













Introduction to the Course

Implementing the Caesar Cipher

	Module Learning Outcomes / Resources	10 min
	A Brief History of Cryptography	5 min
	Introduction	5 min
	Creating and Manipulating Strings	5 min
	Counting Loops	9 min
	Character Class	5 min
	Developing an Algorithm	5 min
	Translating into Code	4 min
	Testing and Debugging	1 min
	Summary	40 sec
	Programming Exercise: Implementing the Caesar Cipher	10 min
	Practice Quiz: Implementing the Caesar Cipher	6 questions

Breaking the Caesar Cipher

Object Oriented Caesar Cipher

Review


Character Building

- The type char is primitive, single quotes
 - 'a', '1', ' ', ..., but using "a" is a String!
- The Character class has several methods
 - Character.toLowerCase('G') for example

Method Name	Functionality
isLowerCase(ch)	returns boolean if ch is 'a', 'b' ...
isDigit(ch)	returns boolean if ch is '0','1',...'9'
toLowerCase(ch)	returns lowercase version of ch
toUpperCase(ch)	returns uppercase version of ch

For example, this call will return a lowercase g,

Character Class



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

Interactive Transcript

Search Transcript

English

0:03
Hello. We'll introduce the class character, which you can use to determine properties of character values in Java. The type char is a primitive type, like int, boolean, and double. Some people pronounce it char, some people pronounce it car, and some people say care. But everyone says the word character the same way, so I typically use the careful pronunciation.

0:31
Char values are specified with single quotes. For example, you can see quote A quote and quote 1 quote. End quote, space quote here. These are character values. The value "a" is a string value. It's usually much easier to write the code than to say all these quote values. The character class has several methods you can use in writing code. You may remember the methods Integer.Parseint and Double.ParseDouble. These are methods of the classes Integer and Double, respectively. The method Character.toLowerCase returns a lowercase equivalent to its argument. [For example, this call will return a lowercase g](#), since an uppercase G is the argument to Character.toLowerCase. If you pass a character that is already in lowercase, the same value will be returned. The table shows Boolean value functions like, isLowerCase and isDigit, as well as conversion functions like toLowerCase and toUpperCase. Using the Java documentation, we'll show you more Boolean and conversion methods. Have fun building character, and writing code. We have the CharacterDemo class here in BlueJ, and we've got two methods that I'm going to run through and illustrate, and I'll add one very quickly. This first method, digitTest, creates a test string that has uppercase characters, lowercase characters, digits, and punctuation. Move through every character of the string and calls the character.isDigit method, a Boolean method. And the Character.isAlphabetic method, another Boolean method. So, let's run through digitTest, and see what it does. I'm going to create a new object on my work bench, by right-clicking. And then I'm going to run the digitTest method by right-clicking on that. And we can see here pretty clearly that A, B, C, uppercase characters, little a, little b, little c, those are all alphabetic characters. And then the digits are labeled as digit characters. Notice that no punctuation was printed, so that when I go back to my editor, we can see that the uppercase characters were all alphabetic. The characters that look like digits were all labeled as digits. And the punctuation wasn't any character in that it didn't have the label, alphabetic and it didn't have the label, digit. Just want to illustrate one quick thing here. I can also say if ch is equal to the character #, then I can print a message that, It's a hashtag. Highly enlightening.

3:29
And now if I compile this program, it compiled without any errors.

3:35
And I'll make another object.

3:39
I'll invoke the digitTest method. And we can see that lo and behold, # is a hashtag.

3:49
That's just a reminder that for characters we use single quotes to differentiate the values. Whereas strings use double quotes. We can see that here, where I've created another string test in the method conversionTest. I've created a similar string with uppercase characters, lowercase characters, digits and some punctuation. I'm going to loop through by using the string charAt method. To store a character variable ch. I'm creating an uch variable and an lch variable, both of type char. I'm creating them calling Character.toUpperCase which will return an uppercase character, and Character.toLowerCase that will return a lowercase equivalent. Remember that converting a digit to upper or lowercase doesn't change the digit at all. And if a character is already lowercase, converting it to lowercase leaves it alone. So, running that method, I will right click on my class and call conversionTest. We can see that I get the characters in my string on the left column. The function that you got by calling toUppercase and the results that you get by calling toLowercase. So I get character, uppercase, lowercase. You can see that in each column, I have all uppercase characters or digits and punctuation. All lowercase characters weren't considered punctuation. And as one quick review of that code, to remind you of where that came from,

5:19
you can see that I called toUppercase, toLowercase, and then printed them as the character, the uppercase version and the lowercase version. Using the Java documentation for characters will help a lot in making your program run smoothly, when you're using character values. Have fun building more character than you did last time.

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?