=162.44) [those overloads, the overload that accepts the number of checked bags and](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=166.52) [the number of carry‑ons. So the compiler will take care of converting](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=170.7) [that short to an int and making the call for us. Once it makes that call,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=174.1) [everything works just as it does in any other scenario. We know](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=178.42) [internally this implementation will call the overload that accepts the](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=181.34) [number of bags,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=184.46) [which in turn calls our original implementation and increment](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=185.38) [the number of passengers. Alright, so to get a better](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=188.56) [understanding of all this, in our next clip,](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=191.89) [let's jump into our CalcEngine project, and we'll start adding some overloads to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=b28fd7a8-4974-4fec-87c4-e5200237cc1d&startTime=194.11)

[Using Method Overloading in CalcEngine](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d)

[Here we are back in IntelliJ, looking at our CalcEngine,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=0.74) [and what we want to do now is see how we can use method overloads to](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=3.25) [improve the usability of our MathEquation class.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=7.18) [Now as you recall,' earlier in the course,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=9.78) [we added some additional constructors to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=11.72) [Now we still have our default constructor,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=15.2) [but in addition to our to our default constructor,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=16.9) [we have a constructor that allows us to specify the opCode,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=19.27) [as well as the leftVal and the rightVal; and this constructor works well](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=22) [for scenarios where you know the operation in both values right at the](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=25.72) [time we're creating our MathEquation instance.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=29.2) [And then we added another version of the](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=31.74) [constructor that just accepts the opCode,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=33.02) [and this constructor works well for scenarios where you](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=35.39) [know the operation you want to perform, but you don't yet know the values.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=38.01) [But in situations where we want to use this version of the constructor,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=42.14) [the class still has a bit of a challenge,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=45.24) [because when it comes time to actually perform the operation,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=47.54) [it takes a number of steps to be able to do so.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=50.17) [The first thing we have to do is call setLeftVal to set the left value,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=52.69) [then we've got to call setRightVal to set the right value,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=56.2) [and then finally, we get to call execute.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=59.81) [It would be nice to be able to simply provide the left and](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=62.64) [right values as part of calling execute.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=65.11) [So let's add an overload of our execute method that does that.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=68.14) [So we'll start out by declaring it as public void execute,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=71.34) [and this overload of execute will accept two parameters of type double,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=75.98) [we'll name them leftVal and rightVal, and then once we have our parameters,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=79.5) [we can start implementing the overload.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=84.09) [So now the first thing we'll do is assign the leftVal](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=85.19) [parameter to the leftVal field.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=88.5) [Then we'll do the same thing for the rightVal,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=91.74) [and then once we do that,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=94.05) [we'll go ahead and call our original implementation](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=96.17) [of execute; and now that easily,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=98.24) [we have an overload of our execute method that accepts the two values.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=101.72) [So let's head back over to our main class.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=105.49) [So now over here in our main class,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=109.1) [we still have all the code we added earlier in the course.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=110.4) [So I add the code to use our new overload after that existing code.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=113.34) [So now the first thing we'll do here is declare a](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=117.54) [reference of type MathEquation.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=119.55) [Let's equation over lewd.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=121.44) [They will create a new instance of MathEquation initialized with an opCode of d.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=124.64) [So that gives us a MathEquation instance that knows how to perform division.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=130.49) [So after that, let's go ahead and declare two doubles.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=134.64) [So now we have leftDouble with a value of 9.0, and rightDouble a value of 4.0.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=138.84) [So let's go ahead and call our new execute overload, passing in those values,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=144.44) [and then we can go ahead and print out the result.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=149.44) [So now with that, we should be all set.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=152.94) [Let's go ahead and run our code and verify that it behaves as we expect; and](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=154.99) [you'll notice that it does exactly what we wanted it to.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=159.88) [When we divide 9.0 by 4.0, we get this result of 2.25. So our overload is](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=162.05) [working perfectly. So let's go ahead and collapse the window here at the](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=167.75) [bottom, and now let's try something else.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=170.55) [Let's go and declare a couple of integer values.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=174.32) [So now we have leftInt with a value of nine, and righInt with a value of four.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=177.74) [So let's try calling our new execute overload, passing in leftInt and](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=182.54) [rightInt. So now we're calling our execute overload,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=186.21) [passing in ints rather than doubles.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=189.79) [Of course the question is,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=192.47) [is this going to work? Because our overload expects doubles, it doesn't](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=193.61) [expect ints. So let's go ahead and run it, and let's see what happens,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=197.56) [and you'll notice it does indeed work.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=202.74) [It prints out the value, 2.25, and that's because Java was able](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=204.9) [to automatically convert our ints to doubles.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=209.62) [Remember, as we talked about in our previous course,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=212.54) [Java can automatically perform what are known as widening](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=214.89) [conversions, and converting an int to a double is](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=217.64) [considered a widening conversion.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=220.49) [So Java took care of converting those int values into double values.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=222.39) [Then once it had those as double values, was able to](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=226.24) [call our existing execute overload.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=228.66) [You know, something interesting to note,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=231.74) [although the code actually ran and printed out the value](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=233.37) [2.25, that's not necessarily the best answer.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=236.31) [Remember, as we talked about in the previous course,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=240.04) [integer division works differently than floating point division.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=242.27) [When you perform integer division, there's no fractional portion.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=246.04) [So the right answer for dividing the integer 9 by the integer 4 is 2, not 2.25.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=249.68) [But because our MathEquation class is always using doubles,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=256.94) [it performed the operation as a double.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=260.08) [So let's see if we can improve the behavior of our MathEquation class when](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=262.64) [working with ints. So let me again collapse this window here at the bottom](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=266.03) [and I'll switch over to our MathEquation class.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=269.62) [So now we're over here in our MathEquation class, just below](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=272.94) [the execute overload that we just added.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=275.45) [So let's go ahead and declare another overload of execute that accepts](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=277.98) [two ints, and then inside this method we're going to do pretty much](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=280.92) [what we did in the other overload. So we'll assign each of the](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=284.72) [parameters to their appropriate fields, and we'll call our execute method.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=288.08) [So now with that code in place,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=293.2) [we will perform the operation, but the result wouldn't be any different,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=294.82) [because remember, our leftVal and rightVal fields are still doubles,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=298.64) [so the actual operation will be performed using doubles.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=302.54) [So somehow we need to convert the value that's inside a result](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=306.07) [to the appropriate value for an integer.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=308.98) [So let's start out by assigning a result field back to itself. Now of course,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=311.83) [assigning result back to itself isn't really going to change anything.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=317.34) [But here on the right hand side of the equal sign,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=321.34) [let's cast that result to an int, and by doing that,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=322.97) [we can actually get rid of the fractional portion. Remember that](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=327.89) [result as a double, we'll start out with a fractional portion, when](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=330.54) [we cast a result from a double to an int,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=333.86) [the value gets converted to an int, which would then drop the fractional portion.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=336.89) [We then take that int value and assign it back to result,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=341.24) [but now will no longer have the fractional portion.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=344.58) [So adding this one line of code will allow us to display the](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=347.58) [appropriate result for dealing with integers.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=350.3) [So let's go ahead and run our code again, and let's see what happens. So](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=353.39) [you'll notice now, when we run our code with ints,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=357.17) [we get the appropriate answer, which is two,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=359.15) [but the code that uses doubles still shows the right result for doubles,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=361.94) [which is 2.25, and that shows us that our code is using the](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=365) [right overloading each of those situations.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=368.52) [Now, in general,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=371.54) [the way Java goes about choosing the appropriate overload is pretty intuitive.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=372.55) [It's generally going to choose the overload that you expect,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=376.35) [but the underlying logic to make that selection is actually fairly complex.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=379.74) [If you'd like to know the details of how Java goes about making that selection,](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=384.04) [I have a URL on the screen right now that will point you to](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=387.8) [more information about the underlying logic that Java follows](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=391.05) [to select the appropriate overload.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=394.73) [All right, so now in our next clip, let's take a look at variable length parameter lists.](https://app.pluralsight.com/course-player?clipId=bb68ad90-c97b-47c4-b3a2-69dbf523978d&startTime=397.44)

[Variable Number of Parameters](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8)

[Let's look, again, at our Flight class. Now as you recall, up until now,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=0.74) [any time you want to add passengers to our Flight class, we have to](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=4.59) [add them one at a time because the only way we have to add a](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=7.17) [passenger is with our add1Passenger method.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=9.93) [It might be nice to add another method, addPassengers,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=13.04) [that allows us to add multiple passengers at one time.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=16.86) [So we'll declare this addPassengers method to accept an array of type Passenger.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=20.24) [Now, of course,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=25.04) [if we're going to add more than one passenger at a time, we](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=25.55) [need to have another hasSeating method that accepts the number](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=27.72) [of passengers that we want to add.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=30.54) [What our hasSeating method will do will take the current number of](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=32.44) [passengers, add in that count of passengers you want to add,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=35.34) [and as long as we have enough seats for all those](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=39.14) [passengers, we'll go ahead and return true.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=40.64) [So in our addPassengers method, we'll call hasSeating, passing](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=43.48) [in list.length. That's the size of our array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=46.73) [As long as that returns true, we'll increase the number of passengers](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=50.31) [by the length of the array. And as part of adding these passengers,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=53.45) [we want to add in the appropriate number of checked bags. So we can](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=56.4) [loop through the array one by one, and for each passenger, we'll call](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=60.14) [getCheckedBags, adding that to the totalCheckedBags.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=63.86) [So now we have this addPassengers method that will allow us](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=68.14) [to add multiple passengers at one time.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=71.12) [So let's see what it's like to use our addPassengers method.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=75.04) [So we'll create a new instance of our Flight class, and then](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=78.93) [let's say we want to add two passengers, Luisa and John. Luisa](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=81.74) [has one checked bag, John has two.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=85.51) [So we want to call addPassengers on our Flight class, and what we](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=88.49) [want to pass in is the references, luisa and john.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=92.23) [But we can't simply pass these references in because](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=96.24) [addPassengers is expecting an array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=98.78) [So in order to pass those two values in,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=101.77) [we have to explicitly wrap them in a new instance of a passenger array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=103.96) [So then if we want to add three more passengers,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=109.24) [Harish, Julie, and Ashanti, we're going to have to do that same sort](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=111.33) [of thing, call addPassengers on our Flight class.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=115.05) [We can't pass in just harish.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=118.6) [julie, and ashanti.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=120.72) [We, again, have to explicitly wrap them in a new](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=122.34) [instance of our passenger array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=125.22) [So although it's nice that our addPassengers method allows to](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=127.74) [add multiple passengers at one time, it's a bit of a burden to](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=130.78) [use the method because we're always having to create these](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=134.1) [arrays to add the passengers.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=136.81) [Now if we think about what makes our addPassengers method hard to use,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=140.24) [it's the way we've declared the parameter.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=143.75) [We've indicated that we always have to pass the list of passengers as an array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=145.74) [It'd be nice if we could simply declare this method to](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=151.04) [accept a variable number of passenger references, and it](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=153.8) [turns out that we can do that.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=157.57) [The way we do it is by indicating the type of the](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=159.54) [parameter followed by an ellipse.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=162.21) [So where we declare the type of our parameter, rather than explicitly saying](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=165.54) [it's a passenger array, we're going to say Passenger..., and that indicates](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=169.29) [that our method accepts a variable number of passenger references. It will](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=175.58) [accept 0 or more passenger references.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=179.47) [Now any time you want to indicate that the parameter](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=183.64) [accepts a variable number of values,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=185.85) [it has to be the last parameter for the method.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=188.08) [The method is allowed to have other parameters,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=191.25) [but only the last parameter could be variable length.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=193.58) [Now one of the things that's really cool about variable length parameter lists](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=197.04) [is the method itself receives that list as an array So what that means is we](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=200.95) [can implement our addPassengers method just like we did previously because our](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=206.74) [parameter named list is still an array of type Passenger. So we can call](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=210.68) [hasSeating, passing in list.length.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=215.42) [We can still increase the number of passengers by list.length, and we can](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=218.34) [still loop through that list to get the number of checked bags for each](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=222.32) [passenger. Where the convenience of a variable‑linked parameter list comes](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=225.98) [in is for the callers of the method.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=229.69) [So if we look at our code here where we call addPassengers, here we](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=232.2) [call addPassengers, passing in luisa and john,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=235.94) [we no longer need to create an array in order to call the](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=238.7) [method. In the same way, when we call addPassengers for](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=241.49) [Harish, Julie, and Ashanti, again, we no longer need to create that array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=245.37) [We can simply provide the list of values we want to pass into](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=250.34) [the method. So variable‑linked parameter lists are a great](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=253.13) [solution for situations like this.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=256.64) [As implementers of the method,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=259.04) [we have the convenience of working with the values as an array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=260.35) [But for the caller of the method,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=263.64) [all they have to do is pass in the individual values themselves,](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=265.17) [and the compiler takes care of the details of turning those values into an array.](https://app.pluralsight.com/course-player?clipId=1a0d653a-6944-4044-8e17-ce3c2379d8e8&startTime=268.81)

[Summary](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc)

[To wrap up, here are some of the key things you'll want to remember from](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=0.94) [this module. Remember that when we pass objects as parameters those](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=3.95) [objects are passed by‑reference. And what that means is the entire](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=7.82) [object is not copied into the method.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=12.77) [Instead, the method receives a copy of the reference,](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=15.13) [which points back to the original object.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=17.98) [And this has some important implications about changes](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=20.74) [within a method. Remember that if a method tries to make a](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=23.13) [change to the reference itself,](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=25.66) [in other words make that reference point to a different object,](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=27.74) [although those changes are visible inside the method such changes](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=30.25) [would not be visible once we exit the method.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=33.62) [But that's okay because in most cases if a method's going to make modifications](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=36.44) [it's going to make modifications to the referenced object.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=40.75) [And if the method makes changes to the object itself that's fine](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=44.2) [because those changes will remain visible even when we exit the](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=47.55) [method because remember the method is still operating on the](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=51.05) [original object instance.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=54.53) [Then we looked at overloading.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=57.49) [Remember,](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=59.6) [overloading allows a class to have multiple](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=60.01) [versions of a method or a constructor.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=62.39) [Now remember that the way those individual versions are](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=65.32) [identified is by having a signature, and each one must have a unique signature.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=67.11) [Remember that a method's signature is made up of three parts.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=72.54) [It's the name of the method,](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=75.94) [along with the number of parameters it has, as well as](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=77.62) [the data type of each parameter.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=80.61) [So as long as that combination of name,](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=83.04) [number of parameters, and type of each parameter's unique the](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=85.67) [method is considered to have a unique signature.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=88.82) [Then we finished up.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=92.89) [We looked at variable length parameter lists.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=94.07) [Remember that variable length parameter lists allows the caller](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=96.8) [of the method to pass in 0 or more values.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=99.35) [The way we indicate that the parameter can accept multiple values is](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=102.89) [by placing an ellipse after the parameter type.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=106.53) [Remember that only the last parameter of a method is](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=109.64) [allowed to be variable length.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=112.32) [Now remember that all the values that are passed are received into an array.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=114.41) [So within the method itself all we have to do to process the values is process](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=118.94) [the contents of that array. Alright, that wraps up this module. In our next module we'll start looking at class inheritance.](https://app.pluralsight.com/course-player?clipId=4eb12520-59e8-4d53-8914-ecbf43119efc&startTime=122.98)

[Class Inheritance](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715)

[Introduction](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715)

[Welcome to our next module, Class Inheritance.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=0.74) [This is part of the Pluralsight course,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=3.67) [Working with Classes and Interfaces in Java,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=5.21) [and my name is Jim Wilson. Now as we design classes](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=7.36) [for application, those classes, of course,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=11.47) [are going to have functionality and features. And](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=14.07) [oftentimes, we may want to design other classes that](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=16.52) [leverage those existing classes and features,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=19.4) [and that's where class inheritance comes in.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=22.24) [So in this module, we'll start out with an overview of inheritance.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=24.74) [We'll see how the leverage inheritance and declare](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=28.64) [one class to inherit from another.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=30.66) [Now when using inheritance,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=32.98) [we have what's known as the derived class and the base class. The](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=34.27) [derived class inherits from its base class.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=37.85) [Obviously, there's going to be a close relationship between those two classes.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=40.94) [So the next thing we want to understand is just what is that](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=44.54) [relationship between the derived class and its base class?](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=46.77) [Now as we declare our classes,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=50.44) [those classes have members. And in some cases, a derived](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=51.88) [class may hide the members of its base class.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=55.4) [In other cases, it may override members in its base class.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=58.61) [So we want to understand each of the situations and how they affect our code,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=62.14) [Then the next thing we'll look at is Java's Object class, and the](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=66.14) [Object class plays a really important role in class inheritance.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=69.79) [So we'll take a look at just what the role of the Object class is](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=73.22) [and how we can use it.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=76.09) [And then we'll finish up with a look at equality checks.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=77.74) [And the question is, if you have two instances of a class,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=80.84) [what does it mean for those instances to be equal?](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=83.63) [Well, the answer is, it kind of depends,](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=86.34) [So we'll finish up this module by understanding the behavior of equality checks and how we can implement those equality checks.](https://app.pluralsight.com/course-player?clipId=b7ac048f-009e-4c41-a35f-77f05e34b715&startTime=88.74)

[Getting Started with Inheritance](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a)

[As we mentioned, Java allows one class to inherit from another.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=0.74) [So the way we do that is by using the extends keyword in our class declaration.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=5.34) [So we'll declare the name of our new class and then](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=9.94) [say it extends its base class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=11.88) [Now this new class, what we call the derived class,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=14.74) [will have the characteristics of its base class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=17.13) [So we'll start out with the same features and functionalities](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=20.34) [that are contained within the base class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=23.11) [But then this derived class can add specialization.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=25.74) [So it can build on those inherent features, then add custom features of its own.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=28.84) [So let's see what this looks like in code.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=34.48) [Let's declare a new class called our CargoFlight class and we want it to](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=36.24) [inherit from our flight class so we'll say the CargoFlight class extends our](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=39.91) [flight class. Now CargoFlight can add its own members.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=44.46) [So the first thing we'll do is have a field here, maxCargoSpace.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=48.06) [So in this case,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=51.68) [we're seeing our CargoFlight class has 1000 cubic meters of cargo space.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=52.41) [Now,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=57.04) [in addition to our maxCargoSpace, we're going to have to](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=57.23) [track how much of that space we've used.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=59.16) [So as a CargoFlight class, we need a way to add cargo to a flight. So](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=62.04) [we'll have a public method here, add1Package, that accepts the height,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=65.9) [width, and depth of the packages being added.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=70.57) [So the first thing we'll do inside this method is calculate the overall size](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=74.14) [of the package by multiplying those three values together, and before we add](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=77.81) [the cargo, we want to make sure we have space for it. So we'll call another](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=81.76) [method we'll declare in our CargoFlight class called hasCargoSpace. And this](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=85) [will return back true or false, indicating whether there's space. If there](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=87.96) [is space,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=93.01) [we'll go ahead and add the size of that package to the amount of cargo space](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=93.67) [we've already used, and if there is not space, we'll call another method in](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=96.76) [our CargoFlight class called handleNoSpace.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=100.55) [Now let's take a look at those two methods. So we have our hasCargo Space](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=104.14) [method; it accepts the size of the package that we want to add.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=107.67) [So all we do inside of here is take the size of that package, add](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=110.94) [it to the amount of space we've already used,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=114.17) [as long as the result of adding those two together is less than or](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=116.12) [equal to our maxCargoSpace, it will return true. And then in the](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=119.06) [case of our handleNoSpace method, we can print out the message](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=122.45) [saying there's not enough space.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=125.64) [So let's see what it's like to use our CargoFlight class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=128.24) [Now just as a reminder, our CargoFlight class extends our Flight class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=130.99) [We have just one public method here, add one package.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=134.84) [So we're going to use our CargoFlight class just like we use other](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=138.34) [classes. We'll create a new instance of our CargoFlight class, and](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=140.98) [we'll assign that to a reference of type CargiFlight named cf. Now](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=144.47) [once we have that reference, we'd of course call our add1Package message,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=148.29) [which allows us to add a package to this instance of our CargoFlight class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=152.54) [So in addition to having the ability to add a package to this flight,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=156.64) [we can also do things that the Flight class allows.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=160.2) [So we declare an instance of a passenger, we can then use](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=162.84) [our cf reference to call the add1Passenger method that was](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=166.01) [declared in the Flight class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=169.57) [So, as you can see,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=172.04) [our CargoFlight class has all the features and capabilities that](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=172.62) [we've declared as part of the CargoFlight class,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=175.89) [but it also has the features and capabilities that](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=178.32) [were declared in the Flight class.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=181.04) [So now in our next clip,](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=183.84) [let's see other type of reference that we use affects the available features we can access.](https://app.pluralsight.com/course-player?clipId=619859b9-1f43-46e0-909a-9d8c7c1eaa8a&startTime=185.16)

[References to Derived Class Instances](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034)

[Now when we create an instance of a class, we're generally going to want to](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=0.74) [store a reference to that class instance. And when we're dealing with](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=3.42) [classes that derive from another class, we have some interesting options](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=6.97) [because we can actually assign an instance of that class to a reference](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=11.24) [whose type is that of the base class.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=15.19) [Now this is a really powerful concept,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=17.84) [but it does have some implications in terms of what features](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=19.81) [are available because the features we can access are dictated](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=22.67) [not by the class instance type, but instead are dictated by the reference type.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=26.49) [We'll only be able to access members that are visible to](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=31.64) [the type of the declared reference.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=34.33) [So let's go ahead and create an instance of our CargoFlight class, but](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=38.04) [let's assign it to a reference whose type is flight.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=40.98) [So the type of our variable here, f, is the Flight class,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=44.64) [even though it's pointing to an instance of CargoFlight,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=47.58) [and that's valid because CargoFlight inherits from Flight. Now we,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=50.34) [of course,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=54.74) [can use any flight‑based features here. So I want to go ahead and create](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=55.1) [an instance of a passenger. I can use my f reference to call](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=58.34) [add1Passenger to add that passenger to the flight.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=61.74) [And, of course,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=65.38) [this works because the method add1Passenger is](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=65.79) [declared as part of our Flight class.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=68.07) [But what if we try to call f.add1Package?](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=70.84) [We're dealing with an instance of our CargoFlight class,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=74.44) [and the CargoFlight class has a method add1Package,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=76.95) [but the Flight class doesn't.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=80.04) [So the attempt to call add1Package with a reference whose type is](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=82.34) [Flight would actually generate an error because the methods we can](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=85.66) [access are tied to the type of the reference,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=89.41) [not the type of the class instance.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=92.72) [And although this may sound like a bit of a limitation,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=95.49) [it's actually a really powerful concept because it makes it easy](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=97.65) [for us to group work that's tied to the base class without regard](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=101.07) [for any specific derive classes.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=104.88) [So let's say we need to process a number of flights. So we declare an array](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=107.24) [here, squadron of type Flight with 5 slots in. And for our 0 element, we're](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=110.98) [going to assign that an instance of the Flight class.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=116.23) [But then for our next element, we'll assign it an instance of the](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=119.34) [CargoFlight class, and we can keep doing this. Now we could](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=122.37) [actually take this array and pass it the code that doesn't know](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=126.87) [anything about the CargoFlight class.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=130.05) [You could loop through the array, do any flight base work that was](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=132.24) [necessary without regard for the specific kind of flight it is.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=135.37) [As long as it is the Flight class or something that derives from Flight class,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=139.19) [the code will be able to process this array.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=142.64) [Now for those elements that are actually cargo flights, we](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=144.89) [have the option to do additional work,](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=147.53) [but we could easily group all the flight base work](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=149.64) [together here in this single array.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=151.68) [Alright, so now in our next clip, let's take a look at an issue known as member hiding.](https://app.pluralsight.com/course-player?clipId=a12268e7-a1a0-4352-aeb5-8c7d793fc034&startTime=154.34)

[Field Hiding](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1)

