hey my name is Mike I'm a developer from Philadelphia and in this video I'm going to give you an introduction into object-oriented programming by answering the question what are classes and objects so without further ado let's get into it and we'll learn about these two really important concepts one of the best things about writing programs is being able to easily work with and keep track of different types of data in fact data is the driving force behind most programs that you'll write the programmers are constantly parsing strings adding and subtracting numbers counting things in passing different pieces of data around and all major applications being able to work with data in an organized and structured way is extremely important now by default most programming languages will allow you to store and keep track of three distinct types of data text numbers and boolean values with these three data types we can represent all different types of information things like people's names with text whether or not someone is an organ donor with a boolean value or the price of an item with a number and being able to keep track of simple pieces of data like this is really great but what happens when we need to work with more complex data we can represent a name with a string and a price with a number but what if we want to represent something more complex like an entity in the real world let's take a student for example imagine we were writing a program for a school and we wanted to keep track of the students who are currently enrolled so now instead of keeping track of names with strings or prices with numbers we want to be able to keep track of students but the problem is we don't have a datatype for students there's no student data type floating around that's built into the language like strings numbers or boolean x' on top of that we might also want to define some functionality around the student like giving school administrators the ability to mark a student as absent or change their grades for example and as we write more and more complex programs like the one for this school we'll want to be able to keep track of and model more complex pieces of data and therefore we need more complex data types in order to solve this problem we can use an object-oriented programming language an object-oriented programming language is a programming language which allows developers to create their own custom data types so even if the language doesn't provide a student data type by default a developer could create the student data type and then use it just like they would a normal string number or boolean value so if we use an object or in a programming language to write our school program we can easily keep track of all of the students by creating a custom student data type so how exactly does this work how can we go about creating our student data type and start creating students in our program the first step is to create a student class in object-oriented programming a class is a specification of the new data type it's essentially a blueprint that tells the programming language what the new data type looks like and what it's made up of in our student class we can define the attributes and the functionality that make up a student by creating a student class the attributes of a class are individual strings numbers or boolean z' which define the attributes of the entity we're modeling in this case a student might have a name attribute which would be a string a GPA attribute which would be a number a year attribute for freshman sophomore junior senior that might also be a string and then maybe it has scholarship boolean which would determine whether or not the student has a scholarship so you can see we can define all these different attributes of a student which themselves would be either strings numbers or boolean z' in addition to attributes we can also define functionality around the student so for example we might create a has honors function inside of the student class which would use the GPA of the student to determine if they qualify for honors so maybe if the student has a GPA of greater than 3.5 then we would say they have honors a week you create a give scholarship function which would set the has scholarship boolean to true thus giving the student a scholarship and so the class can have these attributes and these functions which will essentially make up the data type and allow us to define what a student looks like in our program when we create a class we tell the language what this new data type looks like and what it does but all we're doing here is describing the data type we're not actually creating a student that we can work with to create a student in our program we'll need an object an object is an instance of a class which means a student object would be an instance of our student class so a student object is an actual student with an actual name GPA year and scholarship status so in our program we might create three student objects with names Jack Kate and Sawyer and each of them would be an instance of our student class now the object is the actual thing that we're gonna work with in our programs you could pass it around you can store it inside of a variable you could access each one of those attributes or call any of those functions the object is the implementation of the class the class is just simply the blueprint it defines what the new data type looks like now the cool thing about classes and objects is that once we define what the data type looks like in the class we can create as many objects or instances of that class as we want so our program can keep track of thousands of student objects all with their own name GPA year etc now our code could also use the functions exposed by each of these student objects to for example give them scholarships or figure out if they have honors so if some other piece of code was responsible for granting scholarships to students it could use the give scholarship function for a particular student in order to do that so object-oriented programming is really great because it allows us to model real world entities like students in our programs so we can create a student class and then create instances of that student class student objects and use them now you could do the same thing with other entities in the real world it doesn't just have to be students you could model a light bulb you could model a microphone you can model an animal or a day of the week you can model anything that you can think of an object-oriented programming is great because it allows you to be flexible enough to model just about anything in the real world so that's a little bit about object-oriented programming it's an extremely popular way to write programs and most of the modern programming languages support it thanks for watching I hope you enjoyed the video and you learn something new and I'll see you