

# LAB MANUAL

CS1421 Object Oriented Programming Lab

**Instructor**

Aman Farooq

Lab 3

Java Language Specification & Procedural Programming Constructs using Java

**Objective**

After completing this lab, the students should be able to

* Understand the difference between primitive data types and user-defined/reference type data types
* Understand the conditional statements
* Understand the usage of mathematic functions
* Understand the for, while and do while loops
* Learn how to take user’s input

**Relevant Lecture Material**

**Textbook:** Introduction to Java Programming, Daniel Liang

**Chapters:** 2, 3,4,5

**Arrays**

 An array stores a sequence of values that are all of the same type. We want not just to store values but also to be able to quickly access each individual value. The length of an array is established when the array is created. After creation, its length is fixed. Each item in an array is called an *element*, and each element is accessed by its numerical *index*. The method that we use to refer to individual values in an array is to number and then index them—if we have n values, we think of them as being numbered from 0 to n−1.

Making an array in a Java program involves three distinct steps:

* Declare the array name.
* Create the array.
* Initialize the array values.

We refer to an array element by putting its index in square brackets after the array name.

To use an array in a program, you must declare a variable to reference the array and specify

the array’s *element type*. Here is the syntax for declaring an array variable:

**Declaration:** elementType[] arrayRefVar;

e.g. **int**[] num;

The **elementType** can be any data type, and all elements in the array will have the same data type.

Unlike declarations for primitive data type variables, the declaration of an array variable does not allocate any space in memory for the array. It creates only a storage location for the reference to an array. If a variable does not contain a reference to an array, the value of the variable is **null**. You cannot assign elements to an array unless it has already been created. After an array variable is declared, you can create an array by using the **new** operator and assign its reference to the variable with the following syntax:

**Creation:** arrayRefVar = **new** elementType[arraySize];

e.g. num= **new int**[5];

**Initialization:**  arrayRefVar[index]=arrayelement;

e.g. num[0]=2;…

**Shorthand notation:**

Java has a shorthand notation, known as the *array initializer*, which combines the declaration, creation, and initialization of an array in one statement using the following syntax:

elementType[] arrayRefVar = {value0, value1, ..., value*k*};

**e.g** int[] num={2,4,6,8,10,12};

**Arrays Class**

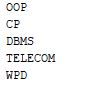
Arrays class which is in java.util.Arrays package, is a provision by Java that provides you a number of methods through which arrays can be manipulated. This class also lets you perform sorting and searching operations on an array.

**Example:**

**import** java.util.Arrays;

**public class** array1 {  
 **public static void** main(String[] args)  
 {  
 String[] arr={**"OOP"**,**"CP"**,**"DBMS"**,**"TELECOM"**,**"WPD"**};  
 **for**(**int** i=0; i<=arr.**length**-1;i++)  
 {  
  
 System.***out***.println(arr[i]);  
 }  
 }  
}

**Output:**

****

**Java Math Class**

The Java programming language supports basic arithmetic with its arithmetic operators: +, -, \*, /, and %. The [Math](https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html)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](https://docs.oracle.com/javase/tutorial/java/package/usepkgs.html#staticimport)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:

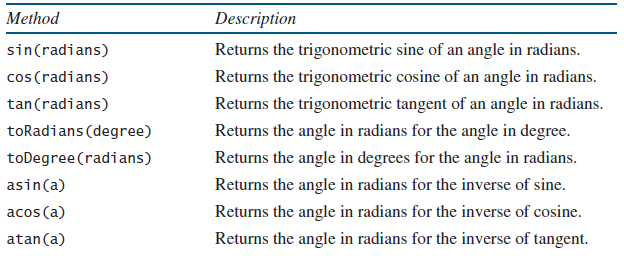
* Math.E, (2.718) which is the base of natural logarithms, and
* Math.PI, (3.14) which is the ratio of the circumference of a circle to its diameter.

**Basic Math Mehtods**

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**

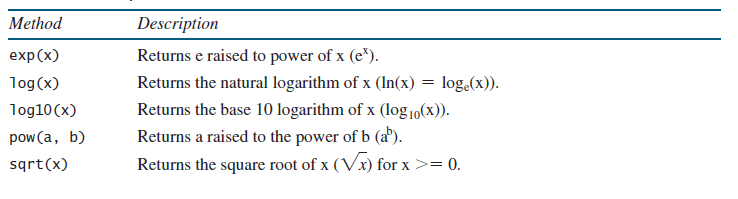
The Mathclass contains the following methods.