[Now as we declare our classes, those classes, of course, have members.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=0.64) [And when we're dealing with inheritance,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=4.49) [we have the members in the base class as well as any](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=5.95) [members we add in the derived class.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=8.17) [And in general,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=10.74) [the members of the base and the derived class blend together in](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=11.89) [a way that's pretty natural, but there are so special](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=15.1) [situations that we want to be aware of.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=17.56) [One of those relates to fields because there's a scenario where a field](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=19.74) [in a derived class can hide a field in the base class.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=23.74) [If your derived class declares a field that has the same name as a field](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=27.64) [in the base class, the derived class doesn't replace the base class](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=32.13) [field, the base class field is still there,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=36.37) [but the derived class field is hiding the field from the base class.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=38.72) [Now just a quick reminder, we have our Flight class. Now as](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=44.01) [you recall, our flight class has a field named seats that we](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=47.17) [initialize with a value of 150.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=49.93) [Our Flight class also has a method add1Passenger.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=52.54) [This add1Passenger method wants to increment the number of passengers, as long](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=55.69) [as this method hasSeating returns a value of true. Remember, hasSeating method](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=59.27) [is pretty straightforward, it just checks to see if the number of passengers is](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=63.83) [less than the value in our seats field.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=68.03) [So let's say when we declare our cargoFlight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=71.54) [we also declare a field named seats with the value of 12 and we know our](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=74.02) [cargoFlight class extends the Flight class and the Flight class already](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=78.45) [has a seats field whose value is 150 and this method hasSeating depends](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=82.63) [on the value in that seats field.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=88.14) [So our cargoFlight class' seats field is hiding the seats](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=90.24) [field in our Flight class and that seats field is an](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=94.01) [important part of our hasSeating method.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=96.98) [So let's see what happens when we start using these classes in code.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=100.54) [So let's create an instance of our Flight class, we'll assign it to](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=103.69) [a reference F1 whose type is flight.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=106.39) [So if we now print out the value of F1.seats,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=109.94) [we get exactly what we expect, the value 150 because that's the](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=112.87) [value we set seats to in our Flight class.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=116.69) [Then let's create an instance of our cargoFlight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=119.34) [we'll assign that to a cargoFlight reference named CM. So](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=121.6) [if we print out cf.seats, again,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=125.57) [we get what we expect, the value 12, the value we set](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=127.8) [seats to in our cargoFlight class.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=131.01) [So at this point, things look pretty natural.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=133.74) [It seems that the seats field in our cargoFlight class has replaced the seats](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=136.29) [field in our Flight class, but that's actually not the case.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=141.43) [The seats field in cargoFlight is simply hiding the](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=145.24) [seats field in our Flight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=148.33) [and we can verify that that's the case. Let's go ahead and create another](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=150.5) [instance of cargoFlight, but this time we're going to assign it to a reference](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=153.47) [named F2 whose type is flight. So we have an instance of our cargoFlight class](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=157.22) [being accessed by a flight reference.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=162.76) [So when we print out F2.seats, we don't get the value 12,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=165.58) [we get the value 150 because this code, which is using a flight reference,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=169.64) [is using the values of the seats field as it's defined in the Flight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=174.9) [even though the type of our instance cargoFlight has](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=179.34) [declared a field with that same name,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=181.97) [and this can create some really ugly problems.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=184.24) [For example, if we call f2.add1Passenger, remember,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=186.94) [the type of F2 is flight, but it points to an instance of cargoFlight.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=190.79) [So when we call add1Passenger,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=195.24) [which is implemented in our Flight class, it uses the method hasSeating,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=197.07) [which is also implemented in the Flight class. So when hasSeating](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=201.17) [accesses the field, seats, it's accessing the version of seats](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=205.01) [that's declared in our Flight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=208.77) [so it's going to use the value 150. So even though our cargoFlight has](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=211.14) [set a field named seats to a value of 12, our hasSeating method is using](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=214.83) [a version of the seats field that was set to 150, and this problem is not](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=219.2) [tied to the type of reference we use because even if we use](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=224.07) [cf.add1Passenger, remember that CF is a reference of type cargoFlight](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=227.92) [that points the cargo flight.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=233.51) [When it calls add1Passenger, which is implemented in our Flight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=235.34) [which then calls hasSeating,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=239.05) [which is implemented in the Flight class, it's still going to use the version](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=240.63) [of the seats field that's defined within the Flight class,](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=244.27) [which means it's going to use that value of 150, so](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=247.4) [obviously, this is a very concerning situation.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=250.39) [So in our next clip, let's see a better solution to dealing with this kind of issue.](https://app.pluralsight.com/course-player?clipId=1e3776fd-f2a9-481e-b259-69891034c7c1&startTime=253.04)

[Method Overriding](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71)

[Now, as we've seen,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=0.74) [if a derived class declares a field with the same](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=1.5) [name as a field in the base class,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=4.37) [the field in the derived class hides the field in the base class,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=5.76) [but it turns out when it comes to methods,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=9.42) [things behave very differently.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=11.69) [In the case of methods,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=13.49) [if the derived class declares a method with the same](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=14.48) [signature as a method in the base class,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=17.93) [the derived class overrides the base class method.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=20.64) [In other words,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=23.76) [the method in the derived class replaces the method in the base class.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=24.65) [So let's see how we can use this behavior to improve](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=30.54) [our Flight and CargoFlight classes.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=32.73) [So as you recall,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=35.34) [our Flight class has a field here named seats with a value of 150.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=35.98) [Well, let's change this from being a field to a method named getSeats,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=40.04) [which returns 150.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=44.06) [Then down in our hasSeating method,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=45.6) [we'll change that to use our getSeats method rather than our seats field,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=47.67) [and then over in our CargoFlight class,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=52.64) [we'll override the getSeats method and instead have that return the value of 12.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=54.28) [So now our Flight class declares a method named getSeats,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=59.54) [and our CargoFlight class overrides that method.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=62.7) [So now let's see how this change impacts the behavior of our classes.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=67.54) [So we'll start out by creating an instance of our Flight class,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=71.08) [and we'll assign it to a reference named f1 whose type is Flight.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=74.17) [So if we now print out f1.getSeats, we get the value we expect, which is 150.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=78.74) [If we then create an instance of our CargoFlight class,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=84.42) [assigning that to a reference named cf whose type is CargoFlight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=87.61) [if we print out cf.getSeats, again, we get what we expect,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=90.8) [which is the value of 12.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=94.85) [Now so far, our behavior is very much like that of when we were using fields.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=96.25) [But now let's take a look at this example.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=101.08) [Let's create an instance of CargoFlight and assign it to](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=103.15) [a reference f2 whose type is Flight.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=106.13) [And then we'll print out f2.getSeats.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=108.44) [So now in this case, we're using a reference whose type is Flight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=112.44) [pointing to an instance whose type is CargoFlight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=115.94) [and in this case, now we get the value 12.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=118.94) [Notice that we're using the implementation of getSeats](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=121.72) [that's tied to the class instance type, not the reference type,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=125.47) [and that's a key difference between methods and fields.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=130.44) [In the case of fields,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=133.57) [it's the type of the reference that determines](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=135.11) [which version of the field you use, but when it comes to methods,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=137.58) [it's the type of the instance that determines which method you use.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=141.77) [So here, in this case, even though the type of our reference is Flight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=146.54) [because the instance is a CargoFlight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=150.37) [we use the implementation of getSeats provided by the CargoFlight.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=152.94) [And this is a really powerful concept.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=156.87) [We can see that when we do something like cf.add1Passenger.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=159.28) [Remember that cf's type is CargoFlight, pointing to an instance of CargoFlight.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=164.27) [We call add1Passenger, which is implemented in the base class Flight.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=168.03) [It then calls hasSeating, which is also implemented in the Flight class.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=173.44) [And remember that hasSeating calls the getSeats method.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=177.74) [Well, because this is an instance of CargoFlight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=180.38) [when hasSeating calls getSeats,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=183.49) [it calls the implementation of getSeats within CargoFlight,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=185.55) [which means the hasSeating method will use the value 12.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=189.68) [So even though the hasSeating method is implemented in the](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=192.88) [Flight class, it knew to use to the getSeats implementation](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=195.41) [from the CargoFlight class.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=199.48) [And this is true no matter what kind of reference we use.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=201.44) [So if we say f2.add1Passenger, again, remember,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=205.14) [the f2's is Flight, but it points to an instance of CargoFlight.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=208.83) [So when we call add1Passenger, it calls hasSeating, and again, because this is](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=213.54) [an instance of CargoFlight, when hasSeating calls getSeats,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=218.59) [it calls getSeats from CargoFlight, which again means that hasSeating](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=222.19) [uses that value of 12 when doing the comparison.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=227.11) [So to help us get a better understanding of all this, in our next clip,](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=230.94) [let's get back into IntelliJ, and we'll see how we can use inheritance to improve our calc engine project.](https://app.pluralsight.com/course-player?clipId=3b385f51-599b-4b49-b578-d45967a08b71&startTime=233.97)

[Improving CalcEngine with Inheritance](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417)

[Here we are in IntelliJ, again, looking at our CalcEngine project. And what we](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=0.64) [want to do now is see how we can use inheritance to simplify our application and](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=4.86) [actually make maintaining the code a bit easier.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=9.77) [And as you recall,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=12.84) [the bulk of the work in our CalcEngine project is done with](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=13.37) [this MathEquation class. It gives us the ability to do the](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=16.48) [four basic math operations.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=19.81) [So now to do that, the MathEquation class has some fields here.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=22.14) [So we have our leftVal and rightVal fields, which are both doubles.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=25.64) [We have a result field, which is a double, then we also have this opCode,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=29.14) [which is a char. Remember, the opCode indicates](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=33.18) [what operation we want to perform.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=36.37) [So using that opCode,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=38.54) [the bulk of the work is actually done in the MathEquation class' execute](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=40.34) [method. So let's scroll down and take a look at that.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=44.17) [So here we are now at our execute method. And as you recall, the](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=48.34) [bulk of the work in the execute method is this switch statement. We](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=51.04) [need to have this switch statement because this execute method](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=54.88) [doesn't really do just one thing.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=57.58) [It actually does four different things.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=60.04) [And so based on the opCode, the execute method has to choose which of those](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=62.64) [things to do and then perform the appropriate operation.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=65.88) [And although there's nothing wrong with this approach,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=69.24) [but as we'll see by using inheritance,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=71.34) [we can actually allow the JavaType system to make some decisions for us,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=73.63) [which will allow us to actually write code that's much simpler to maintain.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=77.28) [So to do that,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=81.74) [let's start out by creating a new class that'll service](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=82.7) [the base class for our calculations.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=85.02) [So to create that class,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=87.44) [I'll go over here to the project window. I'll, again,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=88.92) [right‑click on our package name.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=91.05) [I'll choose New, and then we'll go and say Java Class. For the](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=93.84) [name of this class, let's call it CalculateBase. So to create our](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=99.13) [CalculateBase class, I'll go ahead and hit Enter. So that gives](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=104.13) [us our CalculateBase class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=108.06) [So what we'll want to do here now is add in some of the fields that we'll need.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=109.94) [So let's add fields for our leftVal, rightVal, and result.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=113.64) [So now we have those three fields in place,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=118.74) [and we've marked them all as private. And, remember, by](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=120.55) [marking them as private, it means those fields are not](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=123.41) [accessible outside this class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=126.16) [So that means that in addition to the field, we'll also need to](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=128.34) [have getters and setters for each of those fields.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=131.25) [So now we have our getters and setters in place, so let's go and](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=135.04) [add a method we'll use to do the calculations.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=137.99) [So let's create a public void method named calculate. So that](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=140.85) [gives us our calculate method, but we're not going to do any](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=145.41) [actual calculations in this class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=148.37) [We're going to actually have classes that derive from](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=150.99) [this that will handle those details.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=152.8) [So let's add another class to our project named Adder. So, again,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=155.34) [over here in our project window,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=159.3) [we'll go up to the package name, and we'll right‑click,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=160.94) [choose New, and then we'll choose Java Class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=163.54) [So this class will be used to do our additions, so I'll name this class Adder.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=167.68) [Then I'll go ahead and hit Enter.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=173.74) [Now we want our Adder class to extend CalculateBase.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=176.14) [So that means that Adder will have all the characteristics of CalculateBase.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=181.14) [So all we need to do is incorporate the specializations](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=185.64) [that we need for this Adder class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=188.76) [What we want to do is specialize the calculate method. So](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=191.04) [let's, again, declare our calculate method.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=194.45) [Now, remember,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=197.84) [we want this calculate method to override the calculate method in CalculateBase.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=198.55) [So let's go ahead and mark it with the @Override annotation.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=203.44) [So now the work we want to do in this calculate method is really, really simple.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=207.94) [We simply want to add our left value and right values together.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=211.82) [So let's start out by declaring a local variable named value of type double,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=215.54) [and we want to add the left values and right values together.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=221.04) [But remember, the leftVal and rightVal fields are private in CalculateBase,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=224.6) [so we can't use the field directly, but we can use the getters.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=229.14) [So now we get our left and right values,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=233.84) [add them together, and then store the sum and value or local variable.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=235.85) [So now what we want to do is set the result field, so](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=240.34) [we'll call setResult, passing in value, and that's all there is to it.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=243.26) [The Adder class is now a specialization of CalculateBase that knows how](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=248.57) [to handle the details of adding two values together.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=252.44) [So I'm going to add three more classes to our project.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=255.74) [I'll add Subtractor, Divider, and Multiplier. So now we have](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=258.24) [these four classes that all inherit from CalculateBase and](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=263.24) [do specific math operations.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=266.68) [So let's see how we can use these classes.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=269.11) [So let's head over here to our Main class. And just after our Main](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=271.14) [class' main method, let's add a method named doCalculation. Now](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=275.01) [the first parameter to doCalculation will be a reference to](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=280.4) [CalculateBase named calculation.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=283.42) [And then we'll also pass in a leftVal and a rightVal. And](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=287.54) [then here inside of doCalculation, we want to do the work it](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=292.22) [takes to perform a calculation.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=295.22) [So the first thing we'll do is use our calculation reference and set its](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=297.44) [leftVal and rightVal. And then once we set those two values, we're going](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=301.15) [to want to perform the calculation, so we'll call the calculate method.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=306.01) [And then once we calculate, we'll go ahead and print out the result.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=310.19) [So now as we look at this code,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=315.04) [it has all the steps we need to perform a calculation, but](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=316.47) [there's one thing that's kind of curious.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=320.32) [When we call calculate on CalculateBase, we know it](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=322.14) [doesn't actually do anything.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=325.4) [The implementation of calculate on CalculateBase was just an empty method,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=327.24) [but that's okay because, remember,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=331.44) [there are classes that inherit from CalculateBase to](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=332.98) [actually override that method,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=335.47) [and we can actually use instances of those classes.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=337.34) [So let's head up here to our main method,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=340.64) [and we'll add some code that does that.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=342.26) [So let's start out by declaring an instance of our Divider class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=344.24) [And then once we have our instance of divider, let's call doCalculation,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=349.38) [passing in divider and then two values, 100 and 50.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=353.29) [So what happens now is we know that our Divider class overrides the calculate](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=358.64) [method to perform division. So when we call doCalculation,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=362.71) [passing in divider,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=366.6) [even though doCalculation receives that into a CalculateBase](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=367.97) [reference, when that reference calls calculate, it will use the](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=371.49) [divider implementation. Therefore,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=375.81) [we should print out the result of dividing those two values together.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=377.9) [So let's do one more of these kinds of calculations. Let's go](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=381.5) [ahead and create an instance of the Adder class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=384.3) [Then we'll call doCalculation,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=387.84) [passing in Adder along with the two values, 25 and 92.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=389.5) [So now we have our code in place that will do calculations using a divider](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=395.44) [and using our adder. Let's go ahead and run our code, and let's see what](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=399.33) [happens. If we look down here at our Run window,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=402.45) [we can see our results.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=405.96) [Well,](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=407.64) [the first calculation we performed was dividing 100 by](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=407.77) [50, and we can see our result is 2.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=410.92) [So our Divider class worked.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=413.84) [Then our second calculation was adding 25 to 92, and we could see](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=415.88) [the result is 117. So our code is using inheritance to achieve the](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=420.01) [exact result we were looking for. Alright, so now in our next clip, let's take a look at the Java Object class.](https://app.pluralsight.com/course-player?clipId=75f47b95-e3ec-49ed-a9d2-7cd8983e6417&startTime=424.35)

[Object Class](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1)

[Let's take a look now at the Java Object class. The Java Object class is](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=0.74) [really important when we talk about inheritance because the Object class](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=5.11) [is the root of the Java class hierarchy.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=9.09) [Every class we declare has the characteristics of object,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=11.94) [and this is really important when it comes to talking about references](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=16.14) [because an object reference can reference any class instance.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=18.99) [In fact,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=23.57) [an object reference can even reference an array instance because in Java,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=23.83) [arrays are a type of class. And the thing to keep in mind,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=28.89) [when we're talking about inheritance, every class in Java inherits](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=34.06) [either directly or indirectly from the Object class.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=37.98) [So let's take a look at some of the classes that we've declared.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=42.24) [Now, of course, we've talked a lot about our Flight class.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=44.85) [Well, if we were diagramming the classes we've created for our Flight class,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=47.54) [we might just draw a box like this.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=51.12) [And then when we get to our CargoFlight class,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=53.84) [our CargoFlight class extends the Flight class,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=56.04) [meaning that our CargoFlight class inherits from the Flight class.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=59.44) [So to diagram this, we would draw a box for our CargoFlight and](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=63.04) [then draw an arrow up to the Flight class, and that indicates](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=67.04) [that CargoFlight inherits from our Flight class. Now looking,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=70.68) [again, at our Flight class,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=74.39) [our Flight class doesn't appear to extend any other class.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=75.73) [But in Java, if a class doesn't explicitly extend another class,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=80.24) [then it implicitly extends the Object class.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=85.14) [So as we're diagramming our class hierarchy here,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=88.74) [we should indicate that our Flight class is inheriting from](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=91.46) [Object. Now as part of declaring our Flight class, we could](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=94.26) [explicitly write extends Object.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=98.12) [But there's really no reason to do that because by not](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=100.94) [indicating that we extend anything,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=103.69) [we automatically extend the Object class, and this](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=105.54) [is true for any class we declare.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=108.94) [So when we declare our Passenger class, we don't](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=111.14) [indicate that it extends any class, so it automatically extends the Object class.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=113.66) [So this shows us that every class inherits from the Object class,](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=119.28) [and this has important implications when it comes to](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=122.99) [dealing with object references. We'll take a look at that in our next clip.](https://app.pluralsight.com/course-player?clipId=6f91499a-d725-413a-81b0-8da3f61b81c1&startTime=125.1)

[Object References](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9)

[Now as we mentioned,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=0.64) [every class inherits either directly or indirectly from the Object class,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=1.42) [and that has important implications when it comes to object references,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=6.14) [because it means a reference of type Object can point to any class](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=9.44) [instance. And this is useful in scenarios where you might need to hold on](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=13.22) [to some otherwise unrelated types.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=17.68) [So for example,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=20.24) [if we declare an Object array here named stuff and we allocate out](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=21.27) [three slots in the array, well, stuff[0] might point to an instance](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=25.31) [of the Flight class, stuff[1] might point to an instance of our](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=29.7) [Passenger class, and stuff[2] might point to an instance of our](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=33.47) [CargoFlight class.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=37.24) [Now when working with our stuff array,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=38.84) [we couldn't deal with any of the type‑specific capabilities of any of these](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=40.56) [classes, but we are able to hold on to the references, and even pass them](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=44.05) [around to methods or other parts of our code.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=47.99) [And object references give us a great deal of versatility,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=51.44) [so let's go ahead and declare another Object reference here named o,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=54.74) [and we'll point it to an instance of our Passenger class. Now](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=57.62) [because the type of o is Object, we can actually assign an](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=60.86) [entirely different type to it.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=64.18) [So we can say o = and create a brand new instance of a Flight array. And notice](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=66.04) [what we're doing here. We've created an array named Flight,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=70.6) [but we're assigning it to a variable o that itself is not an array.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=74.64) [It's simply an object reference.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=78.75) [And a key thing to understand here is that in Java,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=80.64) [the array itself is considered a type of class.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=83.5) [So the Flight array is a type of class,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=86.84) [and that array holds references to another type of class named Flight.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=90.04) [So looking at these two pieces of code, on the left,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=94.74) [we have an array whose type is Object,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=97.47) [which each element in the array pointing to an instance](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=99.57) [of a particular class. On the right, we have this variable o,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=102.39) [whose type is Object, that holds a reference to a class, which happens to](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=106.33) [be an array. But now something to keep in mind, although an object](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=110.38) [reference can point to an instance of any class type,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=114.98) [that object reference can't access the capabilities of that class type.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=118.51) [So for example, say we created a variable here named o,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=123.04) [whose type is Object, and we assign to it a reference to](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=126.08) [an instance of our CargoFlight class.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=129.37) [So that means we have that CargoFlight instance with a](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=131.54) [reference to that instance stored in o.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=133.69) [Now we know that the CargoFlight class has a method named add1Package.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=136.84) [Well, if we try to say o.add1Package, even though](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=141.54) [CargoFlight exposes that method,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=145.29) [remember we can only access features that the reference itself knows about.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=147.5) [And since the object reference doesn't know anything about the](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=151.94) [features in the CargoFlight class, we would actually get a](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=154.23) [compilation error when we try to call add1Package.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=156.66) [Now this doesn't mean that CargoFlight doesn't have all these capabilities.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=160.34) [It simply means that the reference named o doesn't know how](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=163.99) [to access the CargoFlight capabilities.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=166.61) [So what we could do is actually assign that reference o over to](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=169.44) [a reference named cf of type CargoFlight.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=173.39) [Now this line, as it's written here, wouldn't actually compile. Remember that](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=176.48) [the type of o is Object, and it can point to any class type.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=180.19) [So the compiler can't be sure that o will always](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=184.37) [point to an instance of CargoFlight.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=187.57) [So to deal with that, we need to include a cast in this assignment. And](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=190.24) [basically here, what we're telling the compiler is that trust me,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=194.34) [I'm sure that o points to an instance of CargoFlight, so go ahead and](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=198.09) [assign that reference over to cf. So by doing that,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=202.14) [we would allocate out the reference named cf, then we would take](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=205.71) [the reference that's held in o, pass that over into cf, and now cf](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=209.02) [holds a reference to that same instance, which means that now we](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=212.95) [can call cf.add1Package, and that code would work just fine.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=216.79) [But something to be very careful about, the line where we assign o to cf, if](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=220.94) [o happens to point to something that's not a CargoFlight,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=225.83) [we would actually crash at runtime.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=229.24) [So in order to program effectively,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=231.64) [we should verify that o actually holds a reference to CargoFlight.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=233.49) [And we can do that with an if statement that uses the instanceof operator.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=237.74) [What the instanceof operator does is it takes a look at the reference](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=242.34) [that o holds, and it makes sure that the object it points to is an](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=246) [instanceof CargoFlight. If that's not the case, instanceof will return](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=250.35) [false, and we would skip over the code that actually assigns o to cf.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=254.14) [Now as we've mentioned,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=258.41) [a reference of type object can't access any of the](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=260.85) [specific behaviors of our class types.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=263.93) [But there are some behaviors that are common to all classes because](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=266.44) [the Object class does expose certain methods.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=270.02) [For example,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=273.24) [one of the methods we have is what's called the clone method,](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=273.74) [and this allows us to create a new instance of an object that's](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=276.34) [a duplicate of the current instance.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=278.95) [There's a method called hashCode that returns back a](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=281.7) [hash code for a current instance.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=284.02) [And this is useful for scenarios that require a hash code, things like](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=286.24) [hash maps that we use to store collections of objects.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=289.48) [There's a method getClass that will return type information for the](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=293.12) [current instance. There's a method called finalize that we don't really](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=296.26) [override very often, but this handles special resource cleanup scenarios](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=300.01) [that occur in some very specific scenarios.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=304.23) [There's a toString method that will return back a string](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=307.57) [representation of the current instance. And then finally, we have](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=310.09) [the equals method. And the equals method allows us to compare two](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=314.15) [instances to see if they're actually equal. And we'll take a closer look at equals in our next clip.](https://app.pluralsight.com/course-player?clipId=ec656e5d-6c17-4d38-bd46-a82f16bb16d9&startTime=318)

[Equality](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02)

