

Lab 03 Java Class Libraries

Objectives:

1. Strings in JAVA
2. Java Math Class

1: Strings in JAVA

Strings in Java are objects that are backed internally by a char. **String** is a predefined class in the Java library, just like the classes **System** and **Scanner**. The **String** type is **non primitive type**.

Syntax:

```
<String_Type> <string_variable> = "<sequence_of_string>";
```

Example:

```
String str = "oop";
```

The `java.lang.String` class provides a lot of methods to work on string. By the help of these methods, operations on string such as trimming, concatenating, converting, comparing, replacing strings etc can be performed.

Getting String Length

You can use the `length()` method to return the number of characters in a string.

Example:

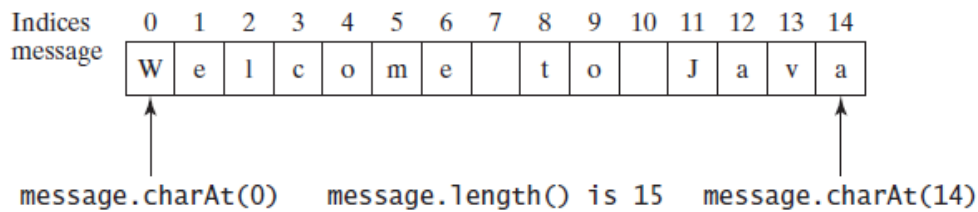
```
String message = "Welcome to Java";
```

```
System.out.println("The length of " + message + " is " + message.length());
```

Getting Characters from a String

The `s.charAt(index)` method can be used to retrieve a specific character in a string `s`, where the index is between `0` and `s.length()-1`.

For example, `message.charAt(0)` returns the character **W**, as shown in [figure](#).



Converting Strings

The `toLowerCase()` method returns a new string with all lowercase letters and the `toUpperCase()` method returns a new string with all uppercase letters.

Example:

"Welcome".`toLowerCase()` returns a new string **welcome**.

"Welcome".`toUpperCase()` returns a new string **WELCOME**.

The `trim()` method returns a new string by eliminating whitespace characters from both ends of the string. The characters ' ', \t, \f, \r, or \n are known as *whitespace characters*.

Method	Description
1. length()	Returns the length of String name. (5 in this case)
2. toLowerCase()	Converts all the characters of the string to the lower case letters.
3. toUpperCase()	Converts all the characters of the string to the upper case letters.
4. trim()	Returns a new String after removing all the leading and trailing spaces from the original string.
5. substring(int start)	Returns a substring from start to the end. i.e.substring(3)
6. substring(int start, int end)	Returns a substring from the start index to the end index. The start index is included, and the end is excluded.
7. replace('r', 'p')	Returns a new string after replacing r with p. (This method takes char as argument)
8. startsWith("Ha")	Returns true if the name starts with the string "Ha".

9. endsWith("ry")	Returns true if the name ends with the string "ry".
10. charAt(2)	Returns the character at a given index position.
11. indexOf("s")	Returns the index of the first occurrence of the specified character in the given string.
12. lastIndexOf("r")	Returns the last index of the specified character from the given string.
13. equals("anystring")	Returns true if the given string is equal to the given string false otherwise [Case sensitive]
14.equalsIgnoreCase("harry")	Returns true if two strings are equal, ignoring the case of characters.

2: JAVA Math Class

The Java programming language supports basic arithmetic with its arithmetic operators: +, -, *, /, and %. The Math_class provides methods and constants for doing more advanced mathematical computation.

The Math is located in the [java.lang](#) package, and not in the java.math package. Thus, the fully qualified class name of the Math class is [java.lang.Math](#)

The methods in the Math class are all static, so you call them directly from the class, like this:

[Math.cos\(angle\);](#)

***Note:** Using the static import language feature, you don't have to write Math in front of every math function: import static java.lang.Math;

This allows you to invoke the Math class methods by their simple names.

For example: [cos\(angle\);](#)

Constants

The Math class includes two constants:

- [Math.E](#), which is the base of natural logarithms, and
- [Math.PI](#), which is the ratio of the circumference of a circle to its diameter.

Basic Math Methods

The Math class includes more than 40 static methods. They can be categorized as *trigonometric methods*, *exponent methods*, and *service methods*. Service methods include the rounding, min, max, absolute, and random methods.

Trigonometric Methods

The Math class contains the following methods.

<i>Method</i>	<i>Description</i>
<code>sin(radians)</code>	Returns the trigonometric sine of an angle in radians.
<code>cos(radians)</code>	Returns the trigonometric cosine of an angle in radians.
<code>tan(radians)</code>	Returns the trigonometric tangent of an angle in radians.
<code>toRadians(degree)</code>	Returns the angle in radians for the angle in degree.
<code>toDegree(radians)</code>	Returns the angle in degrees for the angle in radians.
<code>asin(a)</code>	Returns the angle in radians for the inverse of sine.
<code>acos(a)</code>	Returns the angle in radians for the inverse of cosine.
<code>atan(a)</code>	Returns the angle in radians for the inverse of tangent.