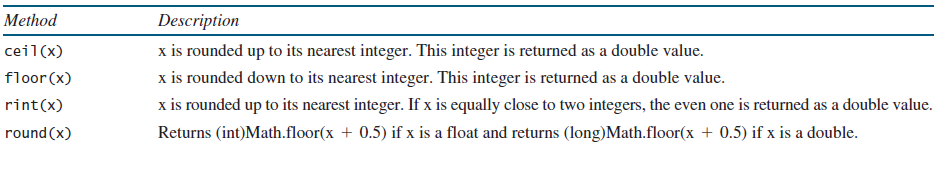
The parameter for sin, cos, and tanis an angle in radians. The return value for asin, acos, and atanis a degree in radians in the range between -pi/2 and pi/2. One degree is equal to pi/180 in radians, 90 degrees is equal to pi/2 in radians, and 30 degrees is equal to pi/6 in radians.

**Exponent Methods**

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



**The Rounding Methods**

The Mathclass contains five rounding methods

**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.

**Java String Class**

The **char** type represents only one character. To represent a string of characters, use the data type called **String**. For example, the following code declares **message** to be a string with the value **"Welcome to Java"**.

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

**String** is a predefined class in the Java library, just like the classes **System** and **Scanner**. The **String** type is not a primitive type. It is known as a *reference type*. Any Java class can be used as a reference type for a variable. The variable declared by a reference type is known as a reference variable that references an object. Here, **message** is a reference variable that references a string object with contents **Welcome to Java**.

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

**Getting String Length**

You can use the **length()** method to return the number of characters in a string. For example, the following code

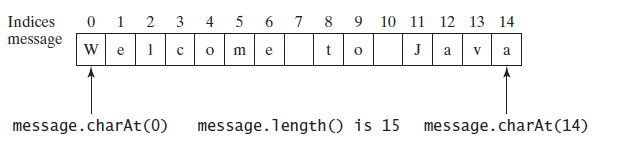
String message = **"Welcome to Java"**;

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

**C:\Users\support\Desktop\string length.JPG**

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. Note that the index for the first character in the string is **0**.

****

**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.

For 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(Escape Sequence Characters)*.

**The Date Class**

Java provides a system-independent encapsulation of date and time in the **java.util.Date** class.

**Practice Task:**

**Task 1:**

**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.

**Task 2:**

**Tax Calculation**

Write a program to calculate income tax by using the following rules.

**Inputs** to be taken from user (Income, age)

* If age is less than 60
* If income is less than or equals to 250000 tax percentage is 0.
* If income is greater than or equals to 250001 and less than or equals to 500000 tax percentage is 10.
* If income is greater than or equals to 500001 and less than or equals to 1000000 tax percentage is 20.
* If income is greater than or equals to 1000001 tax percentage is 30.
* If age is greater than or equals to 60 and less than 80
* If income is less than or equals to 300000 tax percentage is 0.
* If income is greater than or equals to 300001 and less than or equals to 500000 tax percentage is 10.
* If income is greater than or equals to 500001 and less than or equals to 1000000 tax percentage is 20.
* If income is greater than or equals to 1000001 tax percentage is 30.
* If age is equals to or greater than 80
* If income is less than or equals to 500000 tax percentage is 0.
* If income is greater than or equals to 500001 and less than or equals to 1000000 tax percentage is 20.
* If income is greater than or equals to 1000001 tax percentage is 30.

**Task 3:**

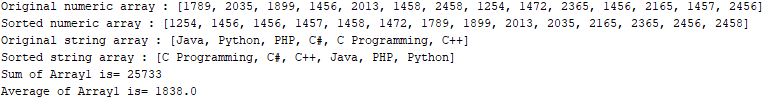
Suppose the tuition fee of a university is Rs.122600 this year and the fee increases 6.5% every year. In how many years will the fee be doubled?

**Task 4:**

Write a Java program in which user input month number and year to find the number of days in a month.

**Task 5:**

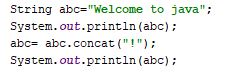
Write a Java program to sort a numeric array and a string array. Also calculate sum and average of numeric array.

****

**Task 6**

**Beautifying the sentences**

Write an application that takes a sentence from user and checks if the sentence stars from a capital letter and ends with a full stop. If it doesn’t, the program should add it.

Hint.

Output