[Let's take a look now at the issue of equality.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=0.7) [If I have two object references,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=3.64) [what does it mean for those references to be equal?](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=5.64) [And the answer is, well,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=8.84) [it kind of depends. It depends on specifically what you're looking for in terms](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=10.51) [of being equal. So to demonstrate, let's start out by creating a new instance](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=14.7) [of our Flight class and setting a flightNumber of 175. We'll reference that](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=19.04) [instance in a variable f1. Then we'll create another instance that also has a](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=24.01) [flight number of 175, and we'll assign that to a reference named f2. So now the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=28.57) [first question is, how do I compare them for equality?](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=34.46) [Well,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=37.94) [one option would be to use the == operator. When we](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=38.1) [compare primitive types such as integers,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=42.28) [that's the operator we use for equality.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=44.46) [So what happens if we use this with references? In Java, if you](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=46.46) [use the == operator with references, what it actually does is](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=50.14) [compare the references themselves.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=53.96) [So it looks in each reference and says,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=56.44) [do both of these references point to the exact same object instance?](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=58.52) [And, of course,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=63.7) [in our case they don't. They point to two separate objects.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=64.11) [So this comparison will result in false.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=66.89) [So now what's our next option?](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=70.74) [Well, remember that every class inherits from the Object class.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=72.31) [And we said the Object class has an equals method,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=76.44) [so maybe we can use that method.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=79.54) [So I'll do a logical test using f1.equals, passing in f2. But, of course,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=81.54) [the problem here is that the full implementation of the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=87.16) [equals method that we inherit from the Object class does the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=90.46) [same comparison as the == operator.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=93.57) [So, again,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=96.74) [it checks to see if both references point to the exact same object instance.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=97.23) [So, again, we would get back false.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=102.25) [And the issue here is that we're always looking at the references.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=105.17) [In general, when working with class types, when we do equality comparisons,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=108.54) [we actually look at the contents of the objects.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=113) [We want to look at the values within each object instance to](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=116.04) [determine if those instances are equal.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=118.66) [And the good news is we can do that.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=121.41) [So looking at our Flight class, we know we have a number of fields,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=124.69) [but two of those fields were linked to identifying a flight.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=128.24) [If you remember in one of our earlier modules,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=131.74) [we said we could identify a flight in one of two ways,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=133.54) [either by using the flight's flightNumber or the flight's Flight class.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=136.64) [We set one or the other, but not both.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=141.14) [So if you create a flight with a flightNumber of 175, the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=143.84) [flightNumber field will have that value 175, Flight class](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=146.85) [will still have its default value.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=151.03) [So one definition of equals we might use for the Flight class](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=152.9) [is to compare these identifying fields. So to do that, we'll](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=155.97) [override our equals method because, remember,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=160.29) [we can override any method we inherit.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=162.88) [Now one quick note on overriding methods.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=165.84) [Remember that when we do an override, we have to match the signature exactly](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=168.37) [of the method we're overriding. And believe it or not,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=173.32) [sometimes that's harder to get right than you might think.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=176.3) [Any small change in the way the method is declared might](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=179.14) [not actually override that method, but instead be declaring a brand‑new method.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=181.93) [So to help us with that, Java provides the @Override annotation.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=186.74) [So by putting this @Override before the method, that tells Java](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=191.24) [it's my intent to override a method that I've inherited. So a](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=195.34) [compiler will verify that for us.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=199.16) [So now here in the body of our equals method, one of the first things](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=201.54) [we'll need to do is take that object reference that we received and cast](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=204.53) [it to be a flight reference because we want to look at the flight aspects](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=208.54) [of the object we're comparing to.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=212.75) [And then once we have that reference, we can do the comparison.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=214.71) [Now something that's important to understand, there is a one‑universal](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=218.04) [definition of how you compare every type of class for equality.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=221.19) [It's up to you as a class implementer to decide the right](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=225.49) [way to determine what equality means.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=229.07) [We're going to make the determination that we consider two flights to be](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=231.72) [equal if they have the same identifying information.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=235.09) [So what we'll do here in our equals method is we'll simply return back](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=238.49) [the result of comparing the current instances' flight number to the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=241.61) [passed in instances' flight number and comparing the Flight class of the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=245.33) [current instance to the Flight class of the passed‑in instance. So](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=249.74) [basically, in our implementation of the Flight class, two flights are](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=253.52) [equal if they have the same identifying information. So now with that](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=256.82) [work done, let's see how our equality comparisons work.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=261.63) [Now, of course, we can still do the f1 == f2,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=264.64) [but that == operator is still going to compare the references themselves.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=268.24) [So that comparison is still going to return false, but now when we call](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=272.18) [f1.equals, passing in f2, it uses our override of the equals method.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=276.47) [So it's actually going to compare the contents of those two](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=281.96) [instances, and that will give us back a result of true.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=284.6) [So now our Flight class can handle equality comparisons of two flight instances.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=288.54) [But there's another situation we need to think about.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=293.54) [What if someone creates an instance of the Passenger class and](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=295.94) [then calls f1.equals, passing in a reference to that Passenger](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=299.05) [class? What's going to happen?](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=302.48) [Well, it turns out,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=304.64) [at that point our code would actually crash. So we need](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=305.44) [to deal with this scenario as well.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=309.09) [So back here in our Flight class,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=312.24) [when someone passes in a reference whose type is Passenger,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=314.44) [we're going to receive it here in our o parameter whose type](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=317.74) [is Object. Then inside the equals method,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=320.53) [the first thing we're going to try and do with it is cast that o to](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=323.03) [be a flight, but o doesn't point to a flight.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=326.41) [It points to a passenger,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=329.77) [so that's not a valid cast, and that's why our program would crash.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=331.74) [So we need to protect against this.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=336.08) [So to do that,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=338.44) [we're going to use that instanceof operator. And remember](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=339.37) [that instanceof makes sure that a reference points to an](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=342.52) [object of an appropriate type.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=345.39) [So as long as o points to an instance of the Flight class, everything is fine.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=347.64) [So what we're going to do is apply a not to that comparison.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=351.74) [So this sort of statement will be true if o points to](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=355.59) [anything that's not an instanceof Flight.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=358.09) [And if that's the case, we're simply going to return false,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=360.74) [which will handle the scenario of receiving a reference](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=364.24) [to something that's not a flight.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=366.94) [So if we, again, go through the equality checks,](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=369.74) [the == operator will still be false, f1.equals passing in f2 will](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=371.97) [still be true, but now when we create an instance of passenger](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=377.9) [and then call f1.equals, passing in the reference to the](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=381.53) [passenger, now we get back false.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=384.41) [So now we have an equals implementation that properly](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=387.67) [compares two flight instances, as well as handles any references whose type is not a flight.](https://app.pluralsight.com/course-player?clipId=991d73d6-9b16-4777-a5a3-60d6bda8ff02&startTime=390.36)

[Summary](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85)

[To wrap up, here's some of the key things you want to remember from this module.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=0.74) [Remember that as we declare our classes, one class can inherit from another.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=4.07) [And this provides some really powerful capabilities because](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=9.3) [remember that the derived class will have the characteristics of](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=12.29) [the base class that it inherits from,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=15.44) [but then the derived class can actually add its own specialization,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=17.47) [so the derived class builds on the work that was done in the base class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=21.39) [Now remember that inheritance affects the type of references we can use.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=27.54) [When one class inherits from another,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=31.44) [we can actually assign an instance of the derived class to a](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=33.23) [reference whose type is that of the base class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=37) [So base class references can point to instances of the derived class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=40.14) [But something to keep in mind, when using that base class reference,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=44.81) [the only features we will have access to are the](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=48.54) [features that the base class is aware of.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=51.39) [Remember the type of the reference affects which features can be accessed.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=54.3) [But then the derived class can override methods that it's inherited.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=58.84) [Now remember, to do that,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=63.48) [the derived class's method must have the same](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=64.88) [signature as the base class method, and Java gives some help to ensure that.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=67.3) [Remember we have that @Override annotation.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=72.22) [We apply that to a method to tell the compiler it is our intent](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=75.04) [to override a method that we've inherited.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=78.59) [And then by overriding that method,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=80.78) [remember that that method now becomes implementation](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=82.43) [used for all instances of your class,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=85.51) [even if your class instance is referenced by a reference](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=87.47) [whose type is that of the base class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=91.23) [So even when using a base class reference,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=93.74) [the overridden version of the method is what gets used.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=96.1) [Then we looked at the Object class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=100.34) [Remember the Object class is the root of the Java class hierarchy.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=102.24) [So that means that every class in Java inherits either](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=106.19) [directly or indirectly from the Object class,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=109.86) [which means that every class in Java has all the](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=112.74) [characteristics of the Object class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=116.66) [And the Object class provides a number of methods that we commonly override,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=118.77) [which allows us to specialize those methods for our particular class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=123.74) [One of those that we most commonly override relates to](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=128.14) [checking for equality because remember there's a couple](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=130.7) [different ways to check for equality.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=133.43) [One way is to use the equality operator,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=135.64) [or the == operator, but remember that does a reference‑based comparison.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=138.05) [In other words,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=142.88) [it checks to see if both references point to the exact same object instance.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=143.25) [In general, that's not what we want.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=147.54) [In general,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=149.94) [we want to compare the contents of each object to see if they're equal.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=150.81) [So to do that,](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=154.64) [we're going to override the equals method that we](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=155.67) [inherited from the Object class.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=157.9) [And within that equals method is where we'll compare the contents of our](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=159.45) [classes to determine what we consider to be equal.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=162.77) [Alright, that wraps up this module. In our next module, we'll continue our discussion of inheritance.](https://app.pluralsight.com/course-player?clipId=474431a2-24e3-418a-ae05-dd264ce0ce85&startTime=166.99)

[More About Inheritance](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e)

[Introduction](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e)

[Welcome to our next module, More About Inheritance.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=0.74) [This is part of the Pluralsight course, Working with Classes and](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=3.49) [Interfaces in Java, and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=6.01) [As you may have guessed by the title, in this module,](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=10.04) [we're going to continue our discussion of inheritance.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=12.54) [We're going to look at some of the finer points of dealing with inheritance.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=14.63) [So the first thing we'll take a look at is the Java super reference.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=18.24) [The super reference is sort of like the this reference,](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=21.94) [but has some specific capabilities as related to inheritance.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=24.53) [And then from there, we're going to look at a few issues related to](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=28.14) [controlling how a class is used in inheritance. For example,](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=31.1) [we have the option to prevent class inheritance.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=34.69) [We can declare a class that no other class is allowed to inherit](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=37.44) [from. That class can only be directly instantiated.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=40.52) [We can also declare classes that have methods that](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=44.24) [are not allowed to be overridden.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=47.06) [So although the class may be allowed to be inherited from,](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=49.08) [we have specific methods that cannot be overridden.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=51.52) [Now we can also require class inheritance. So we can declare](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=54.7) [a class that's not directly creatable.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=58.39) [Instead,](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=61.1) [the class only exists to serve as a base class. And then we can even](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=61.48) [have classes who have methods that must be overridden.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=65.28) [So when we declare a class, we don't provide an implementation of the method,](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=68.54) [but we require classes that inherit from our class to provide their own](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=72.03) [implementation. And then we'll finish up. We'll look at constructors and](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=75.59) [inheritance. And as we'll see, constructors behave a bit differently than methods do when inheriting one class from another.](https://app.pluralsight.com/course-player?clipId=88f3f8dc-ff30-46d0-af30-0ba28089750e&startTime=79.31)

[Special Reference: super](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a)

[So to get started, let's look at the Java super reference.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=0.74) [Now the super reference is very similar to the Java this reference,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=4.64) [and as you recall from our previous course,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=8.97) [Getting Started with Programming in Java,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=10.6) [we mentioned that this reference allowed us to refer to the current](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=12.62) [object and it turns out the super reference does that,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=16.31) [too.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=19.63) [Suppose this and super allows to refer to the current object,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=20.44) [but the super reference has a key difference from this.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=25.24) [When we use the super reference to refer to the current object,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=29.14) [it treats the object as if it's an instance of the base class and the](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=32.64) [reason that that's important because there are scenarios where we may](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=37.1) [have overridden something in the derived class and we want to access a](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=40.29) [member from the base class.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=44.62) [So using the super reference,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=46.94) [we can reach back and get method implementations from](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=48.7) [the class that we inherited from, even if we've overridden that method.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=52.12) [So to help us understand that,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=57.94) [let's start out with a quick look again at our Flight class.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=59.42) [As you recall,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=62.89) [our Flight class has a couple of fields in it that identify a particular flight,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=63.35) [that's flight number and Flight class. And in our previous module,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=67.87) [we overrode the equals method because as you recall that the full](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=71.81) [implementation of equals that we inherited from the Object class didn't](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=76.37) [look at the contents of a particular object, it simply compared to](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=80.25) [references to see if they were equal, and in our case,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=83.71) [that wasn't what we wanted.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=87.08) [We actually wanted to look at these flight number and Flight class](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=88.36) [fields to determine if two flights are equal.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=90.71) [So we overrode this equals method,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=93.94) [the first thing we did was check to see if o refers to](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=95.75) [an instance of the Flight class, if it doesn't,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=99.83) [we're just going to return false, but if it does,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=102.43) [we'll then cast that o reference from an object reference into a](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=105.23) [flight reference to store that into a local variable named flight,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=109.23) [and then once we add our flight reference,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=112.94) [we could go through and compare the flight number field in the](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=114.37) [current instance to the flight number field of the flight we received](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=116.88) [and do the same thing for the Flight class.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=120.28) [And in the overwhelming majority of scenarios,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=122.8) [this is the best way for us to implement the equals method.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=125.04) [But now, let's take a look at a special situation here.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=129.24) [What if I go ahead and create an instance of the Flight class](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=132.24) [specifying a flight number of 175 and I assign that reference to a](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=134.42) [variable of type flight named F1. And then we assign F1 to another](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=138.91) [flight reference named F2.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=144.29) [So now F1 and F2 point to the exact same object.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=146.54) [So then our program would go off and do some other work,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=150.88) [and then later we call F1 = F2.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=153.84) [So now we happen to know that F2 and F1 point to the same object instance,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=158.24) [but when our code runs and we get over to our Flight class and we're inside](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=163.74) [the equals method, what's going to happen?](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=167.23) [Well, we're going to take that reference F2, which](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=169.94) [we've received into our o parameter.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=171.89) [The first thing we're going to do is make sure that o refers](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=173.94) [to one instance of the Flight class, now,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=176.76) [of course, in this scenario, we know it does.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=178.52) [So then we do the work,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=181.04) [the cast o to a flight reference and then we do the work to](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=182.32) [compare those fields and we've done all this work even though the](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=186.29) [reference we received points to the exact same instance of the](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=190.66) [class that were currently in.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=193.69) [Now, in this case,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=195.64) [we only did a few steps, so doing these steps is not that big a deal.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=196.39) [But in some scenarios, you may do a great deal of](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=201.14) [work inside this implementation.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=203.46) [And to do all that work unnecessarily would be unfortunate because remember](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=206.14) [that our base class object, if we recall it's equals method,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=210.57) [it would do that very inexpensive check of simply looking at](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=214.54) [the references themselves and looking to see if they point](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=217.29) [to the same object instance.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=220.1) [So by making a call to the base classes equals method, again,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=222.34) [our object classes equals method, we could have very easily](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=225.66) [determined that these two flights were equal because both references](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=228.95) [refer to the same object instance, so we could have simply returned](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=232.37) [true without doing all that other work.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=235.14) [But the problem here,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=238.44) [if you look at the way we've called equals, we're not calling](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=239.67) [equals as it's implemented in our base class,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=242.48) [we'd be calling equals as it's implemented in our class.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=245.54) [So what that would mean, when we call equals here, we would](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=249.78) [simply call right back into ourselves.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=252.74) [So if we call back into ourselves, the first thing we would](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=255.64) [do is again make that call to ourselves,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=258.04) [calling back into ourselves, again, we'd make that](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=260.84) [call into ourselves, and again, coming back into ourselves,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=263.53) [and we would actually keep doing this until our program finally crashed](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=267.14) [because our equals method is repeatedly and unendingly calling back into](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=271.24) [itself and that's not what we wanted to do. When we called equals here, what](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=275.31) [we wanted to do was call equals as it's implemented in our base class and](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=280.26) [that's where the super reference comes in.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=285.31) [If instead of simply saying equals, if we say super.equals,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=288.14) [that's just that, yeah, I want the equals method,](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=292.74) [but not as I've implemented it, but as my base class has implemented it, so](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=294.95) [that allowed us to override our equals method providing the implementation](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=299.84) [that's right for our particular class while leveraging the work that was](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=304.21) [provided for us by our base class.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=308.02) [Alright, so now in our next clip, let's see how we can prevent inheritance and method overriding.](https://app.pluralsight.com/course-player?clipId=64b90cb5-f063-4195-919d-5e6eade41e1a&startTime=311.04)

[Preventing Inheritance and Method Overriding](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda)

[When we declare a class, by default,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=0.74) [we have no control over how that class is used in inheritance,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=3.64) [because by default,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=7.44) [any class we declare can be extended, so other classes](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=8.61) [are free to derive from our class.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=12.17) [In addition, if someone derives from our class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=14.74) [the deriving class is free to override any of our methods,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=17.28) [and although these are generally good things,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=21.05) [it's not always the behavior we want.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=23.94) [So for that reason, we have the final keyword.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=25.94) [The final keyword allows us to override this default behavior.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=28.84) [So by using the final keyword,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=33.04) [we can actually have a class that we don't allow other classes to extend](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=34.66) [or we may want to allow someone to extend the class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=38.55) [but there may be specific methods that are may be critical to our](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=41.58) [class that we don't want to allow anyone to override,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=44.14) [and we can prevent method overriding by again using the final keyword.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=47.54) [So let's first look at preventing inheritance.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=52.84) [Now remember,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=55.28) [we have this Passenger class and the Passenger class was our very simple class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=55.68) [but it was very important to our Flight class because,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=59.94) [remember,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=62.51) [the Passenger class represents a passenger that's added to our Flight class.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=62.71) [Now, by default,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=67.1) [anyone can come along and extend this Passenger class so they could](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=68.18) [create their own custom versions of passengers.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=71.94) [But one of things you want to keep in mind if you're going](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=74.64) [to allow someone to extend your class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=76.84) [you want to make sure that any of your code that depends on that](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=79.44) [class will behave correctly if someone extends it.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=81.57) [So because we consider the Passenger class so critical](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=85.04) [to the behavior of our Flight class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=87.94) [we might make the decision to not allow other people to](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=89.7) [extend our Passenger class and we could enforce that by](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=93.01) [simply marking the class as final.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=96.29) [So once the class is marked as final, no other classes can inherit from it.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=98.84) [Now one thing to note, here we have final public.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=103.44) [Those two keywords, final and public, can appear in either order.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=107.2) [So we could say public final, as we have here,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=111.4) [or final public, as we had previously.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=114.17) [Now, in some scenarios, you may be willing to allow others to extend your class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=118.19) [but there might be certain methods that are critical](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=122.74) [to the behavior of your class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=125.28) [so you don't want anyone who inherits from your class to override those methods.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=127.04) [So let's take a look here at our CargoFlight class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=131.34) [and with our CargoFlight class had one public method,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=133.05) [which was add one package.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=136.44) [Then we had a couple of private methods, hasCargoSpace and handleNoSpace.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=138.44) [Now let's say for our CargoFlight class,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=143.01) [we consider our add1Package method to be critical](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=145.02) [to the behavior of CargoFlight.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=148.44) [So we don't want to allow anyone to override this method.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=150.44) [Well, then all we need to do it simply mark that method as final,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=154.44) [and then once we mark it as final,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=158.84) [anyone could inherit from our CargoFlight class, but no one else can](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=160.65) [provide their own implementation of add1Package.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=163.9) [Alright, so now in our next clip,](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=167.74) [let's take a look at the other side of this issue, requiring inheritance and requiring method overriding.](https://app.pluralsight.com/course-player?clipId=cbbbefb3-d822-4b3a-a05d-cfa5f4e14dda&startTime=169.5)

[Requiring Inheritance and Method Overriding](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881)

[Now if we look at the default way we use a class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=0.64) [well,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=3.44) [once we declare the class, we're generally going to expect to](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=3.64) [directly create instances of that class.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=6.77) [And then if someone does decide to extend our class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=9.54) [well, it's up to them which of the methods they've inherited](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=12.4) [they wish to override. And in general,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=15.01) [these defaults are what we're looking for. We declare a class so we](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=17.77) [can create instances of it, and so when it inherits from our class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=21.22) [it's up to them what they want to override.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=24.26) [But there are some scenarios where we don't want that default behavior,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=26.94) [and that's where the abstract keyword comes in.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=30.64) [With the abstract keyword,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=33.74) [we can indicate that a particular class can't be directly created.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=35.2) [It can only be extended. And, again, with the abstract keyword,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=39.44) [we could indicate that there are certain methods that](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=43.23) [when you inherit from this class, you have to override those methods.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=45.58) [Now these scenarios may seem a bit strange,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=50.94) [but there are some cases where it's really,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=53.21) [really useful.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=55.07) [Let's look at an example.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=56.31) [So let's declare a new class here named Pilot,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=58.34) [and the pilot is someone who flies a plane. So we want to be able to](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=61.34) [associate a flight with this pilot, so we have this currentFlight field.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=64.95) [And our Pilot class will expose a method named fly that accepts a](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=69.74) [reference to a flight, and that's how we indicate that we want a pilot](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=73.74) [to fly a particular flight.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=77.71) [But in order for the pilot to fly the flight,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=79.64) [they've got to determine if they're able to accept that flight.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=81.68) [So we'll call a method named canAccept. As long as canAccept returns true,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=84.84) [then we'll go ahead and associate that flight with the currentFlight field.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=89.82) [If canAccept returns false, then we'll indicate that we](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=93.34) [can't accept that flight. Now implementation of](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=96.02) [handleCantAccept will be really simple.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=98.78) [We'll just print out a message.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=101.07) [But now when it comes to our canAccept method,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=102.84) [what flights a pilot is allowed to accept is going to](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=105.07) [depend on what kind of pilot it is.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=108.43) [So here in our Pilot class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=111.47) [we're not going to directly implement the canAccept method.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=113.32) [We're going to leave that up to classes that inherit from Pilot.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=117.24) [So what we'll do is as we're declaring our canAccept method,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=121.14) [we'll start out by marking it as abstract.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=124.34) [And then once we indicate that it's abstract,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=127.64) [we'll provide its returned value, as well as the rest of its](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=129.85) [signature. Now because canAccept is marked as abstract, that](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=132.66) [means that here in the Pilot class, we're not going to provide an implementation.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=136.74) [So rather than putting opening and closing brackets on the method,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=141.24) [we're just going to put a semicolon.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=144.48) [Now this might seem a bit odd.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=146.54) [We're declaring a method here, but we're not providing a body for the method.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=148.44) [So what's the reason for declaring the method at all?](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=152.54) [Well, because we've declared this method here in the Pilot class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=155.94) [the Pilot class is allowed to make calls to that method.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=158.77) [So our fly method here is able to call the canAccept method,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=162.34) [even though the Pilot class doesn't know the details of how the](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=166.54) [canAccept method is going to be implemented.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=169.79) [But we can still have logic here that leverages that method.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=172.74) [Now one thing to note,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=176.34) [because canAccept is marked as abstract, that means the method must be](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=177.6) [overridden. So if we go up here to our Pilot class declaration, because](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=182.54) [it contains an abstract method, the Pilot class itself must also be](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=186.78) [marked as abstract, meaning that we cannot directly create an instance](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=190.8) [of the Pilot class.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=194.35) [We can have references whose type are Pilot, and other](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=196.04) [classes can extend the Pilot class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=198.8) [but you can never directly create an instance of the Pilot class.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=200.95) [So any time a class has an abstract method,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=204.34) [the class itself must also be marked as abstract.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=207.14) [Now, technically,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=210.28) [you can have a class marked as abstract that has no abstract methods,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=211.23) [but that would be extremely uncommon. In general, a class is marked as](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=215.63) [abstract because it contains one or more abstract methods.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=219.5) [So now that we have our Pilot class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=224.54) [we can declare other classes that extend the Pilot class.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=226.22) [So let's start out here with a CargoOnlyPilot.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=229.84) [CargoOnlyPilot extends the Pilot class. Because the Pilot](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=232.84) [class contains an abstract method, CargoOnlyPilot is](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=236.45) [required to override that method.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=239.26) [We'll be sure to mark that method with the @Override annotation.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=241.74) [Now the implementation of canAccept is up to this class. And what we're going](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=245.44) [to use the CargoOnlyPilot class for is pilots that are only allowed to carry](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=249.57) [cargo. They are not allowed to carry any passengers.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=254.03) [So in this class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=257.24) [the implementation of canAccept is going to check the number](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=258.42) [of passengers on that flight. And as long as it equals to 0,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=261.48) [we'll go ahead and return true.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=265) [So now let's go and declare another class, the FullLicensePilot class.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=267.14) [Now this also extends the Pilot class, and](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=271.04) [FullLicensePilot is kind of our super pilot.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=273.68) [He's allowed to fly any flight he wants to. But now because he extends the](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=276.34) [Pilot class, he's required to override the canAccept method.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=280.86) [Again, we'll be sure to mark it with our @Override annotation.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=284.27) [And for this class,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=287.84) [the implementation of canAccept is always going to return true. Because we](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=289.17) [would use the FullLicensePilot class, the representing pilot who has a](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=293.5) [license that allows him to fly any flight he'd like to.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=296.91) [So what we have here now are two classes, CargoOnlyPilot and FullLicensePilot,](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=300.44) [that fully leverage the capabilities of our Pilot class, but each one has](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=304.82) [provided its own specialization of the canAccept method. So now in our next clip, let's take a look at inheritance and constructors.](https://app.pluralsight.com/course-player?clipId=2de01f89-2dde-4bb2-b21b-24b1c4e8d881&startTime=309.92)