The parameter for `sin`, `cos`, and `tan` is an angle in radians. The return value for `asin`, `acos`, and `atan` is a degree in radians in the range between $-\pi/2$ and $\pi/2$.

- One degree is equal to $\pi/180$ in radians,
- 90 degrees is equal to $\pi/2$ in radians
- 30 degrees is equal to $\pi/6$ in radians.

Exponent Methods

There are five methods related to exponents in the `Math` class.

<i>Method</i>	<i>Description</i>
<code>exp(x)</code>	Returns e raised to power of x (e^x).
<code>log(x)</code>	Returns the natural logarithm of x ($\ln(x) = \log_e(x)$).
<code>log10(x)</code>	Returns the base 10 logarithm of x ($\log_{10}(x)$).
<code>pow(a, b)</code>	Returns a raised to the power of b (a^b).
<code>sqrt(x)</code>	Returns the square root of x (\sqrt{x}) for $x \geq 0$.

The Rounding Methods

The `Math` class contains five rounding methods

<i>Method</i>	<i>Description</i>
<code>ceil(x)</code>	x is rounded up to its nearest integer. This integer is returned as a double value.
<code>floor(x)</code>	x is rounded down to its nearest integer. This integer is returned as a double value.
<code>rint(x)</code>	x is rounded up to its nearest integer. If x is equally close to two integers, the even one is returned as a double value.
<code>round(x)</code>	Returns <code>(int)Math.floor(x + 0.5)</code> if x is a float and returns <code>(long)Math.floor(x + 0.5)</code> if x is a double.

The Service Methods

The `min`, `max`, and `abs` Methods

The `min` and `max` methods return the minimum and maximum numbers of two numbers (`int`, `long`, `float`, or `double`).

For example, `max(4.4, 5.0)` returns `5.0`, and `min(3, 2)` returns `2`.

The `abs` method returns the absolute value of the number (`int`, `long`, `float`, or `double`).

This method generates a random `double` value greater than or equal to 0.0 and less than 1.0 (`0 <= Math.random() < 1.0`). You can use it to write a simple expression to generate random numbers in any range.

3: JAVA Random Class

Import Random class available in Java.util Package

Import java.util.Random;

Creating object of Random Class:

Random rand=new Random();

Using methods of Random Class:

Rand.nextInt();

Rand.nextInt(int);//range

Lab Tasks:

Exercises

1. Write a Java program to concatenate a given string to the end of another string
2. Write a Java program that reads two floating-point numbers and tests whether they are the same up to three decimal places.
3. Write a Java program to replace each substring of below given sample string
 - a. Sample string : "The quick brown fox jumps over the lazy dog."
 - b. In the above string replace all the fox with cat
4. Write a program which solves quadratic equations of the form: $ax^2 + bx + c = 0$. Values of a, b, c can be taken as input from user.
5. Write a Java program to round up the result of variable division.

6. Game: heads or tails

Write a program that lets the user guess whether the flip of a coin results in heads or tails. The program randomly generates an integer 0 or 1, which represents head or tail. The program prompts the user to enter a guess and reports whether the guess is correct or incorrect.

7. Random Sentences

Write an application that uses random-number generation to create sentences. Use four arrays of strings called article, noun, verb and preposition. Create a sentence by selecting a word at random from each array in the following order: article, noun, verb, preposition, article and noun. As each word is picked, concatenate it to the previous words in the sentence.

The words should be separated by spaces. The application should generate and display 20 sentences.

The article array should contain the articles "the", "a", "one", "some" and "any"; the noun array should contain the nouns "boy", "girl", "dog", "town" and "car"; the verb array should contain the verbs "drove", "jumped", "ran", "walked" and "skipped"; the preposition array should contain the prepositions "to", "from", "over", "under" and "on".

8. Wind-chill Temperature

How cold is it outside? The temperature alone is not enough to provide the answer. Other factors including wind speed, relative humidity, and sunshine play important roles in determining coldness outside.

In 2001, the National Weather Service (NWS) implemented the new wind-chill temperature to measure the coldness using temperature and wind speed. The formula is

$$t_{wc} = 35.74 + 0.6215t_a - 35.75v^{0.16} + 0.4275t_av^{0.16}$$

where t_a is the outside temperature measured in degrees Fahrenheit and v is the speed measured in miles per hour. t_{wc} is the wind-chill temperature. The formula cannot be used for wind speeds below 2 mph or temperatures below -58 °F or above 41°F.

Write a program that prompts the user to enter a temperature between -58 °F and 41°F and a wind speed greater than or equal to 2 and displays the wind-chill temperature. (**Hint:** Use `Math.pow(a, b)` to compute $v^{0.16}$.)

9. Generate vehicle plate numbers

Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.