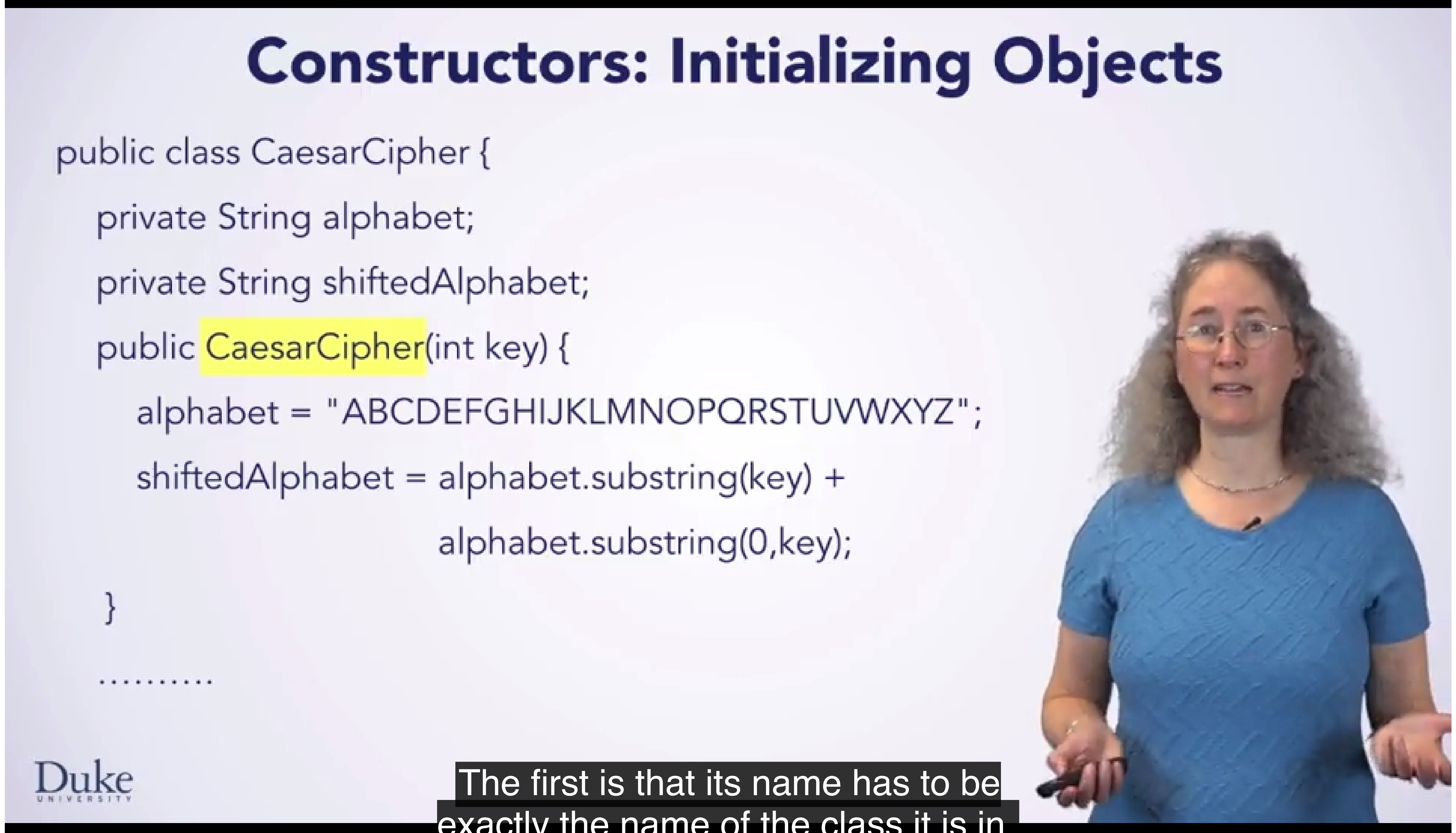

Constructors: Initializing Objects

```
public class CaesarCipher {
   private String alphabet;
   private String shiftedAlphabet;
   public CaesarCipher(int key) {
      alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
      shiftedAlphabet = alphabet.substring(key) +
                        alphabet.substring(0,key);
   }
   ..........
```

Duke UNIVERSITY

The first is that its name has to be exactly the name of the class it is in.

Constructors

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

Interactive Transcript

Search Transcript English

0:03
Welcome. The last object oriented concept you're going to learn in this lesson is constructors. Recall that the object oriented CaesarCipher we wrote had a constructor, which took the key as a parameter and initialized the fields on the object.

0:19
When you want to write a constructor, there are certain rules for declaring it. The first is that its name has to be exactly the name of the class it is in. Here, the constructor's name must be CaesarCipher because it's in a class called CaesarCipher. The second rule is that the constructor has no return type, not even void. Normally you would write a method's return type here between public and the name of the method, but for constructor you do not need to write anything between them.

0:51
Like a normal method, a constructor has its parameter list in parentheses. You can make constructors with any number and types of parameters you want. You can have no parameters to a constructor or several.

1:04
Here we have one parameter of type int, the key.

1:08
Finally, a constructor has a body, like a normal method. You can write whatever code you want in the constructor's body to specify how to initialize the object.

1:19
Constructors are not quite like normal methods, however. You do not call them directly. Instead, they are automatically as part of newing an object. When you create a new object, the constructor is invoked to initialize that object immediately after the new object is made. This is one of the great benefits of constructors. They allow you to specify how an object should be initialized, and you can be sure that code will always be run for every object, as soon as it is made. You do not have to worry about bugs in your code from forgetting to call some initialization code. What happens if you do not write a constructor for your class?

1:58
Whenever you write no constructors as you've been doing up to this point, the Java compiler will provide a default constructor for you. The constructor that the compiler provides looks like this. It is public, which you should recall, means any piece of code may use it to initialize an object.

2:17
It takes no arguments, so you would pass no arguments when you create new objects. And it does not do anything, there is no special initialization of the object.

2:27
Now that you know the rules about constructors, let's see how a constructor works. Let's suppose you had a line of code somewhere else in your program like this, CaesarCipher cc = new CaesarCipher(22). This line of code asks Java to make a new instance of the CaesarCipher class and initialize it by passing 22 to your constructor. Let us see what this does by starting when Java is just about to execute this line. The first thing it does is create a new variable, cc. Then it creates a new instance of the CaesarCipher class. That means you have a new object with its own copy of the fields of that class, alphabet and shiftedAlphabet. Then it calls the constructor, passing 22 for the key. Once inside the constructor, Java begins executing the code you wrote to initialize the object. This code initializes alphabet, then initializes shiftedAlphabet. Having finished the constructor, Java returns back to where you are creating the object. The key was just a parameter to the constructor so it only exists during that call. However, the fields are part of the object so they continue to exist in it.

3:38
And then finishes the assignment statement by initializing the cc variable to the newly created object. Great, now you know the basic rules of how to write constructors, what they do, and how they are used to initialize objects. Thank you.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

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