[Inheritance and Constructors](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6)

[When one class inherits from another,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=0.64) [one of the really neat things that happens is a derived class](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=2.04) [automatically exposes all the public methods of its base class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=5.53) [But in case of constructors, things work differently.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=9.54) [Constructors are not inherited.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=12.51) [In other words,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=14.8) [the only constructors that are available in our class are ones](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=15.39) [that are directly tied to that class itself.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=18.48) [And this has important implications about how we handle](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=21.64) [constructors when we derive one class from another.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=23.9) [So when we create an instance of a class that derives from another class,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=28.14) [remember that that class is itself an instance of the base class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=31.93) [In other words, the derived class has the characteristics of its base class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=35.92) [So as part of constructing the derived class,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=40.91) [a base class constructor is always called.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=43.31) [So any construction work a base class expects has an opportunity to happen.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=46.25) [Now by default, when we construct an instance of the derived class,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=50.67) [it will call the no‑argument version of its base class constructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=54.78) [So it does kind of the default construction on the base class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=58.37) [But we have the option to explicitly call one of the base class constructors,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=62.24) [and we do that by using the super keyword.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=67.04) [Remember, we could call our own class's constructor by using the this keyword.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=69.66) [We can call our base class constructor by using the super keyword.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=73.31) [And we're able to pass any necessary parameters to that base class instructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=76.45) [But now one thing to keep in mind,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=81.14) [if we're going to call the base class constructor explicitly,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=82.3) [it has to be the first line of our constructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=85.01) [Now let's again take a quick look at our Flight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=89.53) [Remember that one of the fields in our Flight class was flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=91.94) [When our Flight class exposed a number of constructors,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=94.4) [it had its default constructor, which accepts no parameters,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=96.86) [and it also had a version of its constructor that accepted the](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=100.54) [flightNumber that it used to set its flightNumber field.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=103.26) [Remember, the Flight class serves as the base class for our CargoFlight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=106.94) [So our CargoFlight classic extends the Flight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=111.13) [But, remember, the CargoFlight class has no explicit constructors of its own.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=113.82) [So how does that affect things when we try to create instances of these classes?](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=117.98) [Now, of course, in the case of our Flight class,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=121.5) [it explicitly exposes a constructor that accepts the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=123.85) [So here we can create an instance of Flight, setting its flightNumber to 175.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=127.24) [Now, remember, in the case of our CargoFlight,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=130.74) [because we have no explicit constructors,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=132.48) [Java will automatically provide the no parameter constructor for us.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=134.94) [So we can create an instance of CargoFlight that specifies no](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=138.95) [parameters because we're using that default constructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=141.98) [But now what happens if we try to create an instance of](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=145.24) [CargoFlight specifying its flightNumber?](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=148.01) [Its base class Flight has a constructor that accepts a flightNumber,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=149.98) [but that does not mean that CargoFlight can use that same kind of constructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=153.79) [If CargoFlight does not explicitly expose a](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=158.24) [constructor that accepts a flightNumber,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=160.38) [we'll get an error when we try to use a constructor that accepts a flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=162.33) [So let's see how we can improve our CargoFlight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=167.1) [Well, if we want to create a CargoFlight instance that accepts a flightNumber,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=169.64) [we'll need to explicitly declare a constructor that accepts that flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=172.74) [Now as part of constructing this instance,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=177.04) [we don't want to duplicate the work that's already in the Flight class,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=178.51) [so we'll simply use a super keyword to execute the Flight](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=181.74) [class's constructor that accepts the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=184.7) [So this way we're creating a very simple constructor that says that](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=187.44) [our CargoFlight class accepts a flightNumber,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=190.33) [but we're not duplicating the work that's in the Flight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=193.23) [We're just explicitly calling the constructor provided by the Flight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=195.94) [Now our CargoFlight class has some fields of its own.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=200.24) [One of those fields is the maxCargoSpace, and it has the default value of 1000.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=202.9) [But in some cases,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=208.14) [we might want to create a CargoFlight that has a](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=208.98) [different amount of cargo space.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=210.78) [So we might expose a constructor here that accepts both a](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=212.94) [flightNumber and the amount of cargo space.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=215.26) [So in this constructor, we'll go ahead and set the maxCargoSpace field, but we](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=218.64) [still need to do the work of dealing with the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=222.58) [Now remember that any calls we make to another constructor have](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=225.54) [to be the first line in this constructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=228.32) [So we could again use the super keyword to call the](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=230.64) [Flight class's constructor that accepts a flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=233.15) [And that would certainly be fine.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=236) [But in our case,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=238.34) [our CargoFlight class already has a constructor that accepts the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=239.3) [So it would be better that rather calling the Flight class's constructor that](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=243.74) [accepts a flightNumber, we should use the this keyword to call the CargoFlight](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=247.28) [class's constructor that accepts the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=251.16) [That way the work of constructing an instance of CargoFlight that](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=253.87) [accepts a flightNumber is centralized in one constructor.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=256.62) [Now our CargoFlight class should probably have some](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=260.14) [additional constructors as well.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=262.36) [So let's go ahead and expose a constructor that accepts no parameters.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=264.34) [Now this constructor has an empty body.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=267.49) [But that doesn't mean this constructor does nothing. Because the](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=269.6) [CargoFlight class inherits from the Flight class, if we do not](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=272.6) [explicitly call one of the Flight class's constructors, then the](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=275.77) [CargoFlight class's constructor will automatically call the Flight](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=278.51) [class's constructor that accepts no parameters.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=281.83) [Now let's go ahead and add one more constructor. Let's add a](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=284.94) [constructor here that accepts just the maxCargoSpace.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=287.27) [And inside this constructor, we'll go ahead and set that](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=290.84) [maxCargoSpace field. So, again, we make no explicit call to](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=292.94) [the base class's constructors.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=297.44) [So, again,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=299.16) [we're going to automatically call the Flight class's](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=299.56) [constructor that accepts no parameters.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=301.67) [So now with all this in mind, our CargoFlight class has four constructors,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=305.14) [one that accepts the flightNumber, one that accepts](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=309.14) [flightNumber and maxCargoSpace,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=311.4) [one that accepts no parameters, and one that accepts just maxCargoSpace.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=313.34) [So let's go ahead and create some instances of the CargoFlight class.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=318.14) [So now we'll go ahead and create a CargoFlight instance](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=321.34) [that sets the flightNumber to 294.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=323.46) [When we do that,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=326.24) [we're calling into our CargoFlight class's](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=327.22) [constructor that accepts the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=329) [And remember that this constructor makes an explicit call to the Flight](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=331.13) [class's constructor that accepts the flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=333.96) [Now let's go ahead and create another instance of CargoFlight.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=336.9) [This one specifies both the flightNumber and a maxCargoSpace,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=338.99) [so we'll call the constructor that accepts those two values.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=343.54) [Remember that this constructor sets our maxCargoSpace field. But before](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=346.54) [it does that, it calls the CargoFlight class's constructor that accepts](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=350.29) [flightNumber. And then that in turn calls a Flight class's constructor](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=353.86) [that accepts flightNumber.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=357.88) [Now let's go ahead and create an instance of CargoFlight that](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=359.6) [specifies no parameters to the constructor. So that calls the](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=361.7) [CargoFlight class's default constructor,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=364.99) [which will automatically chain to the Flight class's default](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=367.11) [constructor. And let's create one more instance of CargoFlight.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=370.35) [This one specifies only the cargo space. So that will call into](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=374.18) [our constructor that accepts maxCargoSpace.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=378.14) [Now this constructor sets our maxCargoSpace field.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=380.94) [But before that happens,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=383.66) [the implicit call to the Flight class's default constructor will still occur.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=384.96) [So as you can see,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=389.48) [any time we create an instance of a class that derives from another class,](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=390.31) [that base class constructor always gets called. We can do it explicitly, or it will happen implicitly.](https://app.pluralsight.com/course-player?clipId=67e98146-3b73-4c29-8249-ab3bd26010e6&startTime=393.87)

[Implementing Abstract Classes and Constructor Inheritance](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87)

[Here we are in IntelliJ, and what we want to do now is take a look at how we](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=0.64) [can use some of the techniques we've learned in this module to improve the](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=4.54) [use of inheritance in our CalcEngine project.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=7.94) [As you recall in a previous module,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=10.74) [we added some classes that were specific to the individual math operations.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=12.53) [For example, we have our class here, Divider.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=17.24) [We also have a class, Adder.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=19.35) [In addition to those, we created classes to handle](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=21.04) [subtraction and multiplication.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=23.26) [As you recall,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=25.74) [as we added the code to our Main class to leverage these math classes,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=26.62) [we added this method here, doCalculation.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=30.71) [And in this doCalculation method, what we did was receive a reference to the](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=33.54) [individual class, as well as the left and right values.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=37.64) [Once we had those, we could go through and set the left and right](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=40.94) [values, perform the calculation, and display the result.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=43.49) [And the reason we were able to do that is because each of the](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=47.64) [classes that derive from CalculateBase override the](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=50.43) [CalculateBase class' calculate method.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=53.78) [But now an interesting question here, what would happen if someone created an](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=56.94) [instance of CalculateBase rather than one of the derived classes and called](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=60.35) [doCalculation using that? Well, let's take a look at our CalculateBase class,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=64.41) [and let's see what we have there.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=68.11) [So now we're here looking at our CalculateBase class,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=70.44) [and you can see here where we have the calculate method.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=73.04) [Now notice the calculate method has an empty body,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=75.74) [and the reason it has an empty body is because we expect each class it](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=79.04) [inherits from CalculateBase to actually override that method.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=82.66) [We never really expect anyone to directly create an instance of CalculateBase.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=86.34) [But as this class is currently written, they could.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=91.84) [So what we want to do now is take advantage of the abstract modifier.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=94.79) [So here where we have our calculate method, we expect everyone](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=99.44) [who inherits from CalculateBase to override it,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=102.34) [so let's go ahead and mark this calculate method as abstract. Now by](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=104.84) [marking this method as abstract, that means we don't expect it to](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=108.73) [actually have a body here in CalculateBase.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=111.55) [So let's replace those opening and closing brackets with a semicolon.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=114.54) [Now one last thing we need to do, since](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=119.14) [CalculateBase contains an abstract method,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=120.95) [the class itself must also be marked as abstract.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=123.4) [So now with that, we can be assured that no one can ever create an](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=127.74) [instance of CalculateBase. So that takes care of one key issue, but](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=130.65) [we have another important consideration.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=134.83) [The way our classes are currently written, in order to use any of](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=137.24) [the classes that inherit from CalculateBase,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=140.07) [we have to explicitly set the left and right values.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=142.99) [It would be nice to be able to construct those instances](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=146.34) [providing those left and right values.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=148.84) [So to do that, let's add a constructor here to our CalculateBase class.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=151.34) [So we'll have a public instructor, CalculateBase, that accepts a](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=155.24) [left and right value. Then inside this constructor, we'll go and](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=158.36) [set the left and rightVal fields.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=162.83) [So now we have a constructor that accepts the left and right values.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=166.04) [Let's go and add a default constructor as well.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=169.36) [So now we have the constructors we need here in our](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=172.54) [CalculateBase class, but remember that other classes won't](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=174.72) [automatically inherit those constructors.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=178.07) [We'll need to add those constructors to those classes individually.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=179.88) [So let's head up here to our Adder class. And here in our Adder](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=183.34) [class, we'll first add our default constructor.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=186.75) [Now remember,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=190.99) [because this constructor doesn't explicitly call one of the](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=191.67) [base class constructors, it will implicitly call the default](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=194.14) [constructor in CalculateBase.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=197.72) [But now we also want a constructor that accepts the left and](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=200.04) [right values, so let's go and add that.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=202.46) [And then here in this constructor,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=205.74) [we want to call the CalculateBase class' constructor that accepts those values.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=207.16) [So to do that, we use the super keyword.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=212.29) [Then as parameters, we'll pass leftVal and rightVal.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=215.88) [So that takes care of our Adder class.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=220.04) [We can create an instance of Adder passing no parameters,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=222.3) [or we can create an instance of Adder passing in values](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=224.93) [for our leftVal and rightVal.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=227.7) [So now I'll go and do the same thing for each of the other three classes.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=230.07) [So now all of our classes, Adder, Subtracter, Divider, and](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=234.74) [Multiplier each have two constructors.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=238.57) [They each have a default constructor, as well as a constructor that](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=241.44) [allows us to specify the left and right values.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=244.37) [But now before we finish up here,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=247.44) [we have one last little thing we need to do. Notice I'm looking here at the](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=249.11) [Mulitiplier class, and as I mentioned in a previous module,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=252.54) [I made a typo in this class name when I created it.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=256.17) [Notice that I have an extra i here.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=259.18) [And, of course, a typo like that could have far‑reaching impacts in my code.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=261.44) [For example,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=265.24) [each of the constructors needs to have that same misspelled](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=266.05) [name, and anywhere I use this class would also have that same](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=268.71) [improper spelling. And often fixing that sort of thing can be](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=271.87) [cumbersome, but fortunately,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=275.26) [IntelliJ makes fixing this sort of thing really easy.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=276.92) [So if we go up here and right‑click on the class name,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=279.74) [I'm going to head down here to where it says Refactor, and](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=282.44) [then I'm going to head up here to where it says Rename,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=285.53) [and then I'll choose Rename.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=287.82) [You'll notice what this will let me do now is go through here and](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=290.64) [actually make any changes to the name I need to.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=293.49) [So I'll delete that extra i. And you'll notice that when I make that change,](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=296.04) [IntelliJ is automatically making that same change anywhere the name appears.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=299.75) [So I'll go ahead and hit Enter, and now that easily, I fixed the typo I made](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=304.74) [in the name of my Multiplier class.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=309.98) [So now in our next clip, we'll write some code that takes advantage of the improvements that we've made.](https://app.pluralsight.com/course-player?clipId=580c2cfb-66ce-4439-a966-48b2005d4f87&startTime=312.34)

[Using Constructor Inheritance in CalcEngine](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8)

[Here we are back in IntelliJ, and what we want to do now is use](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=0.64) [our new and improved MathOperation classes.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=4.13) [So now I'm looking at our Main class, and I'm in our main](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=7.44) [method. So let's do all that work in a new method we'll](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=9.84) [create named performMoreCalculations.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=12.29) [We'll let IntelliJ create the method for us, so hit Alt+Enter,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=16.84) [Enter, and then Enter once more.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=20.44) [So we scroll up a bit so we have some room to work.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=23.44) [So here I perform more calculations. Let's create an array of type](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=26.94) [CalculateBase, and we'll name it calculations.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=30.85) [And then here in our calculations array, we'll create instances of our](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=35.08) [individual classes that inherit from CalculateBase.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=38.69) [So let's start out by creating a new instance of Divider,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=42.28) [then here as we create this instance, let's pass in values](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=45.44) [for the leftVal and rightVal.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=49.33) [So that gives us an instance of Divider that will divide 100 by 50.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=52.34) [So now let's go and create instances of the other three classes.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=56.84) [So now we have our array initialized to have one instance of Divider,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=61.14) [one of Adder, one of Subtracter, and one of Multiplier.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=64.38) [So now we can do some work that will actually walk through](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=68.14) [that array and perform the calculations.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=70.18) [Now before we do that,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=72.74) [we just print out a message that indicates we're going to do the array](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=73.94) [calculations. Now once we print the message, we can go and process our](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=77) [array. So we'll start out with a for loop, we'll declare a variable named](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=81.6) [calculation of type CalculateBase, and here in our loop, we'll walk](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=86.48) [through our calculations array. And then here in the body of our loop,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=91.7) [our work is really, really simple.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=95.82) [We can simply call calculation.calculate,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=98.18) [and then we can print out the result of our calculation.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=102.14) [And then once we do that, let's go and run our application.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=106.2) [And if we look here at our output window, we can see our results.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=110.1) [Dividing 100 by 50 gives us,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=113.24) [2, adding 25 and 92 gives us 117, subtracting 17 from 225 gives us](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=114.71) [208, and multiplying 11 times 3 gives us 33.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=121.03) [And as you can see,](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=125.07) [the work we've done has really improved our classes. By making](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=125.74) [CalculateBase abstract, it assures that no one ever tries to](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=129.18) [directly create an instance of CalculateBase. And by providing](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=132.14) [the appropriate constructors, it was very easy to create instances of our individual classes.](https://app.pluralsight.com/course-player?clipId=a91bc742-0945-4e1a-be92-d99ecfd4b2e8&startTime=135.69)

[Summary](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4)

[Here are some of the key things you want to remember from this](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=0.64) [module. Remember that we have the super reference.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=2.87) [The super reference is kind of like the this reference in the](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=5.84) [sense that it refers to the current object.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=8.4) [The key is the super reference treats that object as if](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=11.47) [it's an instance of its base class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=14.79) [So it allows us to access method implementations in the base class](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=17.24) [that we might have overridden in the derived class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=20.62) [Then we looked at some of the different ways we could override](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=24.54) [default behavior of classes. One of the things we saw is we](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=27.36) [can actually prevent inheritance.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=30.98) [Remember that by default, a class can be extended. In some cases,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=32.97) [we don't want our classes to be extended.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=36.87) [So in those cases we can mark that class as final, and once it's marked as](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=38.88) [final, Java will not allow any class to extend that final class. In other](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=42.84) [situations, we're okay with having our class extended, but there are certain](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=48.31) [methods that we don't want overridden. In general, these are methods that we](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=52.24) [consider critical to behavior of our class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=56.38) [So in these situations, we can mark individual methods as final, and](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=59.34) [when we do that. people are free to extend our class, but any methods](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=63.82) [that are marked as final cannot be overridden.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=67.64) [Then we saw we actually had the option to require inheritance. So we could](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=70.54) [declare a class that we never want directly created.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=75.24) [Instead, we want it to serve as a base for other classes.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=78.47) [We consider that an abstract class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=82.02) [So we use the abstract keyword to mark that class as abstract, and in general,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=84.43) [when we have an abstract class, it's because that class has one or more](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=89.01) [methods that we're not going to provide implementations of.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=92.3) [We're going to leave that implementation work to](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=95.84) [anyone who derives from our class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=97.61) [So in that case, we would have one or more abstract methods.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=99.48) [Remember, that an abstract method has a signature and a return type,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=103.14) [but it doesn't have a body.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=106.67) [So it's the responsibility of classes that inherit from](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=108.64) [our class to provide that method body.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=111.44) [And then we finished up with a look at constructors.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=115.19) [Remember that constructors are not inherited.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=117.64) [Each class has its own constructors.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=120.27) [So any special constructors we want our class to have, we have](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=123.34) [to explicitly make them part of the class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=126.16) [We won't inherit those constructors from our base class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=127.92) [And remember that when we construct an instance of our derived class,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=131.44) [because that derived class has the characteristics of its base class, a](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=134.53) [base class constructor always has to be called.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=138.17) [Now, we don't have to have an explicit call to a base class constructor,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=141.89) [because by default,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=144.72) [the no‑argument instructor of the base class is automatically called.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=146.14) [But we also have the option to explicitly call one of](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=149.74) [the constructors in the base class.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=152.75) [Remember,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=155.04) [we do that by using the super keyword and providing any](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=155.37) [parameters we want to pass to that constructor.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=158.2) [Alright,](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=161.64) [that wraps up this module. In our next module, we'll take a look at a special kind of type in Java known as enums.](https://app.pluralsight.com/course-player?clipId=a739c9bf-f2fd-4d31-bd8a-2b5ccb7c39e4&startTime=162.3)

[Working with Enums](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc)

[Introduction](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc)

[Welcome to our next module, Working with Enums.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=0.74) [This is part of the Pluralsight course,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=3.47) [Working with Classes and Interfaces in Java,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=5.02) [and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=7.11) [Oftentimes in our applications,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=9.94) [we may want to have a pre‑defined list of values that we](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=11.78) [want to be valid particular fields or variables, and](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=14.43) [that's where enum types come in.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=17.76) [So in this module, we'll start out, we'll look at how enum](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=19.94) [types solve this particular kind of problem.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=23.21) [We'll also see how we can declare enum types,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=25.64) [which allows us to define a type that only supports this finite list of values.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=28.15) [As we work with these types, we're, of course,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=33.24) [going to need to do conditional logic with it.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=35.06) [So we'll see how conditional logic works when dealing with an enum type.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=37.34) [And these kind of comparisons don't have to be limited to simple,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=41.19) [equal to, or not equal to comparisons.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=44.34) [So next, we'll see how we can do relative comparisons with enum types.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=47.34) [So one of these values can be compared to another to see if it's](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=51.34) [greater than or less than that other value.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=54.5) [Now enum types go beyond just simple values. Enum types also have methods.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=57.54) [And whenever we declare an enum type,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=62.89) [there are certain methods we automatically have access to.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=64.51) [And in fact, enum knew types are actually classes,](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=67.54) [which means we can actually associate characteristics with](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=70.94) [each of the values within our enum type.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=73.74) [So we'll finish up this module.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=76.48) [We'll look at how we can provide our own characteristics for each of our enum types.](https://app.pluralsight.com/course-player?clipId=f41c27da-c1b6-4cdf-9eab-a8eb08aaa1bc&startTime=78.04)

[Conditional Logic](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27)

