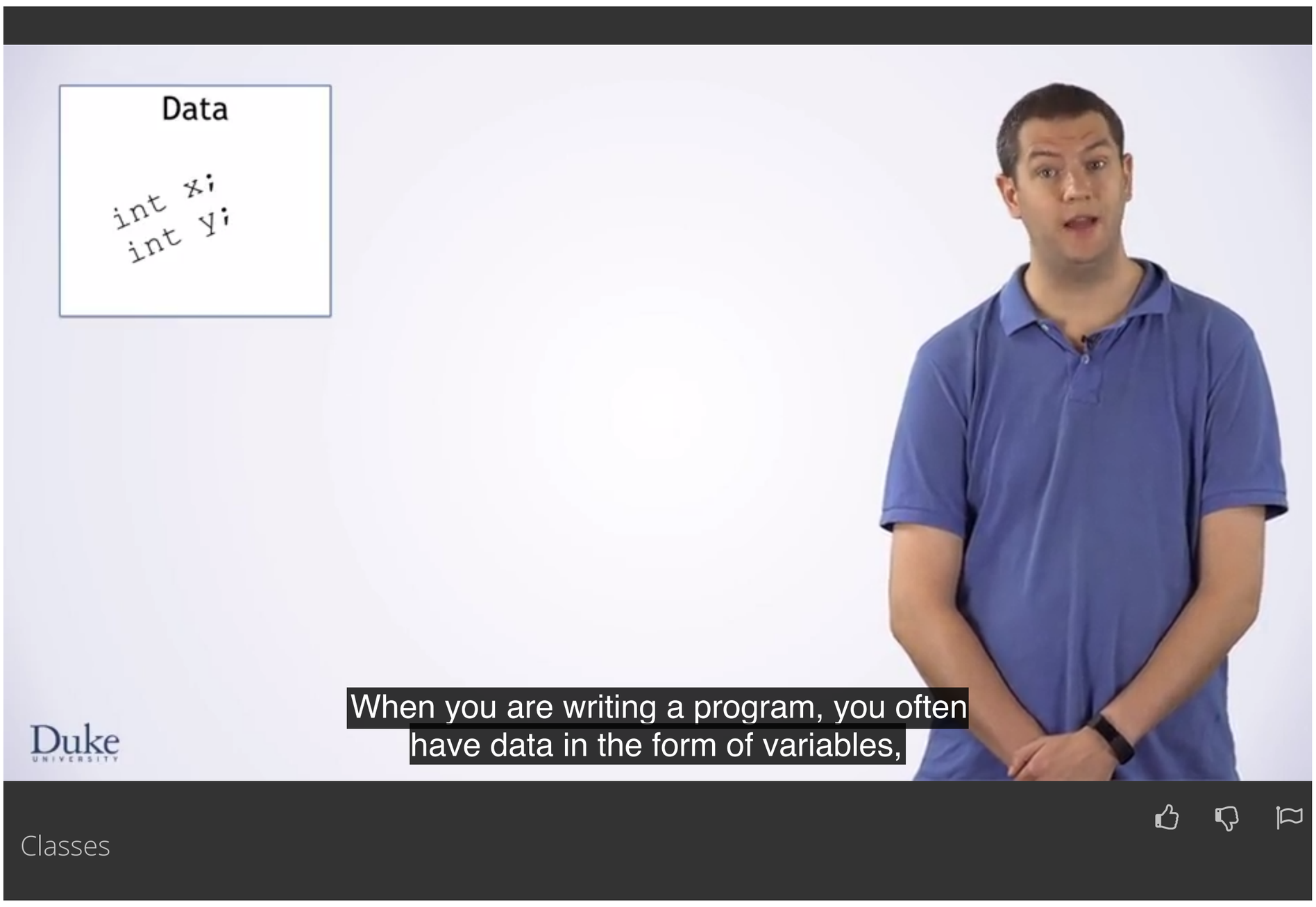
Classes, Types, and For Each Loops



Classes

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript English

0:03
As you continue learning how Java works, it is important to talk for a bit about objects and classes. Before we delve into the specific semantics, let's talk for a minute about the high level concepts. When you are writing a program, you often have data in the form of variables, which store the values you are computing on. And code, when it manipulates that data according to the algorithm you have designed. Object-oriented programming is a paradigm of programming languages that groups data, and the code that manipulates it, together into logical units called objects. This language design aims to help programmers think about their program by grouping the code and data together into one logical unit. As you write larger and larger programs, this principle becomes more and more helpful. As you progress and learn more about Java, you will learn about a lot of important principles that go along with these ideas. For now, however, we're just going to start with the basics. Here, you can see an example of a class declaration. A class is a template that specifies how to make objects. Let's look at each piece of this declaration. The first line tells Java that we are declaring a class, called Point. As with variables, we have a lot of freedom in what we name classes that we create. But we should name them descriptevly. Here, we are making a class which represents a two-dimensional point, so Point is a good name for it.

1:28
Next we declare two fields, int x and int y. Field is the name for a variable that is inside of an object. They're also called instance variables since they are variables that are in each instance of the objects created from the class. These look like variable declarations except that the word private comes before them. Private means that only code inside of this class can directly manipulate these fields. You'll learn more about why that is important as you become more skilled in Java, but for now we'll just make all fields private. Next is the declaration of a constructor for the class. A constructor specifies how to create objects of this class. It is code that gets run when an object is created to initialize that object. Note that the constructor looks like a function but has no return type. And is named the same as the class. These are the hallmarks of a constructor declaration.

2:22
In front of the constructor, we have the word public, which means that any code can use this constructor to create a point.

2:29
After the constructor, we have three methods, getX, getY and distance. Methods are functions that are inside of classes. In Java everything is inside of a class, so technically all functions in Java are methods. These methods are invoked or called on a particular object and implicitly act on that object. You can see a method call here where the code says otherPt.getX. This calls the get X method on the other point object. It will get the X of that particular point object.

3:03
Last, we have the Declaration of a static method. These behave a little differently from regular methods. They don't act on any particular instance of a class, they just belong to the class in general. That concept is a little tricky and we'll explain it more later. This method is called main, which is a special method. If you run your programs outside of Blue Jay, main is the starting point. Execution begins in main before any objects are even created.

3:30
The ideas of objects are supposed to help programmers think about their data in terms of objects that make logical sense. For example, if we make a new point, we are creating an object which represents something we can concretely think about, in this case a point on a plane.

3:47
We can then make another point which has its own X and Y and represents a different instance of the same type of thing, another point on the plane. You can, of course, create as many of an object as you need for your algorithm. Once you have some objects, you can invoke methods on them, such as p1.distance(p3), which you can think of as asking p1 to compute the distance to p3. That is, you can think of this line of code as saying p1 go figure out how far you are from p3. You can think of the code that executes for this method call as logically belonging to the p1 object.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?