[As we mentioned,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=0.74) [enumeration types are useful for when we define a type](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=1.4) [that has a finite list of valid values.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=4.91) [In other words,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=7.44) [enumeration types allow us to create a brand‑new type and then](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=8.33) [specify what the valid values for that type are.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=11.75) [Now the way we declare an enumeration type is by using the enum keyword.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=15.8) [So, for example, here we have a type named FlightCrewJob,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=20.04) [and we specified that it's an enum.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=23.7) [So as we declare this type,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=26.34) [one of the things we want to do is provide the values that are part of the type.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=27.64) [So we're going to do that with a comma‑separated list.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=31.72) [Now, by convention, each of the values will be in all uppercase.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=34.94) [So if we're going to provide a list of](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=38.81) [FlightCrewJobs that our application allows,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=40.41) [we could say we have a FLIGHT\_ATTENDANT,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=42.69) [we can have a COPILOT, and we can have a PILOT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=44.33) [Now once we declare this,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=47.04) [FlightCrewJob is a first‑class type within our application,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=49.69) [and what that means is if we declare a variable to have the type FlightCrewJob,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=53.94) [the compiler will assure that only one of these values,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=58.64) [FLIGHT\_ATTENDANT, COPILOT, or PILOT is ever assigned to that variable.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=62.11) [Now once we have our enumeration type defined in our application,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=68.34) [in general, we're going to want to perform some kind of conditional logic.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=71.42) [And enumeration types support equality tests.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=75.28) [So we can use the == or != operators.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=78.77) [So let's say we have a variable here, job1, whose type is FlightCrewJob.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=83.14) [We've given it a value of PILOT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=87.44) [Now notice that when we assign this value, we can't simply say PILOT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=89.94) [We have to qualify it by the type.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=93.52) [So we say job1 = FlightCrewJob.PILOT, not simply job1 = PILOT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=96.14) [So let's go ahead and declare another variable here, job2.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=102.1) [We'll specify the value for job2 as FlightCrewJob.FLIGHT\_ATTENDANT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=103.84) [So now, once we've assigned values to the variables,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=108.19) [we can do our conditional logic.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=111.62) [So I can check job1 here against that value FlightCrewJob.PILOT. And,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=113.84) [again, we're doing this comparison with the == operator.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=118.63) [So in this case, we'd go ahead and print out "job1 is PILOT." Now](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=122.37) [we can do these equality or inequality comparisons using not just](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=126.91) [the constants, but also we can compare two variables. So I could](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=130.81) [say if(job1 != job2) which, in this case,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=134.36) [they are unequal, we could then take an action, like print out "job1 and job2](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=137.87) [are different." Now our enumeration types will generally have multiple](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=142.25) [values. So, oftentimes, we'll want to go beyond a simple == or +!. We'll want](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=146.03) [to have different branches in our code,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=151) [depending on which particular value a variable has. And](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=152.72) [we'll do that using switch statements.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=156.16) [So within an enumeration type,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=158.44) [we can perform a switch and have a separate case for](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=159.92) [each of the values for that type.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=162.71) [So let's look at some code. We'll add a method displayJobResponsibilities.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=165.78) [Notice it has one parameter named job of type FlightCrewJob.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=169.94) [So inside this method, we can have a switch statement that switches on job.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=174.44) [Then inside the switch statement,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=179.04) [each case can be a value for FlightCrewJob. So I can](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=180.41) [have a case here for FLIGHT\_ATTENDANT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=184.12) [Now one thing to note, inside the switch statement, we](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=186.71) [don't have to type qualify the value.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=189.52) [We don't say Case FlightCrewJob.FLIGHT\_ATTENDANT.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=192.04) [We simply say case FLIGHT\_ATTENDANT. Then inside the case,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=195.25) [we just do whatever work we need to do.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=198.66) [So in this case,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=200.75) [we'll print out the message "Assures passenger safety." Now, remember,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=201.43) [at the end of each case, we want to have a break statement,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=205.12) [and then we could provide our next case.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=207.98) [So we'll have case COPILOT, and for COPILOT, we'll print out the message](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=209.42) ["Assists in flying the plane." And for the PILOT, we'll say,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=213.11) ["Flies the plane." So as you can see here, by using an enumeration type,](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=216.26) [we've got really readable code.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=220.98) [It's very clear what scenario each case is handling. All right, so now in our next clip, let's take a look at relative comparisons.](https://app.pluralsight.com/course-player?clipId=26dc4468-54bb-4d3a-a31e-1d3f92382a27&startTime=222.84)

[Relative Comparisons and Common Methods](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082)

[As we deal with our enum types,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=0.74) [it's important to understand that the values are](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=2.42) [not just a simple list of values.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=4.61) [Those values are actually ordered.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=6.31) [So one value could be considered greater than or less than another value.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=8.7) [Now the first value on the list is going to have the lowest ordering.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=13.64) [The last value on the list is going to have the highest ordering.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=16.75) [And this allows us to do relative comparisons.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=19.59) [So when we declare a variable to be of an enumeration type,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=23.24) [that variable actually holds a reference to that value,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=26.12) [so we can call the compareTo method against one reference](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=29.74) [to compare it to another reference.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=32.69) [Now compareTo returns back either a negative value, zero, or a positive value.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=35.54) [And this indicates the relative ordering in the comparison.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=40.31) [So we call compareTo against one reference, passing in another reference.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=44.14) [And return value indicates how the one you've called compareTo](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=48.3) [against compares relatively to the value you pass in.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=51.79) [So if the return value is negative,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=56.1) [the one you've called compareTo against is ordered](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=57.56) [lower than the value you've passed in.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=60.38) [If the value is positive,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=62.62) [the one you've called compareTo against is ordered](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=64.03) [higher than the value you passed in.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=66.23) [So let's see what this looks like in code.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=69.79) [So we again have our FlightCrewJob enumeration.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=71.62) [We have our three values, FLIGHT\_ATTENDANT, COPILOT, and PILOT.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=74.3) [So since FLIGHT\_ATTENDANT is the first value on our list,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=78.69) [it's considered to have the lowest ordering.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=81.78) [Then we have COPILOT.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=83.94) [So COPILOT would be considered to be ordered higher than FLIGHT\_ATTENDANT.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=85.42) [And then PILOT we'd consider to be ordered higher than](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=90.04) [both COPILOT and FLIGHT\_ATTENDANT.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=92.7) [So we also have a class here, CrewMember,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=95.54) [and CrewMember has a field named job whose type is FlightCrewJob.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=97.95) [Let's go ahead and add another field, name, whose type is String.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=102.38) [We'll give CrewMember a constructor that accepts a job and a name](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=105.8) [and then assigns those values to our fields.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=109.59) [So now we have our CrewMember class,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=112.34) [which is leveraging our FlightCrewJob enumeration type.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=113.9) [So let's go ahead and create some instances of CrewMember.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=118.74) [So we'll create a new CrewMember here.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=121.54) [We'll specify the job as PILOT, and the CrewMember's name is Geetha.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=123.15) [Then we'll create another CrewMember whose job is](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=127.61) [FLIGHT\_ATTENDANT and name is Bob.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=130.3) [Let's go ahead and call a method we'll create called](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=132.89) [whoIsInCharge passing in Geetha and Bob.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=134.77) [So we have our whoIsInCharge method that accepts two parameters,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=137.71) [both are CrewMembers.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=141.39) [These parameters are member1 and member2.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=142.63) [And the job of whoIsInCharge is to figure out which](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=145.22) [of these two members are theBoss.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=148.88) [So we have this local variable here, theBoss, whose type is CrewMember.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=151.21) [So what we'll do is say member1.getJob.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=154.44) [So that would give us a reference to the job field for this CrewMember,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=156.62) [and then we'll call compareTo against that job,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=161.58) [and we'll pass in member2.getJob.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=164.5) [So, basically,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=168.33) [what we're doing here is seeing how the job for](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=169.02) [member1 compares to the job for member2.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=171.97) [And we'll check its return value.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=174.86) [So here we'll check and see if it's greater than 0.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=176.9) [So if the return value is greater than 0,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=179.84) [that means that member1 is ordered higher than member2,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=181.64) [and that's a value we'll assign to theBoss.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=184.47) [Otherwise, we'll assign member2 to theBoss.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=187.44) [So now theBoss will have a reference to the](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=190.54) [CrewMember whose job is ordered higher.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=192.32) [So we could then use theBoss to print out that CrewMember's name.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=194.72) [So in this case,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=199.04) [because Geetha was a PILOT and Bob is a FLIGHT\_ATTENDANT, in](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=199.74) [this case, we'd print out that Geetha is boss.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=203.55) [Now the capabilities of enumeration types go well beyond the values themselves.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=207.94) [There's actually a number of methods that we have access to as part of an](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=212.37) [enumeration type. A couple that we use commonly are values,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=215.32) [which will give us back an array containing all of the](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=219.3) [values for that enumeration type.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=221.57) [And we also have valueOf. ValueOf can actually translate a](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=223.39) [string into the corresponding enumeration value.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=227.15) [So if you type in a string with the value of that enumeration type,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=230.44) [we'll get back the corresponding enumeration. Now it's important to](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=233.47) [note that valueOf is actually case sensitive.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=236.88) [So the case of the string has to match the case that's used for that value.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=239.89) [Since our enumeration values are normally all uppercase, that means the](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=244.62) [string would have to be all uppercase as well.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=248.25) [So to give us a better understanding of all this, in our next clip,](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=251.44) [let's get into our CalcEngine project and see how we can leverage enumeration types.](https://app.pluralsight.com/course-player?clipId=d9933a03-5438-467f-bc23-45e2cb7f9082&startTime=254.18)

[Representing Math Operations as Enums](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651)

[Here we are in IntelliJ,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=0.64) [and what we want to do now is add some interactive](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=1.69) [capabilities into our CalcEngine project.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=4.65) [We want to allow people to type in an operation and some values,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=7.74) [and the application will perform that operation.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=10.36) [And as part of creating this behavior, we're going to use an enumeration type.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=13.13) [So to create our enumeration type, let's head over here to our project window.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=18.09) [Then we'll right‑click on our package name.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=21.84) [We'll head up to New.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=24.24) [Then even though we're creating a new enumeration type,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=26.94) [we're going to choose Java Class. And you'll notice that](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=29.38) [when this New Java Class dialog comes up,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=34.12) [it has a series of options down here at the bottom.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=36.01) [One of those options is Enum.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=38.7) [So we'll go ahead and choose that.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=39.91) [And this enumeration type is going to contain values](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=42.68) [for the four basic math operations.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=45.48) [So we'll name this type MathOperation.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=48.08) [So go ahead and hit Enter.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=51.88) [And then here in the enumeration type,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=53.39) [we just need to add the values we want for this type.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=55.87) [So our first value will be ADD.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=59.13) [So now, remember, as we add these values,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=62.28) [each value has to be comma‑separated, and by convention,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=65.06) [the value name is all uppercase.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=68.29) [So we've got our ADD value here.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=70.36) [Let's go ahead and provide values for SUBTRACT, MULTIPLY, and DIVIDE. So now our](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=72.37) [MathOperation type has values for all four math operations.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=77.87) [Now as you recall in our previous module,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=82.44) [we added classes that could perform each of the math operations.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=84.33) [We had an Adder class, a Divider class, a Multiplier class,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=87.89) [and a Subtracter class.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=91.63) [So we could, of course,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=93.64) [use those classes to perform the operations that correspond to the](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=94.41) [value of our MathOperation enumeration type.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=98.2) [Now remember that one of the things those classes had in common](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=99.95) [is they all inherit from a common base class,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=103.38) [CalculateBase.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=105.74) [So what we could do is have some code that uses the MathOperation](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=107.49) [value to create the appropriate class instance.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=110.52) [And since all those classes have a common base class,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=113.66) [we could put that into a method whose return type was CalculateBase.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=115.94) [So let's add that code over in our Main class.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=119.44) [So I'll go ahead and collapse our project window](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=122.54) [and head over to our Main class.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=124.36) [So now we're here in our Main class.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=127.54) [So let's go ahead and add a method named createCalculation](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=129.08) [with a return type of CalculateBase.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=131.88) [So now our first parameter to createCalculation will be of type MathOperation,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=135.54) [and we'll name it operation.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=139.72) [So that allows us to specify what operation we want](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=142.74) [to perform. Now as you may recall, as we create each of these classes,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=145.41) [one of the constructors they have supports passing in the](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=149.71) [leftVal and rightVal. So let's go ahead and include](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=152.68) [parameters for those two values.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=155.32) [So now our createCalculation method accepts parameters that indicate the](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=158.44) [operation and the two values that we want to operate on.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=162.13) [So now inside of here, we can do the work of creating the different types.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=165.88) [So now the first thing we'll do is create a local variable](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=169.64) [named calculation of type CalculateBase,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=171.8) [and we'll use our calculation variable to hold a reference](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=175.74) [to whichever class instance we create.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=178.62) [Now remember that our MathOperation is an enum. And since it's an](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=181.3) [enum, we can actually perform a switch on it. So let's add in a](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=183.34) [switch for our operation parameter.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=188.05) [Then within our switch, we can have a case for each of the values.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=191.39) [So let's start with a case for ADD. And in the case of ADD, we'll create a new](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=195.14) [instance of our Adder class passing in leftVal and rightVal,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=200.17) [and we'll assign it to our local variable calculation.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=204.09) [Then after we create the instance of the Adder class,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=208.5) [we'll put in a break statement.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=210.67) [So with that, when we receive an operation of ADD,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=213.24) [we take care of creating an instance of our Adder class.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=215.73) [So now let's do the same sort of thing for SUBTRACT, MULTIPLY, and DIVIDE.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=218.64) [So now with that, our switch statement handles all four values in our](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=224.19) [MathOperation type, and in each case creates an instance of the](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=227.44) [appropriate class and assigns that reference to a local variable](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=230.93) [calculation. So the last thing we'll need to do here is have our](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=234.13) [createCalculation method return calculation.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=237.57) [So now our createCalculation method is complete.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=241.74) [We passed in an operation and two values,](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=244.44) [and we get back a reference to a type that will perform that operation.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=246.61) [So now we're ready to add our interactive capabilities, and we'll do that in our next clip.](https://app.pluralsight.com/course-player?clipId=7e5937ac-b567-4c05-8fe6-c798ed0fe651&startTime=250.94)

[Processing Interactive Input in CalcEngine](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854)

[Here we are back in IntelliJ,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=0.64) [and what we want to do now is finish the job of adding](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=2.26) [interactive capabilities to our application.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=5.12) [Now I've actually added a little bit of code in here to get us started, and](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=7.69) [notice we have this method here, executeInteractively.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=10.38) [And this method takes care of the details of prompting the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=13.64) [user and getting that user's input back.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=15.94) [And the technique I'm using here is the same technique we used in our](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=18.57) [earlier course, Getting Started with Programming in Java.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=21.61) [Now let me just walk through it quickly as a reminder.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=24.6) [Now as the first thing we do is print out a message of the user to tell them](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=26.97) [to add an operation and two numbers. And then we use this type named scanner](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=30.49) [to actually get back the input from the user.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=34.7) [So we call this method nextLine. That gives back a string containing all the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=37.68) [values the user typed in up until they hit the Enter key.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=41.71) [So what we're expecting the user to type in is something](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=45.44) [like add 10 20, an operation name, along with two values,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=47.9) [with each separated by a space.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=53.76) [So then what we do is use the split method to break](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=56.24) [that string up into its three parts.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=58.5) [So if the user types in add 10 20, this parts array will have three elements.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=61.04) [Element 0 will be the string add, element 1 will be the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=66.6) [string 10, element 2 will be the string 20.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=70.42) [So we then take our parts array and pass it into a](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=74.04) [method we have here, performOperation.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=76.33) [So we're going to do our work to process that input here in performOperation.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=79.24) [So now the first thing we need to do is identify the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=83.64) [operation that the user wants to execute.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=85.83) [So let's declare a local variable here named operation of type MathOperation.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=88.34) [Now remember that a string identifying the operation is in part subzero.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=93.64) [And as we mentioned in the slides,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=97.7) [enumeration types can actually translate strings into the equivalent](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=99.53) [enumeration value, and we do that with the method valueOf.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=103.02) [So let's call MathOperation.valueOf, passing in parts subzero.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=106.94) [So calling valueOf will attempt to translate that string into the corresponding](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=113.04) [MathOperation value, but remember that valueOf is case sensitive, and all our](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=116.96) [operation names are all uppercase. So we want to make sure that this string that](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=121.67) [we're processing is also in uppercase. So to take care of that, we'll call](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=125.57) [toUpperCase against parts subzero.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=129.41) [So now, no matter what case the user used when typing in the operation,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=133.34) [we'll process it as uppercase,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=137.39) [and that should give us back the appropriate operation.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=139.46) [So now the next thing we need to do is translate the strings](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=141.94) [containing the values into doubles. So let's declare local](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=144.67) [variable named leftVal of type double.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=148.06) [And to do the translation,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=151.44) [we use the same technique we used in our previous course.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=152.77) [Remember, there's a class named Double, and it's Double spelled with a capital D.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=155.84) [Well,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=160.14) [the Double class has a method parseDouble that can translate a](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=160.24) [string into its equivalent double value.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=163.15) [So we'll call Double.parseDouble, passing in (parts[1]). So that](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=166.14) [will set our leftVal to the appropriate value.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=171.49) [So we'll do the same thing for our rightVal with (parts[2]).](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=174.54) [So that gives us our operation and both values.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=178.74) [So now we can use our createCalculation method, and we'll](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=181.45) [pass in operation leftVal and rightVal.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=184.1) [Now remember, when we wrote the createCalculation method,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=188.54) [we receive the operation in both values.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=191.41) [We then create an instance of the appropriate class, passing back a](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=193.67) [reference to it as a CalculateBase reference.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=196.77) [So let's assign the result of createCalculation to a local](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=200.04) [variable name calculation of type CalculateBase.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=202.41) [So once we get back our reference into calculation, we can call calculate.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=207.04) [Now remember when we call calculate, it performs the appropriate operation](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=212.24) [and stores the outcome of that operation into its result field. So that](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=215.98) [means we've actually done all the work at this point. The only thing we](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=219.4) [need to do now is print out the outcome.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=222.72) [So let's print out a couple things.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=225.21) [The first thing we'll do is print out the value in our operation variable.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=226.94) [And then after that,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=231.74) [we'll go ahead and print out the result, and that's all there is to it.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=232.53) [Basically,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=237.04) [the executeInteractively method took care of getting the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=237.49) [input from the user and passing the three parts of the input](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=240.15) [into our performOperation method.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=243.11) [Here inside performOperation, we determine the operation and the two values.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=245.24) [We then got a reference to the appropriate type to perform that operation](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=250.29) [called calculate to do the work and then simply printed out the name of the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=253.97) [operation and the result of the calculation.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=258.34) [So with that all done, let's go ahead and run our application,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=261.84) [and we'll verify that everything works.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=264.21) [So now if we look down here in the Run window,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=267.24) [we can see our prompt to enter an operation and two numbers.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=269.1) [I'll go and click in the Run window to give it focus,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=272.8) [and that's going to enter multiply 5 10.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=275.91) [So now we've our operation and both values with each part separated by a space,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=280.44) [so I'll go ahead and hit Enter.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=285.34) [And when I do that,](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=288.14) [we see the output is Operation performed: MULTIPLY, and the result is 50.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=289.18) [So our application worked perfectly.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=294.24) [So by using our enumeration type, implementing this behavior was very simple.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=296.24) [We were able to easily get the operation from the user and then create the](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=300.04) [appropriate type that corresponded to that operation.](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=303.66) [Alright, so now in our next clip, let's take a look at some of the ClassBase features of enumeration types](https://app.pluralsight.com/course-player?clipId=0121334d-e423-4d14-96a3-e42cbe384854&startTime=307.44)

[Enum Types Are Classes](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997)

[And although it may not look like it, when declaring an enum type,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=0.74) [we're actually declaring a class, but there are some key](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=3.38) [differences from other classes we declare.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=6.16) [One key difference is that they implicitly inherit from Java's](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=9.04) [Enum class, and this provides many of the special capabilities](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=12.25) [that we associate with enum types.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=15.53) [Now there are many aspects of enum types that are similar to other classes,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=17.94) [but there's some really key characteristics that are special about enum](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=21.52) [types. Now in terms of similarity with other classes, one of the key things](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=25.44) [is enum types can actually have members.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=31.1) [So that means that our enum types can have fields, our enum types can have](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=33.94) [methods, and our enum types can even have constructors,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=37.84) [which means we can actually pass in initial state when we're](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=41.02) [creating an instance of our enum type.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=43.85) [But now one of the really key differences about enum types is](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=47.14) [the way we create instances of our enum types.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=50.52) [It turns out that each value within an enum type is](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=53.19) [actually an instance of that enum type.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=57.19) [So that means that simply declaring the value creates the instance. And as](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=60.22) [part of declaring that value, we can even leverage a constructor. So as part](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=64.58) [of declaring the value, we can pass initial state into that instance as we](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=68.43) [declare it. Now this all may sound really confusing, so I think the best](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=73.09) [thing for us to do is look at some code. So we'll get a look at our](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=78.32) [FlightCrewJob enum type.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=81.59) [Remember, we have three values here, FLIGHT\_ATTENDANT, COPILOT, and PILOT.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=83.66) [Now, so far, we've only listed the values.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=88.34) [So let's get ready to add some members to this FlightCrewJob](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=91.31) [enumeration type. Now one key thing to note. If we're going to add](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=94.26) [members to our enumeration type, here at the end of the list of](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=98.01) [values we need to provide a semicolon.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=100.85) [So when we were just listing the values, we didn't need the semicolon, but if](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=103.84) [we're going to add members, we do need to have a semicolon at the end of the](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=107.73) [list. Now in terms of the members we want to put inside of this type, pretty](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=110.82) [much any members you would put into a class.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=115.27) [So let's go and add a field to our enumeration type. So we](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=117.42) [have this field title, which will be the title we want to](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=120.42) [associate with a particular job.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=122.73) [For example,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=124.64) [the pilot's title might be captain. Now we could have](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=125.11) [more fields here if we needed to, but in this case,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=128.26) [we just need one. Let's go and add a method, getTitle, so](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=130.34) [that will be a getter for our title field.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=134.22) [And then we need some way to specify the title,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=136.94) [so we'll do that with our constructor.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=139.74) [So we have a constructor here that accepts title as a](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=141.35) [parameter, and that will set our title field.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=143.46) [Now as we mentioned,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=146.89) [the values that we list here are actually instances of](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=147.93) [our FlightCrewJob enumeration type.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=151.04) [So it's these values themselves that are creating the instance.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=153.61) [So with each of these values, we can actually call the constructor.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=156.94) [So in the case of PILOT, we want the title to be Captain.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=160.84) [So after the value PILOT, we'll pass in the parameter](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=164.16) [we want to pass to the constructor,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=166.57) [which is the string, Captain. And then here for our COPILOT, we'll set](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=168.12) [that title to be First Officer. And then here for our FLIGHT\_ATTENDANT,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=172.46) [we'll set that title to be Flight Attendant.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=176.48) [So now here within our FlightCrewJob, we have three instances of FlightCrewJob,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=179.24) [FLIGHT\_ATTENDANT, COPILOT, and PILOT, and in each case,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=183.95) [we've passed in the title we want to associate with that instance.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=187.25) [So now let's revisit some code that we wrote earlier. As you recall,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=192.19) [we had that method we wrote, whoIsInCharge.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=195.6) [And when we were using whoIsInCharge,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=198.44) [we created one instance of a crew member who was a pilot name](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=200.22) [Geetha, then we created another instance of a crew member who](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=202.9) [was a flight attendant named Bob, and we called whoIsInCharge,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=206) [passing in Geetha and Bob. Then down inside of whoIsInCharge, we used compareTo](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=209.72) [to figure out which one of those two members is the boss.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=215.47) [And then once we determined who is the boss,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=218.84) [we wrote out a message to that effect.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=220.6) [So in this case, because Geetha is a pilot, we wrote out, Geetha is boss.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=222.9) [So now let's see how we can leverage the new capabilities](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=228.64) [we've added to our type FlightCrewJob.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=230.95) [So up here before we print out the name, we'll use our](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=233.78) [theBoss reference to get access to the job.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=236.46) [Now remember, that'll give us back a reference to pilot,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=239.9) [and pilot is an instance of FlightCrewJob, and](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=242.66) [FlightCrewJob has a method, getTitle.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=246.07) [So this will give us back the title that's associated](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=248.94) [with that instance of FlightCrewJob,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=251.55) [which in this case, would be Captain. Then after the](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=253.94) [title, let's go and concatenate a space.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=256.71) [So now as we print out the message, rather than simply being Geetha is boss,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=259.14) [the message will now be Captain Geetha is boss.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=263.7) [So now,](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=267.64) [thanks to the enhancements we've made to our FlightCrewJob](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=268.25) [enumeration type, each job beyond simply being a value, actually has state associated with that value.](https://app.pluralsight.com/course-player?clipId=c843211b-5013-4212-990a-caf2e29df997&startTime=270.64)

[Summary](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2)

[To wrap up, here are some of the key things you](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=0.74) [want to remember from this module.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=2.93) [Remember that as we talk about enumeration types, we're talking about types](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=4.69) [that allow us to define a finite list of valid values.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=8.36) [In other words,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=12.14) [we're in effect creating our own type and specifying what values](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=12.93) [can be assigned to variables of that type.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=16.31) [Now as we work with our enumeration types,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=19.24) [we're going to need to perform conditional logic.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=20.95) [We want to check to see what values are contained where. Now we can](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=23.36) [do simple equality tests or inequality tests,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=26.88) [comparing a variable to one of the values in the enumeration type,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=30.17) [as well as comparing two variables against one another.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=33.5) [And oftentimes as we work with enumeration types, we want to use switch](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=36.58) [statements, and switch statements work really well with enumeration types](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=40.09) [because it allows us to branch our logic depending on which of our enumeration](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=43.88) [values are contained within a particular variable.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=48.23) [Now remember that the values that we specify are ordered.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=51.5) [The first value we list has the lowest ordering,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=55.04) [the last value we list has the highest ordering,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=57.92) [so we can perform order‑based comparisons.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=60.94) [We do that by using the compareTo method.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=63.63) [Also remember that enum types are actually classes. They](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=67.54) [implicitly inherit from Java's Enum class.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=71.63) [And we can do many of the things with enum types that we](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=75.14) [do with other classes, for example,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=77.62) [we can define members. So our enum types can have fields, they can](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=79.61) [have methods, they can even have constructors.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=83.21) [Now when we want to create an instance of our enum type, we](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=86.14) [don't do that by using the new operator.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=88.81) [Instead,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=91.04) [we rely on the values that we specified for that enum type, because](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=91.7) [the values are actually instances of that enum type.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=95.9) [So simply declaring a value within the enum type creates an](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=99.74) [instance of that enum type, and we can specify the state of that](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=102.68) [instance by leveraging our constructors.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=106.88) [So as part of declaring the value, we can specify the parameters](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=109.44) [that we want to pass into that constructor.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=112.86) [Alright,](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=115.74) [that wraps up this module. In our next module, we'll see how we can create abstract relationships using interfaces.](https://app.pluralsight.com/course-player?clipId=3a7e4bb2-f508-4dfa-a304-226a34c973c2&startTime=116.47)

[Creating Abstract Relationships with Interfaces](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d)

[Introduction](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d)

[Welcome to our next module Creating Abstract Relationships with Interfaces.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=0.64) [This is part of the Pluralsight course Working with Classes and](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=4.67) [Interfaces in Java, and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=6.92) [Throughout this course we've been focused very much on concrete types, that](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=11.04) [is classes and using inheritance to extend those classes.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=14.38) [But there are times where we need to move beyond concrete relationships and](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=18.34) [instead use abstract relationships. In other words, relationships that allow](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=22.06) [us to specify rules for conformance without giving you the details of how we](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=26.31) [actually implement those rules.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=30.21) [And that's where interfaces come in.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=31.81) [Interfaces allow us to define abstract relationships.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=33.84) [In a sense,](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=37.26) [an interface allows us to define a contract for interoperability.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=37.93) [So throughout this module we'll be looking at interfaces. So we'll](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=42.09) [start out and look at why we need more than inheritance to build](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=45.24) [on existing functionality.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=48.7) [Why do we need these abstract relationships known as interfaces?](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=50.34) [From there,](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=53.84) [we'll take a look at what an interface is. We'll understand how we](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=54.15) [can implement an interface on one of our classes.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=56.49) [The next thing we'll look at is what's known as generic interfaces.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=59.24) [Generic interfaces allow us to specialize an interface for a](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=62.58) [particular type. So it allows us to have these abstract](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=65.69) [relationships that are tied to a specific type.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=68.26) [We'll then look at how we implement multiple interfaces. One of](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=71.6) [the key powers of interfaces is one class can implement multiple](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=74.68) [interfaces. So one class is able to conform to multiple](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=78.15) [contracts for interoperability.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=81.62) [Next, we'll see how we can declare interfaces.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=84.04) [In other words, we'll see how we can specify our own contracts for](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=86.6) [interoperability. And one of the challenges we have when we specify interfaces](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=89.64) [is that sometimes we need to change those interfaces over time,](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=94.1) [which always runs the risk of breaking existing code.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=97.14) [So that's a really serious challenge.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=100.34) [So we'll finish up, we'll look at default methods, and we'll see how default methods allow us to overcome that challenge.](https://app.pluralsight.com/course-player?clipId=3e8bad76-ee84-4a95-9d91-43808c8f3f9d&startTime=102.08)

[Understanding Interfaces](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d)

[As software developers, we want to be as effective as possible,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=0.69) [and a key part of being effective as a software](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=4.14) [developer is relying on reusability.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=5.97) [We don't want to spend the time and the effort to build](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=8.74) [something new when a good solution already exists.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=10.95) [Now a key part of usability is, of course,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=14.44) [class inheritance because class inheritance allows one class](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=16.58) [to leverage the implementation of another.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=20.84) [So if you have an existing class that does mostly what you](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=23.08) [want, you can inherit from that existing class and then](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=25.43) [specialize it for your current need.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=28.29) [But inheritance has some limitations.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=30.94) [Remember that in Java, a class can directly extend only one other class.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=33.64) [Now the class that it extends can, of course, extend other classes.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=39.14) [But one class cannot directly extend two or more classes.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=42.46) [You can only extend one class,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=46.1) [and that constrains the amount of reusability we can access using only](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=48.38) [inheritance because with inheritance all our reusability would have to be](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=52.06) [in base classes, and these base classes could become unmanageably large if](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=55.62) [we had to put every feature and every function that we needed in the](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=59.79) [hierarchy of base classes. And that's where interfaces come in. Interfaces](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=62.7) [allow to define a contract for behavior, and an interface simply provides a](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=68.45) [list of operations and the behavior that goes along with those operations.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=73.76) [But interfaces do not focus on implementation details.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=78.24) [They simply provide the methods and the contract for their behavior.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=82.54) [Then from there, classes can then implement interfaces.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=86.39) [So when a class implements an interface, it expresses](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=90.34) [that it conforms to a contract.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=93.02) [But the details of how an individual class conforms to](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=95.44) [the contract is up to the class itself.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=97.74) [The class implements the methods that are part of the interface](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=100.34) [in a way that's appropriate for that class.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=103.16) [I think really the best way to understand interfaces is with an example.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=107.04) [So we're going to look at an interface called a Comparable interface.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=110.84) [And the Comparable interface is a very widely used interface in Java. The](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=114.08) [Comparable interface is used to solve a particular challenge.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=118.38) [That challenge is that objects often need to be ordered.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=121.84) [In other words, we often need a way to sort different objects.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=125.74) [But the rules we're ordering are different for each class. Depending on the](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=129.04) [data inside of a class and the purpose of that class,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=132.72) [the way it's ordered is going to be different from other classes. So](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=135.36) [that's where the comparable interface comes in.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=138.63) [The comparable interface provides a contract for ordering,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=141.09) [and by using this contract, we get the key benefit of being able to have](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=144.74) [broadly reusable sorting utilities with those utilities knowing nothing at](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=148.24) [all about the classes that they need to sort.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=152.04) [The simple fact that the class implements the Comparable](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=154.94) [interface is all that's needed for the class to be able to](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=157.1) [leverage one of these utilities.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=159.91) [So let's take a closer look at the Comparable interface.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=162.94) [Now this interface has just one method.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=165.6) [That method is the compareTo method. The compareTo method has one parameter,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=167.94) [which is a reference to another object. And the purpose of the compareTo](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=172.85) [method is to indicate the relative relationship between the current object](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=176.85) [and the object that it receives as a parameter.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=181.59) [In other words,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=183.91) [it indicates the ordering between the current object and the](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=184.46) [received object, and the way it indicates this ordering is by the](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=187.2) [return value of the compareTo method.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=190.63) [So if the compareTo method returns back a negative value, that indicates](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=193.64) [that the current object is ordered first. In other words,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=197.58) [the current object comes before the received object.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=200.41) [If the compareTo method returns back a positive value, that](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=203.74) [indicates that the received object is ordered first.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=206.76) [In other words, the received object comes before the current](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=209.87) [object. And if the return value is zero,](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=212.49) [then the two objects are equal.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=215.19) [So in our next clip, let's take a look at an example of implementing the Comparable interface.](https://app.pluralsight.com/course-player?clipId=0e0da572-3822-4e5c-961c-644c6dec802d&startTime=217.64)

[The Comparable Interface](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a)

[Let's take a look now at how we can leverage the](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=0.74) [capabilities of the comparable interface.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=2.51) [And to do that, we'll use our Passenger class.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=5.24) [So let's add some new fields to our Passenger class.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=7.94) [Now one of the fields will be name,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=10.3) [which is a passenger's name and we'll have another field,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=11.79) [member level,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=14.59) [which indicates what level of membership a passenger](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=15.76) [has in the airline's loyalty program.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=17.78) [We often think of it as a frequent flyer program.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=20.55) [Level three members of the highest level members,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=23.44) [the ones most entitled to benefit, level one are the lowest level members,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=25.44) [those least entitled to benefits.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=29.43) [And in addition to the membership level,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=31.74) [let's also keep track of how long a passenger has been a member.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=33.55) [We'll track that information in terms of number of days.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=36.54) [So let's go now to constructor to our Passenger class that accepts values](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=39.91) [for all those fields and then sets the appropriate fields.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=43.04) [So now let's add some ordering capabilities for our Passenger class. We want](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=46.68) [to order passengers based on their membership information.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=50.34) [Now the primary way we want to order members is based on membership level.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=53.98) [Now remember that level three members are the highest priority members,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=58.09) [so level three members should always be the first ones to receive any benefits.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=61.64) [Then our level two members would come after our level three members,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=66.04) [and at the end of the list would be all the level one members.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=69.17) [But of course,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=72.74) [one of the challenges we have is that we only have three membership levels,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=73.11) [but a flight can have hundreds of passengers on it.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=77.57) [So there is a very strong probability that on any given flight there will](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=80.54) [be multiple members with the same membership level.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=84.16) [So the deal with ties in the member level will use member days.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=87.38) [So that's our ordering behavior.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=91.63) [I want to use the member level is the primary way we ordered things](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=93.54) [and then using member days as a way to break a time.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=96.7) [And we want to be out of order our passengers by doing the least](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=99.64) [amount of work that we reasonably need to.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=102.2) [So here in our Passenger class,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=104.53) [what we're going to do is implement the comparable interface and again,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=105.96) [what this means, by implementing this interface,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=110.08) [we conform to the contract to be able to compare a current](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=112.5) [passenger instance to another passenger instance and](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=116.04) [indicate which one is ordered first.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=118.5) [Now remember,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=121.47) [the comparable interface has one method and that's the compareTo method.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=121.76) [Now compareTo receives one parameter and we're going to](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=126.14) [receive that parameter as an object,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=128.61) [but it's going to be a reference to another passenger. So the first thing we'll](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=130.64) [need to do is cast that object to be a passenger reference.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=134.12) [So now once we have this passenger reference p,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=138.24) [we can compare the current passenger instance to the](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=140.75) [passenger that's referenced by p.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=143.42) [Now we could do these comparisons with a series of explicit](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=146.4) [ifs by comparing the fields between the current passenger](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=149.51) [and the passenger we received.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=153.24) [But remember,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=155.14) [the contract of the comparable interface is simply to return back a negative,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=155.54) [positive, or a 0 value.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=160.29) [With that being the case,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=162.69) [there is a much simpler way to determine ordering of](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=163.92) [passengers than doing explicit ifs.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=166.22) [We can actually use arithmetic. So what we could do here is declare a local](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=168.68) [variable named returnValue and then simply take the member level of the](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=172.87) [passenger we received and subtract from it the current passengers member](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=176.82) [level. Now let's think how that would work. Well if the passenger received](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=180.85) [at a member level of 1 and the current passenger has a member level of 3,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=185.45) [well 1 minus 3 is ‑2.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=190.04) [So if we return that value back, that would be a negative value,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=193.73) [which correctly indicates that the current passenger is](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=197.22) [ordered before the passenger we receive.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=200.39) [So this simple bit of arithmetic takes care of indicating](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=203.34) [which of the passengers is ordered first.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=206.04) [But now what happens if the current passenger and the](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=208.94) [passenger we receive have the same member level?](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=211.53) [Well, in that case, we would calculate a return value of 0.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=214.64) [So in that case,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=218.34) [what we do is simply fall back to doing the arithmetic using member days.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=219.19) [So with that in place, if the current passenger has been a member longer,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=224.44) [we'll correctly return a negative value.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=227.87) [If the passenger received has been a member longer, we'll](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=230.34) [correctly return a positive return value.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=232.73) [So, as you can see,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=235.74) [this simple bit of arithmetic takes care of doing the work](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=236.49) [of indicating ordering using a positive,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=239.63) [negative, or a 0 value.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=242.08) [So now that we have our implementation of the comparable interface in place,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=245.24) [we can now efficiently order passengers.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=248.82) [So let's say we have an array of passengers here named passengers and](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=251.94) [we'll initialize it with four instances of our Passenger class. Now](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=255.74) [currently these passenger instances are in the array in no particular](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=259.92) [order and what we'd like to do is get them back ordered by our](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=263.64) [membership information.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=267.56) [So to do that,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=269.24) [we can use a class in Java named Arrays and the Arrays](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=270.13) [class has a method on it named sort,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=273.04) [and then to our sort method, we're going to pass in our passengers array because](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=275.69) [if we were to look up the documentation on the sort method,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=280.03) [it tells us that it will order the members of the array](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=282.86) [based on the comparable interface.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=285.86) [So what that means is when Arrays.sort returns,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=288.89) [the elements of our passengers array will be reordered and it will order them](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=292.14) [based on how we've implemented the comparable interface.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=297.34) [So, looking at the array members,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=300.88) [we have Ashanti who is the only level three member.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=302.74) [So since she's the highest level member,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=306.64) [the first element of the passengers array will be Ashanti.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=309.04) [So next, we have Harish. Now Harish is the only level two member.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=312.59) [So that means the next member of this sorted passengers array will be](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=317.04) [Harish, but then we have both Luisa and Jack and both of them are level one](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=320.16) [members. Well remember, the way we've implemented our compareTo method, if](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=325.28) [two members have the same member level, we use the member days.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=329.34) [So since Luisa has been a member longer,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=333.48) [Luisa will come after Harish, and then finally,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=336.31) [Jack, who's only been a member for 90 days,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=339.86) [will come at the end of the now sorted array.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=342.54) [And as you can see, this really shows off the power of interfaces.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=345.74) [By implementing the interface and conforming to the contract of the interface,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=349.78) [we were able to leverage the capabilities of this arrays.sort method,](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=354.21) [even though that method knows nothing about our Passenger class.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=358.34) [Okay, so now in our next clip, let's take a look at something known as generic interfaces.](https://app.pluralsight.com/course-player?clipId=794bea0d-6c9c-46b6-88cc-c69db09d9c4a&startTime=362.84)

[Generic Interfaces](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f)

[As we're working with interfaces,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=0.74) [it's important to note that some interfaces allow being](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=2.22) [specialized for a particular type,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=5.15) [and they do this using a concept in Java known as generics.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=7.41) [So if we were to take a look at the comparable interfaces declaration,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=11.34) [we'd notice that after the interface name,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=14.4) [there is a less than symbol, a T, and then a greater than symbol,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=16.48) [and that indicates that this interface could be more strongly typed.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=20) [And what that means is the method within the interface](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=23.64) [doesn't have to use an object type, you can actually use a more specific type,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=26.22) [so let's again look at our Passenger class and we know our](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=31.64) [Passenger class implements the comparable interface,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=34.08) [so we have our compareTo method, and the compareTo](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=36.94) [method as we currently have it written, receives an object as a parameter.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=39.41) [Now the name of that object is o,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=43.24) [and because we want to compareTo passenger references,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=44.93) [the first thing we have to do here is cast that object to be a passenger,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=47.78) [and then once we do that,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=52.14) [we can do the actual work we want to do of comparing the current](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=53.19) [passenger instance to the passenger instance we received. Well up here,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=56.11) [we say we're implementing comparable. Because the comparable interface](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=60.05) [supports generics, after the interface name,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=63.51) [we can specify that we want to specialize on the type passenger,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=66.23) [and by doing that,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=70.54) [the compareTo method here where we receive an object can](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=71.72) [instead directly receive a passenger reference,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=74.77) [which means the first thing we're doing here where we cast one](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=78.34) [object to a passenger is no longer necessary.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=80.61) [So our method compareTo is now more strongly typed,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=83.94) [which means we get all the benefits of strong typing within the](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=87.64) [compiler. The compiler will make sure that we never passed anything](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=89.96) [other than a passenger reference into compareTo, and generics are a](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=93.37) [really powerful concept in Java.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=97.66) [So once you've had a chance to work with Java for a while,](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=100.04) [I would encourage you to check out the Pluralsight class on Java](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=102.52) [generics to get a better understanding of this concept.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=105.17) [You can do that by heading over to the Pluralsight search facility just](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=108.04) [typing in Java generics. Alight, so now in our next clip, let's take a look at how a class can implement multiple interfaces.](https://app.pluralsight.com/course-player?clipId=a782a230-c3de-4f98-aad8-f8d4228cdc0f&startTime=111.08)

[Implementing Multiple Interfaces](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7)

[Now as a class implements interfaces,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=0.74) [it's not limited to implementing just a single interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=2.42) [A class can implement multiple interfaces.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=5.83) [Now to do that,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=8.84) [we simply list the interfaces we want to implement separated by commas.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=9.7) [It's important to note that there's no practical limit on the](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=13.44) [number of interfaces a class can implement,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=16.54) [so a class can conform to as many interface contracts](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=18.94) [as is appropriate for that class.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=21.9) [So let's see what it looks like to implement multiple interfaces.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=25.64) [So this time we'll take a look at our Flight class.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=28.84) [Remember, our Flight class has a number of fields,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=31.2) [including number of passengers and the number of seats.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=33.07) [And let's assume that our Flight class has already](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=36.54) [implemented the Comparable interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=38.46) [As part of implementing the Comparable interface,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=40.39) [the Flight class would include the compareTo method.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=42.55) [It's not important what the details are of how this](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=45.94) [particular class implements compareTo.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=47.93) [The thing we're focusing on here is that it's already implemented one interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=49.64) [So let's say we want to add some additional information to our Flight class.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=53.7) [In addition to keeping track of the number of passengers,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=57.3) [we also want to keep track of the individual passengers that are on the flight.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=59.6) [So, effectively,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=63.07) [we want to have a complete list of all the passengers](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=64.4) [that are currently on the flight.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=66.67) [You'll notice that my passengerList field here has](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=68.09) [a type we haven't seen before, which is an ArrayList.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=70.47) [An ArrayList is just a class in Java that could be used very much like an array,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=73.64) [but it automatically increases its size as we add more members to it.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=78.16) [So unlike a simple array, where you have to pre‑size it,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=82.14) [the ArrayList grows automatically.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=85.04) [So our passengerList here will have the full list of](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=87.37) [all the passengers on a flight.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=90.42) [Now you'll notice that this field is currently declared as public.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=92.74) [And that might be kind of dangerous because that means that any](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=95.74) [code can directly access our list of passengers,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=98.41) [which means that additional passengers can be added to the list or taken](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=101.43) [away from that list without our Flight class ever knowing.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=104.72) [So this isn't the sort of thing we want to make public.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=108.14) [We probably want to go ahead and keep this guy as private.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=110.11) [Now that might create some challenges because something we're very](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=113.04) [likely to want to do is be able to walk through that full list of](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=115.35) [passengers that are currently on the flight,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=118.5) [so we might want to use a for each statement to do that sort of thing.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=121.04) [But with this field being private, that's not possible.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=124.24) [But we can address that by having our Flight class implement another interface,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=127.04) [which is the Iterable interface because it's the Iterable interface](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=131.29) [that actually makes the for each statement possible.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=135.47) [Now one thing to note here before we get into the details of this interface,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=138.14) [remember that when we're implementing multiple interfaces,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=141.7) [we want to separate those interfaces by a comma.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=143.51) [So we're saying here that the Flight class implements both the](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=146.64) [Comparable interface and the Iterable interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=148.89) [Now this Iterable interface also supports generics,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=152.74) [so we're going to specialize this on the type Passenger.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=155.3) [So that means we can actually walk through a list of](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=159.08) [passengers using the Flight class.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=160.78) [Now this interface has one method, which is the Iterator method.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=163.8) [This is responsible to return back a reference to](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=167.28) [something known as an iterator.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=169.65) [No, an iterator is something that knows how to walk through a list of items.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=171.08) [Now, in our case, implementing this method is really simple.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=175.44) [Remember that you can use a for each statement to walk through an array.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=178.7) [You can also use a for each statement to walk through an ArrayList.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=181.56) [So all we need to do is ask the passengerList field,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=184.64) [which is an ArrayList, to return back an iterator.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=187.34) [So what we've done here is provided the ability to walk through a list of](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=190.37) [passengers without directly exposing the passengerList.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=194.51) [And we do that by using the Iterable interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=198.59) [So let's see how we could use this now that we've implemented the interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=202.94) [So let's declare a Flight instance.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=206.06) [We'll assign it to a reference f175.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=207.61) [And we'll add a series of passengers to that flight.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=210.82) [So we have a passenger named Santiago, another named Julie,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=213.44) [another named John, and another name Geetha.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=216.43) [So now that we've added the passengers to the flight,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=219.58) [we can actually walk through the list of passengers with a for each statement.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=221.65) [So as part of our for loop, we'll declare a variable name p of type Passenger.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=225.44) [Then we'll simply specify our flight f175. And then inside the](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=229.54) [loop, we'll print out the name of each passenger,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=233.93) [and with this code in place, we would print out Santiago,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=236.74) [Julie, John, and Geetha.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=239.99) [So now the question may come to mind, How does this code](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=243.04) [now walk through the list of passengers?](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=246.03) [Let's take a closer look at the for loop itself.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=248.21) [Let's go ahead and expand that to what actually happens under the covers.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=251) [So let's add some pseudo code here that shows what goes on with the for](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=254.23) [each statement. Now to be really clear, the code I'm going to show you](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=257.35) [is not the exact code that's produced,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=260.9) [but it demonstrates what goes on when we use the for loop](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=263.44) [to walk through a list of passengers.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=266.31) [So the first thing we want to do is take our f175 reference,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=268.54) [and we're accessing it as the Iterable interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=271.87) [Remember, that's the interface that we just implemented on our Flight class.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=275.34) [One of the methods on the Iterable interface is the Iterator](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=279.04) [method. So we call that Iterator method,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=282.02) [which gives us back the iterator.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=284.38) [So once we have that iterator, Java can then generate a loop that will walk](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=286.26) [through the list of passengers because an iterator has a method that tells](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=289.93) [you if there's anything more in the list, and that's the hasNext method. And](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=293.52) [if the hasNext method returns True,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=297.7) [there's another method called next that will give](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=299.76) [you the next item from the list.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=302.23) [So the next method will give us back a reference to the Passenger,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=304.54) [which we can then print out the name of that passenger.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=308.11) [And we got all this capability by implementing another interface on our class.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=311.34) [So the key thing to understand here is that,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=317.69) [unlike classes where each class can only directly extend one](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=319.65) [class, when it comes to interfaces, a class can implement as](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=323.37) [many interfaces as it needs.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=326.62) [So our Flight class implements the Comparable interface,](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=328.64) [and it meets the requirements of that contract by](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=331.66) [implementing the compareTo method.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=333.91) [And it can also implement the Iterable interface with the contract associated](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=336.04) [with that interface by implementing the Iterator method.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=340.05) [So this allows us to implement classes that can conform to the](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=343.14) [requirements of many different interfaces.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=346.36) [Okay, so now in our next clip, let's see what's involved in declaring our own interface.](https://app.pluralsight.com/course-player?clipId=288a5978-0c38-4633-a624-714e06cccad7&startTime=349.34)

[Declaring an Interface](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809)

[As we're developing applications we can actually declare our own interfaces.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=0.74) [In other words,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=5.31) [we can actually declare our own sort of contracts that](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=5.78) [we expect other code to conform to.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=8.27) [Now declaring an interface is very similar to declaring a class.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=10.74) [The key difference is rather than using a class keyword](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=14.01) [we're going to use the interface keyword.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=16.79) [Now when you declare an interface,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=20.04) [most commonly what you include is one or more methods.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=21.45) [And when you specify the method you're generally just going to provide its name,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=25.64) [the list of parameters, and its return type.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=28.9) [We're not going to provide an implementation for the method](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=31.46) [because we expect the classes that implement our interfaces to](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=34.08) [provide their own method implementations.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=37) [Now it's important to note that methods in an interface are implicitly](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=39.84) [public and interfaces are not limited to methods.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=42.98) [Interfaces can also have fields, but those fields are constant fields.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=46.8) [In other words,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=51.11) [they're just named type values because any field we declare as part of](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=51.8) [an interface is implicitly not only public but also final and static,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=55.72) [meaning that any value we provide as part of the interface declaration](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=61.14) [is a value that field will always have even as other classes implement](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=63.99) [the interface.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=67.89) [Now an interesting note on working with interfaces, one interface can extend](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=70.18) [another interface. And we do that by using the extends keyword.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=74.65) [Now any given interface can only directly extend one other interface.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=79.44) [And an important thing happens when one interface extends another.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=83.74) [Any class that implements the derived interface is automatically](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=87.14) [considered to also implement the base interface,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=90.81) [which means that that class is now conforming to two contracts,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=94.24) [the contract specified by the derived interface,](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=98.2) [as well as the contract specified by the base interface. Now don't get](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=100.76) [too hung up on this, but it's just something important to keep in mind](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=104.79) [that one interface can extend another interface. So now to help us get a](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=107.6) [better understanding of declaring our interfaces, in our next clip we'll](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=112.12) [jump back into our CalcEngine project. Well declare and use an interface of our own.](https://app.pluralsight.com/course-player?clipId=44d73450-6737-44fc-8394-614a3f202809&startTime=115.65)

[Adding and Implementing an Interface in CalcEngine](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4)

[Here we are in IntelliJ and what want to do now is see how we can](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=0.64) [leverage interfaces here within our CalcEngine project.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=4.02) [And as you recall,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=7.14) [the way our application currently behaves is we can enter in a command](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=7.83) [followed by two values with each separated by a space,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=11.54) [and then we'll perform the appropriate operation.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=14.49) [But the application is currently implemented in](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=16.94) [terms of a series of concrete types.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=19.01) [We have CalculateBase,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=21.44) [which is a base class that models the idea of setting a leftVal,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=22.71) [a rightVal, and performing an operation.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=26.44) [We then have a series of classes, Adder,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=29.14) [Subtracter, Multiplier,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=31.2) [and Divider that extend CalculateBase and implement the](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=32.67) [details of a particular operation.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=35.65) [Then we have an enumeration math operation that has](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=38.24) [values for each of the operations, add, subtract,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=41.55) [multiply, and divide.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=44.96) [And this solution works great and we have total control over](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=46.64) [all the pieces within our application,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=49.9) [but there are times where we don't want this tight concrete relationship.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=52.44) [Instead, we want a looser,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=55.24) [abstract relationship so that new components can be easily](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=57.19) [plugged in and that's where interfaces come in.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=60.4) [So let's see how we can model the same behavior using an interface.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=63.24) [So now to create a new interface,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=66.25) [we'll go over here to our Project window and I'll](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=68.47) [right‑click on our package name, I'll head up to New,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=70.52) [I'll head over the Java class, I'll choose that.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=73.9) [Now let's head down here and choose Interface is what we want to create,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=77.94) [and let's name our interface MathProcessing.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=80.94) [Then, once we've named our interface, I'll go and hit Enter.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=85.07) [So now that lays out our interface for us,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=88.94) [so now we want to provide members for the interface.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=91.06) [Remember that members can be a method or they can be a field.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=94.04) [Now remember, a field within an interface is really just kind of a constant.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=97.39) [So a good field for our interface might be the separator that we use](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=101.44) [to break apart the value that a user types in.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=105.08) [Remember that the parts of the user's input are separated by a space.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=108.04) [So let's create a field named separator of type string whose value is space.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=111.62) [So now, once we have that,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=116.74) [let's go and add the methods in that we want to be part of this interface.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=118.13) [Well we're going to plug in new components, those components will need](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=120.84) [to identify what keyword they're associated with.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=124.12) [So let's add a method named getKeyWord that returns a string. Then](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=127.44) [once a component identifies the keyword it supports, it needs to have](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=132.12) [the ability to do the actual calculation.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=135.08) [So let's have a method here named doCalculation that returns a double and](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=137.95) [we'll have it accept two parameters, leftVal and rightVal.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=142.54) [So what we have now is an interface that very simply models](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=146.94) [the idea of being able to do calculations.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=150.69) [We've specified what the separator is we expect between the values, we](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=153.64) [can identify the keyword to the component handles, and the component](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=157.4) [has the ability to receive the values, do the calculation, and return the result.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=160.77) [So now what we need to do is provide an implementation of this](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=165.54) [interface and let's do that on our Adder class.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=168.19) [So I'll head over here and select our Adder class,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=171.28) [and our Adder class already does all the work that we want to do in terms](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=174.34) [of the CalculateBase base class so we just need to enhance it to do the](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=177.81) [work in a way that conforms with the interface.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=182.35) [So let's have it implement our MathProcessing interface. Now,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=185.14) [remember that part of implementing the interface means we have to](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=189.27) [implement the methods that are part of the interface and you'll notice](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=191.63) [that IntelliJ is showing us that red light bulb,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=194.7) [which means there is some help they can offer for us.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=197.14) [So to access that help, here on Windows, I'll hit Alt+Enter. You notice](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=199.44) [the first option is Implement methods so I'll choose that. IntelliJ then](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=204.29) [confirms which methods I want to implement and I want both of them, so](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=209.32) [I'll again hit Enter, and you can see now that IntelliJ has stubbed out](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=213.07) [both of those methods.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=217.55) [Now let me just scroll down a bit so we have some room to work.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=219.04) [So first, we'll need to specify what keyword this component supports and we](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=222.39) [know the keyword that are our Adder class likes is add,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=226.99) [so we'll return back the string add.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=229.71) [Now once we implement that, let's head down here to doCalculation.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=233.2) [And here inside of doCalculation,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=236.64) [we need to perform the steps that are necessary for adder to do the work. Well](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=238.54) [we know that the Adder class needs to set both the left and right values. So](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=242.34) [let's call setLeftVal passing in the value of our leftVal parameter, then we'll](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=246.48) [do the same sort of thing for rightVal.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=250.42) [So now once we set the leftVal and the rightVal to do the](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=253.74) [actual work, we call our calculate method.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=256.46) [Now, as you recall,](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=260.34) [calling a calculate method stores the result in the result](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=261.29) [field so we can then return getResult.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=264.42) [So now that's all there is to it. Our Adder class is doing the same](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=268.94) [work it's already been doing, but now it's doing it in a way that](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=272.18) [conforms to our MathProcessing interface.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=275.53) [So now in our next clip, let's see how we can use this interface based capability.](https://app.pluralsight.com/course-player?clipId=bfd67dec-3c7d-4159-991c-4316b71419b4&startTime=278.54)

[Using the Interface in CalcEngine](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07)

[Here we are now back in our CalcEngine project,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=0.64) [and what we want to do now is continue the work of](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=2.55) [using our MathProcessing interface.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=4.87) [Now to make it a bit easier for us to work with the interface,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=7.14) [I've added another class to the project.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=9.77) [This is the DynamicHelper class,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=12) [and this class takes care of the details of performing work](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=14.14) [based on our MathProcessing interface.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=16.68) [Now you'll notice that DynamicHelper has one field named handlers,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=19.34) [which is an array of MathProcessing,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=22.67) [which means it can hold an array of implementations of that interface.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=24.52) [We have a constructor here that accepts the array,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=28.44) [and then sets the field with whatever array is passed in.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=30.79) [Once we have that array, we'll do the work with a method named Process.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=33.94) [Now we'll look at the details of this method in just a moment.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=38.34) [The key thing to understand at this point is that the](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=41.11) [process method has one parameter, which is a string,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=43.19) [which is the statement that we want to process.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=46.46) [So to see how to use our DynamicHelper class, let's head over to our Main class.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=49.24) [Now over here in our Main class I've added a method,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=53.3) [dynamicInteractivity, and you'll notice at this point,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=56.36) [the only thing this method does is prompt the user for input,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=59.42) [read the user's input, and store it into a local variable named userInput.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=62.84) [So in order the process that input, we're going to need our DynamicHelper class.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=67.04) [So let's declare a local variable named helper of type DynamicHelper.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=70.94) [We'll assign it a new instance of DynamicHelper.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=74.46) [Now remember that DynamicHelper constructor accepts an array of MathProcessing,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=79.24) [so let's create a new MathProcessing array.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=82.59) [Now because the type of this array is MathProcessing,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=85.21) [it can be initialized with instances of any class that implement that interface,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=89.32) [so that means we can initialize it with a new instance of our Adder class.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=93.73) [So now by doing that,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=99.14) [we've initialized the DynamicHelper class to use our Adder](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=100.21) [class as a way to process the user's input.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=103.28) [So with that all in place, after we get the user's input,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=107.04) [we can call helper.process, passing in userInput.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=110.17) [So now whatever the user types in, something like add 10 5,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=115.74) [that string is then passed in to our process method.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=121.14) [So to see how that process method works,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=124.54) [let's head back to our DynamicHelper class.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=125.91) [So now we're here in our process method and we've received the userInput.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=129.54) [Now notice the first thing that we do is we split it into its parts,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=132.99) [and we're using our MathProcessing interface's SEPARATOR field.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=136.94) [Remember we set that to a space,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=140.94) [so it's going to split the input based on the spaces.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=142.55) [So then we should have our parts array with three parts in it,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=145.64) [parts[0] will be the keyword for the operation,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=148.96) [parts[1] and parts[2] will be the left and right values.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=152.26) [So now the next thing we need to do is identify what](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=156.74) [handler understands that keyword.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=159.16) [So we've declared a local variable, the handler of type MathProcessing,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=161.64) [and then we have a loop that will walk through the handlers.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=165.58) [For each handler, it takes the keyword,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=167.95) [does an equalsIgnoreCase asking that handler what keyword it understands.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=170.58) [Now remember, at this point the only keyword we understand is add,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=175.39) [but when we find that match,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=178.74) [we then set the handler to be that handler and break out of the loop.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=180.88) [So once we leave that loop, we have the handler who understands the keyword,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=185.18) [so now the work is really simple.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=189.42) [We simply call doCalculation on that handler,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=191.74) [passing in left and rightVal,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=194.87) [and assign the result to our local variable named result,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=197.14) [and then print out that result.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=199.75) [So you can see what we've focused on here is having the](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=201.94) [classes implement the details of doing the work.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=204.41) [The DynamicHelper class takes care of the details of](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=206.93) [splitting apart the statements, finding the right handler,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=209.78) [and just asking it to do the work.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=212.53) [So let's run our program and let's verify that our add capability still works.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=215.04) [So now once the program starts,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=220.24) [down here in the run window you can see the prompt to](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=221.56) [enter an operation and two numbers.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=223.63) [So I'll enter add 10 5, I'll hit Enter,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=225.64) [and you can see our Adder did its work correctly,](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=231.24) [but it did it in a way that conforms to our MathProcessing interface.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=233.55) [But here's the really cool thing about interfaces.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=237.84) [Once we have all this set up, plugging in new capabilities gets really easy. Let's see how we can do that in our next clip.](https://app.pluralsight.com/course-player?clipId=45cc22d6-96bb-4c18-a985-a488732ccc07&startTime=239.73)

[A New Implementation of the Interface](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab)

[Here we are back in IntelliJ. And what we want to do now is see](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=0.64) [how we can leverage our math processing interface to plug in a](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=3.78) [new feature into our application.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=6.66) [So let's say we want a new feature that supports raising numbers to a power.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=9.04) [Now none of our existing classes do that, so we'll](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=13.14) [create a new class to add that feature.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=15.89) [So to do that, we'll head to the project window, then](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=18.14) [right‑click on our package name.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=20.18) [We'll head up to New, then we'll go over to Java Class, and choose Java Class.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=21.94) [Let's name this class PowerOf. Then to create the class, we'll hit Enter.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=27.14) [So now once we create this class, in order to plug it into the existing system,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=33.14) [we'll need to implement our math processing interface.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=37.28) [And by implementing this interface, it allows us to leverage all the](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=41.34) [code we already have in our DynamicHelper class.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=43.71) [Now remember, as part of implementing the interface,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=46.74) [we have to implement all the methods. So to do that, I'll hit](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=48.63) [Alt+Enter, then I'll choose Implement methods.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=52.51) [Then I confirm that I want both methods, so I'll hit enter.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=55.29) [So now our PowerOf class just has to do the work that's](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=58.54) [necessary to conform to the interface.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=61.18) [So the first thing we'll do is go to getKeyWord.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=64.08) [We'll need to return a string that identifies what keyword we understand.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=66.32) [So our keyword will be power.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=70.24) [So that indicates that this class can do the work](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=73.34) [associated with the keyword power, but now,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=75.62) [of course, we need to do the work itself.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=78.16) [Well, it turns out that we can leverage a class provided by Java to do that work.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=80.44) [There's a class in Java called Math that has a method named pow.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=84.17) [So down in our doCalculation method, let's return back](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=88.14) [Math.pow. And pow accepts two parameters, the value and the](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=91.14) [power you want to raise it to,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=97.2) [so we'll pass in leftVal and rightVal. And that's all there is to it.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=98.81) [We've now built a class that conforms to our mass processing](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=104.24) [interface. Our getKeyWord method identifies the keyword that we're](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=107.31) [responsible for handling. Our doCalculation method does the work](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=110.86) [of receiving the two values,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=114.61) [doing the appropriate work, and returning the value back.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=116.14) [So now in order to use these capabilities,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=119.34) [we need to pass in an instance of our PowerOf class to our dynamic](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=121.14) [helper class. So we'll do that over in our Main class.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=124.71) [So we'll head back there.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=127.77) [So now here in our Main class,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=130.04) [we have our dynamicInteractivity method, and up here at the top is](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=131.31) [where we create the instance of DynamicHelper.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=134.87) [And as part of creating that instance, we pass in this](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=137.37) [array of MathProcessing implementations.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=139.4) [So here after Adder, let's go ahead and create an](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=142.64) [instance of new PowerOf, and we're done.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=144.57) [And that's the power of interfaces.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=149.15) [Because all of our code inside of DynamicHelper works only in](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=151.5) [terms of the interface, MathProcessing,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=155.47) [it doesn't know anything about the classes, doesn't know anything](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=157.85) [about Adder, doesn't know anything about PowerOf.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=160.3) [So that makes it very easy for us to plug in new features by](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=162.84) [simply implementing the interface that DynamicHelper understands.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=166.6) [Now just to confirm that our PowerOf class works correctly, we're](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=170.04) [just going to run our code.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=173.9) [So now down here in our Run window, I'll enter power 5 2. So that should raise](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=176.04) [the number 5 to the power of 2. And if that works correctly,](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=182.65) [we should get back 25. So as you can see, interfaces make it very easy](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=186.18) [for us to plug in new capabilities into existing code, but there is a](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=191.12) [challenge with interfaces we need to consider.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=195.81) [Sometimes after you've already released an interface, you](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=197.88) [need to add new features to the interface. So in our next clip, let's see the appropriate way to deal with that scenario.](https://app.pluralsight.com/course-player?clipId=61a15030-fc8c-4e89-a3a4-f7a541b47bab&startTime=200.61)

[Adding a Default Method to an Interface](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b)

[Now one of the challenges that often comes up with](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=0.64) [interfaces is that once you've declared them,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=2.47) [as time goes by, you may want to add some new features to the interface.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=4.88) [Now, for example, in our case,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=9.78) [the way we currently display our output is pretty simple.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=11.51) [We simply say something like results = 25.0.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=14.15) [And maybe we want to provide the capability for classes to return](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=17.08) [back a more descriptive version of their results.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=20.82) [So let's say want to add a method here, getFormattedOutput,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=24.53) [that returns back a string.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=26.56) [Now that seems simple enough.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=30.34) [But the problem is that there are already classes that implement this interface.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=32.74) [If we start adding new methods to the interface,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=36.88) [we need to go back and revisit all of those classes.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=39.15) [Now, in our case, we're only talking about two classes.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=41.94) [But in practice, there could be tens, hundreds,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=44.5) [or even thousands of classes that have already implemented your interface.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=46.76) [And so adding this method would break all of that code.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=50.3) [So to help us with that,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=53.94) [Java supports something known as a default method on an interface.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=55.21) [Now with the specified default method,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=58.81) [we put the default keyword before the method return value.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=60.6) [And because it's the default method,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=65.34) [we can actually provide a default implementation.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=66.8) [So, like in our case, what we might do here is simply return back a null.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=69.54) [So now by doing that,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=75.14) [any class that implements the interface MathProcessing that doesn't](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=76.21) [explicitly implement getFormattedOutput will use this default](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=80.03) [implementation that simply returns back a null.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=83.88) [But the thing to keep in mind is that no class is](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=86.64) [required to use the default implementation.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=88.58) [So let's say, for example, our PowerOf class wants to support getFormattedOutput.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=91.54) [So let's head over to our PowerOf class.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=96.24) [So now here in our PowerOf class,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=99.44) [let's add a field named formattedOutput of type String,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=101.04) [and then what we'll do is go down here to doCalculation,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=104.92) [and as part of doing the calculation, let's set up our formattedOutput.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=108.2) [So here we returned the value of Math.pow.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=113.04) [Instead, let's assign it to a local variable named product.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=115.44) [So now once we have the product,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=119.84) [let's go ahead an assign a nice formatted version of our](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=121.39) [output to our field formattedOutput.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=123.82) [And then, finally,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=127.54) [we still need to make sure we fulfill the requirements of doCalculation.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=128.52) [So the last thing we'll do is return back the product.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=131.87) [So now at this point,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=135.94) [doCalculation does the work of preparing the formattedOutput and returning](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=136.73) [back the product as the return value of doCalculation.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=140.54) [But we still haven't implemented the new method.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=143.76) [Well,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=146.29) [to do that, here on Windows, I compress Ctrl+O. You notice](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=146.42) [that one of the options I have now is that method that](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=150.92) [we've added getFormattedOutput.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=152.94) [So I'll choose that. Then here in getFormattedOutput,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=155.34) [I'll return back formattedOutput.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=158.57) [So now with that,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=162.54) [the PowerOf class supports all three methods in the](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=163.26) [MathProcessing interface. We have getKeyword,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=165.89) [getFormattedOutput, and doCalculation.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=168.61) [But if I head over to our Adder class,](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=171.44) [it still only supports the original two, getKeyword and](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=174.57) [doCalculation. And that's okay because I have default](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=177.68) [implementation for the getFormattedOutput method.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=181.43) [That's a key thing to understand.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=185.04) [As you evolve your interfaces, you want to be sure not to](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=186.5) [break existing code, and default methods give us a mechanism for dealing with that.](https://app.pluralsight.com/course-player?clipId=beddf275-07a2-49f4-988c-c81aa9e62e4b&startTime=189.03)

[Summary](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5)

[To wrap up, here's some of the key things we want to remember from this module.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=0.74) [Remember that an interface defines a contract,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=4.54) [so it actually defines a contract for behavior, and as part of doing](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=7.64) [that, an interface provides a list of operations.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=11.54) [Remember that these operations are not focused on](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=14.54) [implementation details, it's simply a list of methods and the](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=16.68) [behavior that's expected from those methods.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=20.04) [So then it's up to classes to actually implement our](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=23.14) [interfaces. By implementing an interface, a class](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=25.49) [expresses that it conforms to a contract,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=28.22) [and a key part of conforming to that contract is the class is going to have](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=31.24) [to provide all the necessary methods for the interface.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=34.31) [Remember that many interfaces are actually generic interfaces,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=38.44) [which means it's an interface that allows stronger typing.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=41.86) [So as part of specifying that we implement the interface, we can](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=45.34) [actually specialize the interface on a particular type,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=47.87) [which allows the methods corresponding to the](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=50.55) [interface to be more strongly typed.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=52.46) [And remember that a key aspect of interfaces is that one](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=55.18) [class can implement multiple interfaces,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=58.28) [so following the implements keyword we can simply provide the list of](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=61.54) [interfaces that we support separated by a comma.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=64.56) [And remember that there's no practical limit on how](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=68.21) [many interfaces a class can implement.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=70.66) [Any class can implement as many interfaces as is](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=73.07) [needed for that particular class,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=76.04) [which allows one class to conform to many different contracts for behavior.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=78.29) [And remember, as developers we're not just limited to implementing](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=83.54) [interfaces, we can actually declare our own interfaces.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=86.89) [Remember that declaring an interface is very similar to declaring a class](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=90.64) [except we use the interface keyword rather than the class keyword, and](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=94.02) [remember that any members in an interface or implicitly public. Now](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=97.82) [remember that our interfaces can have fields, but fields in an interface](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=102.56) [are not exactly like fields in a class.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=107.05) [Remember that fields in an interface really are just named values.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=109.52) [They're a way to have named constants as part of the](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=113.22) [interface. In general, when we're declaring interfaces we](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=115.7) [focus much more on the methods,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=118.96) [but those methods will generally just be the name,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=120.96) [the list of parameters, and the return type.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=123.41) [The methods are generally not going to have a body, because the](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=126) [interface is not focused on implementation.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=128.29) [We expect the classes to implement the methods. But](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=130.81) [remember, an interface can have default methods.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=134.22) [In other words, there are methods in an interface that can have a](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=136.59) [body, and the reason we do that is because it allows the interface](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=139.44) [to be extended over time without breaking existing classes that](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=143.52) [implement the interface.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=147.18) [So by using a default method,](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=148.84) [we can specify the behavior for the method for any class that](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=150.54) [doesn't explicitly implement that particular method.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=153.91) [Alright, that wraps up this module. In the next module, we'll take a look at nested types and anonymous classes.](https://app.pluralsight.com/course-player?clipId=af0b4950-f970-4b33-9bb0-63cc398c38e5&startTime=157.24)

[Nested Types and Anonymous Classes](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9)

[Introduction](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9)

[Welcome to our next module, Nested Types and Anonymous Classes.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=0.74) [This is part of the Pluralsight course Working with](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=4.51) [Classes and Interfaces in Java, and my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=6.42) [Most of the times, when we declare classes and interfaces,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=11.04) [we declare them as top‑level types,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=13.79) [meaning that each type is declared within its own file,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=16.33) [but there are times where we want to have a closer relationship.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=19.02) [There are times where we we want to actually nest one type inside of another.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=22.54) [So we'll start out looking how we can declare one type inside of another,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=26.24) [and then we'll take a look at why we'd want to do that,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=30.44) [and there are actually a couple different reasons.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=32.78) [In some cases,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=34.9) [you want to have one type that's nested inside of another type just](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=35.93) [to provide this new type with some naming scope.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=39.37) [In other cases,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=42.24) [we actually want to create a very tight relationship between these classes,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=43.31) [and that uses a concept known as inner classes.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=46.8) [And there are even some cases where we want to use was a](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=50.34) [very special kind of nested class,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=52.26) [known as an anonymous class where we actually create a class,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=54.13) [but never give that class of name.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=57.32) [Now, as we mentioned, throughout this module,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=60.54) [we're going to look at nested types,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=62.45) [and a nested type is simply a type that's declared within the](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=64.44) [opening and closing brackets of another type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=68.23) [And so the type that's inside is known as the nested type. The](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=71.24) [type that contains the nested type is known as the enclosing](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=74.13) [type. And when you declare a nested type,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=76.98) [it's actually considered a member of the enclosing type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=79.49) [And that's really important because as a member of the enclosing type,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=83.04) [it means the nested type has access to the private](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=86.93) [members of its enclosing type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=90.06) [And because nested types are actually considered part of their enclosing types,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=92.47) [nested types can actually utilize all of the access modifiers.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=96.34) [Remember, the top‑level types can only be package private or public.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=100.74) [Nested types can use all of the access modifiers.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=104.74) [So a nested type can be package private, meaning they could only be](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=108.14) [used within the package where it's declared. A nested type can be](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=111.48) [public, and it can could be used anywhere,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=114.48) [but also, a nested type can be private, meaning that's only usable by its](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=117.04) [enclosing type. And nested types can be marked as protected,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=121.82) [which means that that type is only usable by other types](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=126.09) [that inherit from its enclosing type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=129.17) [Now, there are actually two broad categories of nested types.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=132.74) [In one case, we use nested types just to provide naming scope.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=136.24) [In other words,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=140.5) [the nested type's name has to somehow be qualified by the enclosing type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=140.96) [But in other cases, the relationship is much more close.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=145.44) [In these scenarios,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=148.69) [each instance of the nested type is linked to an instance of the enclosing type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=150) [Now, the thing is, as we declare these different kinds of nested types,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=154.84) [they're similar, but different.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=158.14) [The similarity is the syntax.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=160.34) [There's not a big syntactical difference between a nested type](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=162.52) [that's used for naming scope and one that actually links the](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=165.67) [instances of the enclosing type.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=168.63) [But their behavior is incredibly different,](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=170.64) [so it's important we understand how to work with both types.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=172.82) [So in our next clip, let's take a look at nested types that are used for naming scope.](https://app.pluralsight.com/course-player?clipId=833151a8-f256-480f-9523-aec9782f33d9&startTime=176.74)

[Nesting Types for Naming Scope](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1)

[One of the common reasons we use nested types is to provide naming scope.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=0.69) [In other words,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=5.04) [we want to declare one type and have the name of this](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=5.5) [type scoped within another type.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=7.83) [There's a number of places where this gets used.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=10.68) [If you've seen any of my Android courses here on Pluralsight, you know](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=12.7) [the Android platform uses this idea quite a bit.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=15.5) [It just makes it easy to declare types that are qualified](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=18.58) [by other types that they relate to. Now something that's](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=21.77) [very important to understand,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=24.41) [in this scenario there's no relationship between instances of the](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=25.69) [nested type and instances of the enclosing type.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=29.74) [We're simply scoping the name of the nested type](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=33.24) [within the name of the enclosing type.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=35.52) [There's a few ways we can declare types using this technique.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=37.94) [One way is to declare a class inside of another class and](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=40.78) [mark that nested class as static.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=44.43) [But interestingly,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=47.24) [you can declare classes not just in other classes, you can](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=48.22) [also declare classes within interfaces. So any time we](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=51.24) [declare a class within an interface,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=55.08) [we're just doing that, again, to provide naming scope.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=56.83) [Also,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=60.24) [you can declare an interface within another interface or an](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=60.93) [interface within a class. In both of these cases,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=64.08) [nesting the interface just qualifies the name of the](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=67.28) [interface within its enclosing type.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=69.84) [Now the best way to understand how this kind of nesting works is to](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=73.44) [take a look at an example, and so to do that,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=76.24) [let's take a look here at our Passenger class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=78.62) [As you recall, at this point our Passenger class](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=81.24) [has a number of fields within it.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=83.51) [The fields we want to look at right now are memberLevel and memberDays.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=85.54) [Remember,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=89.37) [both of these relate to information for the reward program for passengers.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=89.64) [Now remember, also,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=94.19) [our Passenger class has a number of constructors. One of those](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=94.95) [constructors accepts the passenger's name,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=98.09) [as well as their memberLevel and memberDays, and](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=100.13) [then sets the appropriate fields.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=102.64) [Now here in our Passenger class,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=105.24) [these two fields, memberLevel and memberDays, are actually very closely related.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=106.55) [They're not really useful by themselves.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=111.34) [They're part of a larger idea of a reward program tied to a passenger.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=113.35) [So this might be a really good place where we could use a nested class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=118.34) [So let's put memberLevel and memberDays into a nested class named RewardProgram.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=121.91) [So now those two fields are no longer part of the Passenger](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=126.94) [class, they're part of this class RewardProgram.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=129.35) [So let's go ahead and indent those fields just make](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=132.24) [our code a bit more readable.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=133.88) [Now notice that both those fields are private, so since they're private,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=135.54) [we also probably want to put getters and setters to be](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=138.33) [able to interact with those fields.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=140.89) [So now we have RewardProgram, which is our nested type,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=143.04) [and it's nested within Passenger, and Passenger is considered the enclosing](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=146.13) [type. And there's a really important thing we want to notice about our](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=150.03) [RewardProgram class. Notice that it's declared using the static keyword, and](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=153.1) [by using the static keyword that indicates that RewardProgram is being nested](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=158.14) [just to provide scope for its name.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=162.5) [The name RewardProgram is scoped within the Passenger class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=164.63) [So now here on screen we actually have two classes, the](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=168.74) [Passenger class and the RewardProgram class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=172.38) [And, again, our fields,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=175.09) [memberLevel and memberDays, are now part of RewardProgram. So what that](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=176.04) [means is down here in the constructor where we set memberLevel and](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=180.54) [memberDays, that code will no longer compile, because memberLevel and](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=183.49) [memberDays is not part of Passenger. And remember that RewardProgram,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=187.19) [just like with any other class, declaring the class doesn't create an](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=191.37) [instance of anything. In order to be able to interact with](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=194.86) [RewardProgram, we're going to need a field.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=197.59) [So here in our Passenger class we'll have a field named](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=199.79) [RewardProgram whose type is rewardProgram, and we'll assign](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=201.93) [that field a new instance of rewardProgram.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=206.04) [So now our field rewardProgram refers to an instance of our RewardProgram class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=209.34) [So that means that down here in the constructor where we set](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=214.94) [memberLevel and memberDays, we want to change this code to say](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=217.4) [rewardProgram.memberLevel, rewardProgram.memberDays.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=220.76) [So now we have a RewardProgram class that's nested within our](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=226.64) [Passenger class, and our Passenger class is updated to use](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=229.38) [that nested RewardProgram class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=232.95) [So now in our next clip,](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=235.84) [let's see what it's like to interact with our nested class from outside the Passenger class.](https://app.pluralsight.com/course-player?clipId=2111a063-0a96-4613-9279-00ee8efb52c1&startTime=237.04)

[Accessing a Nested Type](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47)

[So let's see what it's like to work with our Passenger](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=0.74) [class and our RewardProgram class.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=2.34) [Now, as you might expect,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=5.24) [creating a new instance of the Passenger class is not really going to change.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=6.26) [So here we have a reference named Steve, who's a passenger,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=9.74) [and we've assigned it a new instance of Passenger, the name set to Steve, the](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=12.6) [member level set to 3, and the number of days set to 180.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=16.24) [But remember the reward program, now rather than just being a couple of fields,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=20.14) [is actually a type,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=24.03) [so that means I could declare a variable of type RewardProgram.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=25.84) [But RewardProgram is not a standalone type. It's](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=29.34) [actually nested within the Passenger class.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=32.05) [So the name of that type is now Passenger.RewardProgram.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=34.94) [Because RewardProgram is nested inside of Passenger, anytime we](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=39.54) [interact with it outside of the Passenger class,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=42.96) [we have to qualify it by the Passenger class.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=45.15) [So to create an instance of this class, we'll simply](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=48.54) [say new Passenger.RewardProgram.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=50.45) [So our platinum variable refers to a new instance of Passenger.RewardProgram,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=53.77) [and from here we can interact with it just like any other type.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=58.64) [So for our platinum reward program, we'll set the member level to 3.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=62.34) [So let's go ahead and add some code now that can check to](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=66.54) [see if Steve is a platinum level member.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=68.41) [And remember that our RewardProgram field within our](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=71.09) [Passenger class was actually private.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=73.22) [So in order to get to that reference,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=75.44) [we had to have a getter there, so we'll say steve.getRewardProgram.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=76.98) [That gives us the overall RewardProgram, and then the MemberLevel is](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=80.13) [within that instance of RewardProgram.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=84.78) [So we'll get the MemberLevel back from Steve's RewardProgram,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=87.24) [and then from there we can just check to see if it's](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=90.7) [equal to the MemberLevel within platinum.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=92.29) [And it turns out that comparison is true,](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=95.74) [so we could print out the message that Steve is platinum.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=97.95) [So as you can see, the way we've declared](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=101.44) [RewardProgram, it's a class just like any other.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=103.31) [We simply declared the class in a way so that the class name](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=106.84) [is scoped within the Passenger class. Now, in our next clip, let's take a look at something known as inner classes.](https://app.pluralsight.com/course-player?clipId=147e4733-6542-492f-bfd3-cea0e89f8d47&startTime=109.97)

[Inner Classes](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3)

[Let's take a look now at another type of nested class known as an inner class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=0.7) [Now, when we create an inner class,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=5.24) [the class' name is still scoped by the enclosing class,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=6.52) [but the key thing about inner classes is it actually creates an instance](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=11.04) [relationship between the inner class and its enclosing class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=14.2) [Basically, each time we create an instance of the inner class,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=18.64) [that instance is associated with the instance of the enclosing class that](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=22.31) [was used to create the instance of the inner class,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=26.12) [so you're linking instances of these two classes together.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=29.34) [Now, there's only one way to create an inner class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=33.44) [And to do that,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=36.14) [you have to declare one class inside of another class and](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=36.94) [not mark the nested class as static.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=40.97) [So non‑static nested classes are inner classes.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=43.71) [So to see how this works, let's take a look at our Flight class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=48.74) [Now, you remember, our Flight class has a number of members.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=52.31) [One of those members is our passengerList,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=54.66) [and that's an ArrayList that contains all of our passengers.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=57.29) [Remember, on our Flight class, we implemented this Iterable interface.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=60.07) [And as part of that interface, we had to provide a method named iterator,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=63.27) [and we return back the Iterator from our passengerList,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=67.71) [so that allowed us to write code that could walk through the list of passengers](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=71.74) [without having direct access to the passengerList itself.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=75.54) [So now, the way our Flight class implements the Iterable interface,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=79.44) [we walk through the passengers in whatever order to have](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=83.19) [them be added to the passengerList, and most of the time,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=86.12) [that's probably fine.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=89.19) [But remember,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=91.04) [there are times where we want to get back the list of](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=91.49) [passengers in order by member level, highest level members first,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=93.72) [lowest level members last.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=98.08) [So it might be helpful to be able to return back a different](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=100.14) [implementation of Iterable that lets us walk through the list](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=103.14) [of passengers in sorted order.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=106.01) [This is a great chance for us to use an inner class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=108.5) [So I'm going to declare a class here named FlightIterable.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=111.74) [And you notice, the way this class is declared,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=115.24) [its private,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=117.1) [meaning that the class itself can only be used](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=118.06) [directly inside of our Flight class, and notice that it is not marked as static,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=120.16) [which means that each time we create an instance of FlightIterable,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=124.84) [it'll be associated with the instance of the Flight](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=128.27) [class that was used to create it.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=130.57) [Now, our FlightIterable class will go ahead and implement the Iterable interface.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=133.19) [And again, as part of that interface, it will implement the iterator method.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=137.94) [So what we want to do in this implementation is put together a sorted list of](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=142.14) [passengers that could be walked through with a for loop.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=145.66) [So to do that, let's go and declare a Passenger array.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=148.74) [And when we create the instance of the Passenger array,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=152.44) [we want to set it to be the correct size,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=154.41) [so we're going to use passengerList.size. PassengerList is not](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=156.09) [a field within the FlightIterable class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=161.69) [It's actually a field within our Flight class, but that's okay.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=164.42) [Since FlightIterable is an inner class,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=168.46) [it can actually reference the members of its enclosing class, Flight.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=171.25) [The reason that it can do that is that an inner class](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=175.64) [actually has multiple this references.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=178.65) [It has its standard this reference, which will let it access its own members,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=181.74) [but it also has a this reference for its enclosing type.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=186.24) [So FlightIterable actually has a flight.this reference,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=189.74) [which means that here in FlightIterable, we can](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=194.39) [access those fields within Flight,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=196.78) [so that allows us to size our Passengers array based](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=199.64) [on the passengerList within Flight.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=203.03) [Now, once we have the array sized,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=205.54) [we want to go ahead and put the passengers into the array.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=207.02) [So the ArrayList class actually has a toArray method that](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=209.37) [will copy the references into an array.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=212.32) [So once we do that,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=214.94) [remember that our Passenger class actually implements the comparable interface,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=216.11) [so we know that we can actually sort that Passengers array.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=219.89) [Remember,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=223.84) [our sort puts the highest level members first and the lowest level members last.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=224.08) [So once we do that,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=228.44) [we can simply create another ArrayList containing the newly](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=229.61) [sorted passengers and return back that Iterator.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=233.2) [So now, once we do this work,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=237.14) [we actually have two separate Iterable implementations that can](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=239.01) [walk through the passengers. On our Flight class, we have our](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=243.22) [initial implementation of Iterable,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=246.92) [and that simply returns back the passengers in whatever](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=249.01) [order they were added to the passengerList.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=251.59) [But in our FlightIterable class, we have a separate implementation,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=254.54) [and this implementation does the work to sort those passengers and return](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=258.34) [them back so they can be walked through in sorted order.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=262.42) [Now, remember, FlightIterable is just a class declaration.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=265.88) [We need to actually create an instance of this class to use it.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=269.54) [So here in our Flight class, let's add a method named getOrderedPassengers.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=272.48) [So now, here inside of this method,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=277.54) [we'll declare a variable of type FlightIterable named orderedPassengers,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=279.07) [and then we'll create the new instance of FlightIterable.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=283.24) [Remember, since FlightIterable is an inner class, creating](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=286.74) [this instance associates this instance with the current](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=290.07) [instance of the Flight class,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=293.81) [which allows it to interact with the fields of this current instance.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=295.22) [So when we return back orderedPassengers,](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=299.34) [we're returning back an Iterable implementation that has](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=302.24) [done the work to sort the passengers. So now, in our next clip, let's see what it's like to use our inner class.](https://app.pluralsight.com/course-player?clipId=ec2ecb3c-1040-45f4-a20c-e2fcb79ec9a3&startTime=304.92)

[Anonymous Classes](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5)

[So let's see what it's like to work with our Flight class now.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=0.81) [So we'll create a new instance of our Flight class, giving a flight](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=3.02) [number of 175, and assign it to a reference here, f175. And then we'll](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=5.46) [add four passengers to the flight, Luisa,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=10.59) [Jack,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=13.36) [Ashanti, and Harish. And remember that since our Flight class actually](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=13.92) [implements the iterable interface, we can use it with our for loop.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=17.78) [So we'll say for Passenger p, and we'll walk through f175. And remember](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=21.74) [that this code will get back a reference to the iterable interface, as](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=27.04) [we've implemented it directly on the Flight class.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=30.42) [So we're going to walk through the list of passengers](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=33.28) [in the same order they were added.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=35.34) [So if we walk through printing out their names, we get back Luisa,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=37.34) [the first one added, then Jack,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=40.6) [then Ashanti, and then finally, Harish, which was the last one added.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=42.97) [But remember, we also have our inner class.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=47.44) [So let's change our for loop just a little bit.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=50.74) [Let's make our for loop for Passenger p,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=53.19) [and instead of walking directly through f175, we'll](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=56.1) [call its getOrderedPassengers method.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=59.61) [Remember that when we call that method internally, it will create an instance](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=62.54) [of our FlightIterable class. Because it was an inner class, it could access](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=66.35) [the passenger list member of our Flight class,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=70.97) [and this implementation returns back an iterable with the passengers sorted.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=73.73) [So then, in this case,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=77.42) [the first passenger we would print out would be Ashanti,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=78.56) [because remember earlier in the course, we had our Passenger](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=81.13) [class implement the comparable interface,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=84.08) [and it orders passengers based on their member level.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=86.84) [Ashanti is a level three member,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=89.95) [so she comes back as the first one in our sorted list.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=92.28) [Then we get Harish, who's a level two member, then after that,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=96.24) [we'll get back Luisa, because both Luisa and Jack are level one](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=99.94) [members, but Luisa's been a member longer,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=103.07) [and then finally, at the end, we would get Jack.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=105.74) [So,](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=108.64) [as you can see by using these inner classes, we can nest one class inside](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=108.8) [of another with that nested class doing work that uses the members of the](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=112.69) [enclosing class. Okay, so now in our next clip, let's take a look at something known as anonymous classes.](https://app.pluralsight.com/course-player?clipId=d6d3348e-97a3-47cc-809d-58def877cdd5&startTime=117.17)

[Using Anonymous Classes](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2)

[So let's take a look now at Java's anonymous classes, and an anonymous](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=0.74) [class is a class that doesn't actually have a name.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=5.49) [Now, as we normally work with classes,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=8.84) [we first go off and declare the class, and,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=10.47) [of course,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=12.61) [part of declaring a class we give it a name, and then later on we go ahead and](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=12.92) [create instances of the class by using the class's name. Well, in the case of](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=16.61) [an anonymous class, we do all that in one step.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=20.7) [We actually declare the class as part of creating an instance of the class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=23.54) [And there's a few common scenarios where we use anonymous classes.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=28.44) [Oftentimes we want to implement a simple interface and just use it in one place,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=31.84) [or maybe we want to provide an extension of some simple](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=35.71) [class and just use it in one place.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=38.04) [Well, these are great opportunities to use anonymous classes.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=40.74) [Now an anonymous class is actually a form of an inner class,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=44.24) [and remember, as an inner class,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=48.14) [the anonymous class is associated with the instance of its enclosing class,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=50.1) [which means it can actually access private members of the enclosing class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=55.01) [Now the syntax for creating an anonymous class is actually a little bit odd.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=60.39) [Remember, we actually declare the class as part of creating an instance of it,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=65.08) [so it will always involve the new keyword, and that will be](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=69.28) [followed by either the base class that you want to extend or the](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=71.98) [interface name that you want to implement.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=75.82) [And because we're creating a new instance right over that type name,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=78.34) [we're going to place opening and closing parentheses, just like we would](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=81.3) [when creating an instance of any other type.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=84.92) [And then right there where we're creating the instance,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=87.24) [we can go ahead and add the code.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=89.54) [So after the parentheses,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=91.44) [we'll have our opening and closing brackets, and inside those](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=92.72) [brackets we can implement any methods that we need, and override](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=95.57) [any method from the base class or the interface that we want to](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=99.45) [provide. Now to help us understand anonymous classes, let's again](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=102.47) [look at our Flight class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=107.21) [Now remember,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=108.82) [our Flight class has a field, passengerList, which contains a list of](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=109.16) [passengers, and our Flight class provides an implementation of Iterable](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=112.82) [which allows us to walk through that passengerList in the order the](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=116.76) [passengers were added to the list. And then earlier in the module, we also](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=120.73) [wanted to provide an implementation of Iterable that would let us walk](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=125.38) [through the passengers in sorted order as defined by the Passenger class's](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=128.34) [Comparable implementation.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=132.52) [Well, to do that,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=134.44) [we used an inner class named FlightIterable, but now](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=135.16) [this inner class had exactly one job,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=138.39) [and that was to provide an implementation of the Iterable interface.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=141.34) [So we added the method, Iterator,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=144.84) [which is required for the Iterable interface, and then added the code](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=146.69) [necessary to sort that list of passengers and give us back an iterator](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=149.99) [that would walk through that sorted list.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=153.97) [But now there's a couple of important things to](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=156.54) [understand about our FlightIterable class. And I've](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=157.91) [already mentioned it has exactly one job,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=160.76) [which is to provide this implementation of the Iterable interface,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=163.39) [but also note that the class is marked as private, meaning that](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=167.34) [this class, FlightIterable, is not even accessible outside the](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=171.04) [Flight class, so the only places it can be used is right here](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=174.72) [inside of the Flight class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=178.44) [Let's take a look at what we did with that class inside the Flight class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=180.44) [Well,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=183.94) [we had our method getOrderedPassengers, and inside there we](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=184.12) [created an instance of FlightIterable, and then immediately](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=187.97) [returned back a reference to that instance.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=191.22) [So FlightIterable has one job, provide an implementation of an](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=194.34) [interface, and it is only used in exactly one place,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=198.12) [which is our getOrderedPassengers method.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=201.46) [Well, this is a perfect scenario for using an anonymous class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=204.04) [We don't actually need to explicitly declare a class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=208.04) [We can use an anonymous class and provide all the implementation](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=211.59) [right here where we use the class, so here in getOrderedPassengers.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=215.1) [Since we need a class that extends the Iterable interface, we'll](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=219) [start out with the interface name.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=223.19) [Remember, when we're working with the anonymous class,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=225.6) [we declare it right where we use it, so we're going to create a new instance of](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=227.89) [a type, so we'll have our opening and closing parentheses.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=231.01) [We're going to provide all of our code here, so we'll have our](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=234.09) [opening and closing brackets, and then as an anonymous class right](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=236.13) [here where we're working with the type,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=240.06) [we'll go ahead and new it up, and then in this case,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=241.38) [we'll even just go ahead and return it back.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=243.74) [So now we have one statement that's doing a whole bunch of things.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=246.14) [We're specifying an anonymous class that's going to implement the Iterable](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=250) [interface. As part of specifying that, we're creating the instance with the](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=253.47) [new keyword and returning it back. Now since this is treated as a single](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=256.83) [statement, after that closing bracket,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=261.16) [we need to be sure that we provide a semi‑colon,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=263.95) [because again we're ending a statement.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=266.34) [Then with that all done, as an anonymous class we can start](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=268.77) [providing our method implementations.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=271.96) [So right here where we new it up and return back a](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=274.54) [reference, we're going to provide the code.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=277.27) [So here inside of this method,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=279.74) [we're going to go ahead and create our passengers array,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=281.24) [and we want to size it up based on the passengerList inside of our Flight class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=284.18) [So the fact that we can access that passengerList shows that this](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=289.04) [anonymous class is treated as an inner class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=292.56) [Remember, inner classes can access the private members of their enclosing class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=295.88) [So then once we size the array, we can go ahead and copy the passengers](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=300.74) [from the passengerList into the array, do the sort work, and then return](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=304.22) [back the Iterator. And that's it, we're done.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=308.17) [And that's the power of anonymous classes.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=311.82) [We needed a type that implemented a specific interface,](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=314.64) [and we only needed to use it in one place, so rather than](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=318.23) [separately declare it and then create it right here in the method](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=321.66) [where we need it, we were able to do both of those things at once by using an anonymous class.](https://app.pluralsight.com/course-player?clipId=3c4f2ffb-66f9-4196-86dc-43887d8948c2&startTime=325.14)

[Summary](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4)

[To wrap up, here are some of the key things we want to remember](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=0.74) [from this module. Remember, as we talked about nested types, a](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=3.09) [nested type is a type that's declared inside of another type,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=5.97) [what's known as its enclosing type.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=9.34) [And remember that a nested type is a member of the enclosing type.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=11.39) [This is an important thing to keep in mind because a nested type can](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=15.24) [actually access the private members of the enclosing type.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=18.09) [Now, as we talked about,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=23.21) [there are two broad categories of how we use nested types.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=24.41) [In one scenario, we actually nest a type to provide naming scope.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=28.44) [In other words,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=32.67) [the name of the nested type is actually scoped by its enclosing type.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=33.06) [So when we declare it a RewardProgram class inside of our Passenger class, the](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=37.04) [name of that class was actually Passenger.RewardProgram.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=41.31) [But remember, in this scenario,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=45.54) [there's no relationship between instances of the type. We're using](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=47.25) [nesting just to provide scoping for the type name.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=51.38) [But then remember,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=55.12) [there's another kind of nesting known as inner classes, and in inner classes,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=55.6) [there's actually a very close relationship between the](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=59.53) [nested type and the enclosing type, because in this situation,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=62.3) [each time you create an instance of the inner class,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=66.52) [that instance is associated with the instance of the enclosing class that](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=69.77) [was used to create the inner class. And as we saw, this mechanism makes it](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=74.57) [very easy for us to create utility classes or helper classes that support](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=78.94) [the work of the enclosing class.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=83.89) [And then we finished up with anonymous classes.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=87.44) [Remember that anonymous classes are classes that we actually declare as](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=89.93) [part of the class creation. And these were helpful for scenarios where you](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=93.61) [just needed to use a particular type in one scenario. We use these](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=97.92) [oftentimes when we're implementing a simple interface or providing a simple](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=101.79) [extension to a base class. Because,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=105.49) [remember, as part of declaring a class, right there where we](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=108) [create it, we can implement any methods we need and even override](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=110.5) [methods from the interface or base class.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=114.46) [And then remember that an anonymous class is actually an inner class,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=117.14) [which means that when we create an instance of the](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=121.14) [anonymous class, it's associated with the instance of the](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=123.02) [enclosing class that created it.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=126.43) [So again,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=129.14) [we have that very tight relationship between the anonymous class](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=129.56) [we create and its enclosing class. All right, that wraps up this](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=132.54) [module and it wraps up this course.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=137.44) [Now again,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=139.94) [I want to remind you, that every bit of code you've seen, both in](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=140.34) [the slides and in the demos, is available for download as part of](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=143.64) [the exercise files for this course, so I really encourage you to](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=147.81) [take advantage of that.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=150.84) [As developers,](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=152.14) [we work with code, so download that code, play with all those](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=153.05) [examples, and use them as a tool to build your understanding.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=156.48) [Again, my name is Jim Wilson.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=160) [I hope you've enjoyed viewing this class as much as I](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=161.97) [enjoyed creating it, and I wish you all the best as you continue developing your Java skills.](https://app.pluralsight.com/course-player?clipId=41325476-5c1c-41c9-8b5c-eb5f75eecee4&startTime=164.17)