**National University of Computer & Emerging Sciences, Karachi**

**Computer Science Department**

**Spring 2022, Lab Manual – 02**

|  |  |
| --- | --- |
| **Course Code: CL-217** | **Course : Object Oriented Programming Lab** |

**LAB - 2**

# Operators, Control Structures, loops Math Library & user define Functions in JavaCONTENTS:

# Operators

# Control structures.

# If statements

# If else statements

# If else if statements

# For loop

# While loop

# Do while loop

# Java Math class

# Functions

# Operators

# Operators are classified into different types shown below:

# Ternary operator:

# bool data= age>18? “can vote”: “cannot vote”

# Control structures:

Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear. However, [Java](https://www.javatpoint.com/java-tutorial) provides statements that can be used to control the flow of Java code. Such statements are called control flow statements. It is one of the fundamental features of Java, which provides a smooth flow of program.

* Decision making (if statements, if else statements, switch statements etc.)
* Loop statements (while, do while, for, foreach)
* Jump statements (break, continue)

# Decision making statements:

In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false. In Java, there are four types of if-statements, i.e. if statement, if-else statements, if else-if ladder, nested if statements.

# 

Figure : if statement

# 

Figure :if else statements

# 

Figure :if else if statements

# 

Figure : nested if statements

If an if statements is placed with in the else statement then it is termed as if-else if ladder. Consider it a DIY exercise.

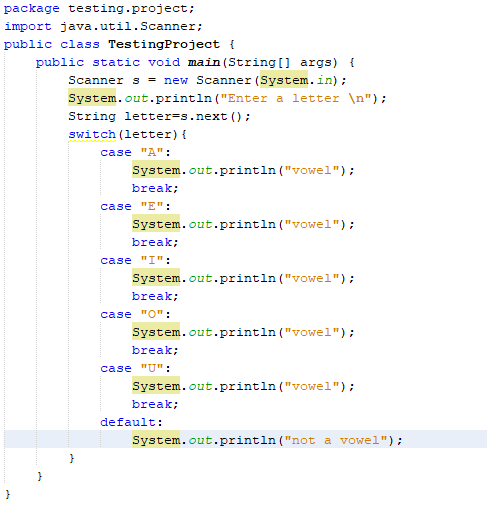


Figure : Switch case

Will cover break statement in next sections.

# 3.2 Loops:

In programming, sometimes we need to execute the block of code repeatedly while some condition evaluates to true. However, loop statements are used to execute the set of instructions in a repeated order. The execution of the set of instructions depends upon a particular condition.

In Java, we have three types of loops namely for loop, while loop and do while loop.

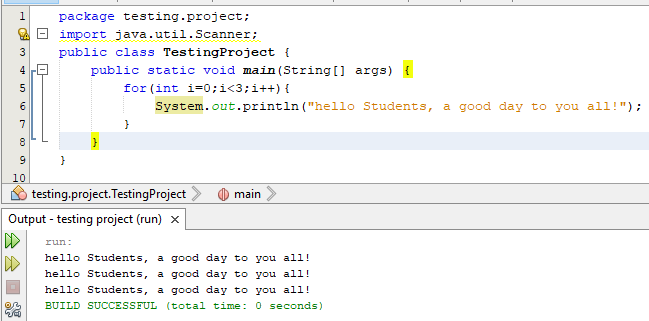


Figure : for loop

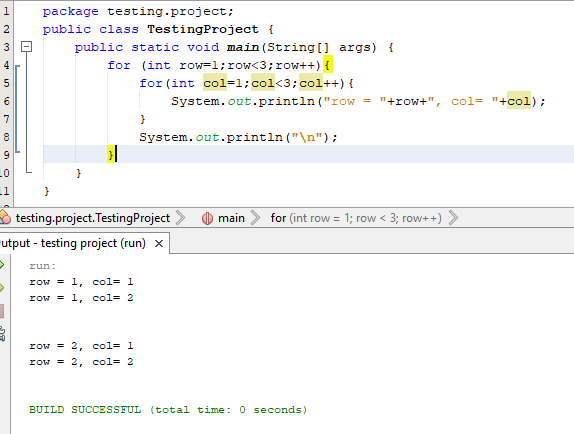


Figure :nested for loop

While loop will check the condition first and then execute the statements if condition is true.

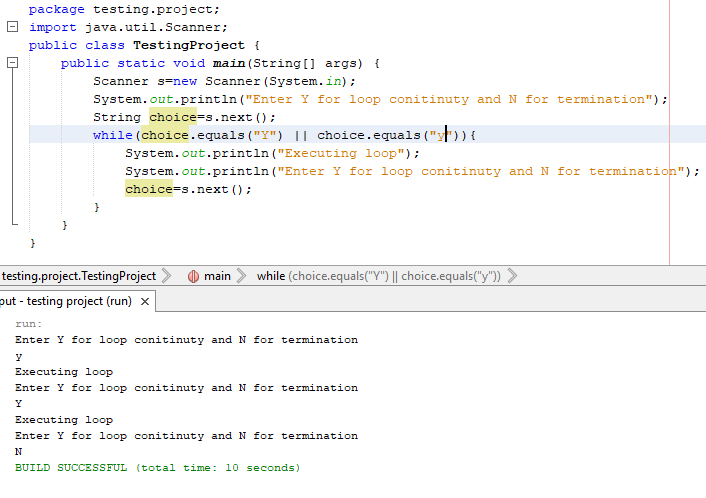


Figure : while loop

Do while will execute loop body once and then check the condition.

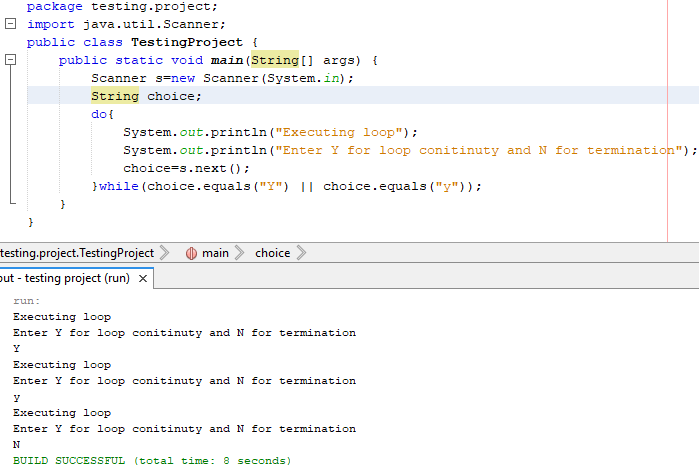


Figure : do while loop

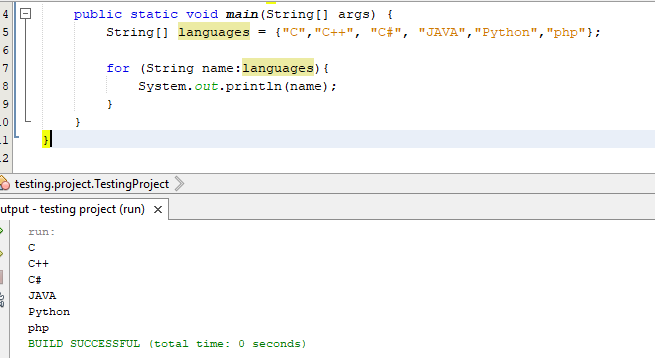
For-each is another array traversing technique like for loop, while loop, do-while loop introduced in Java5. It has a slightly different syntax than for loop shown below:

Figure 10:foreach loop

# 3.3 jump statements:

The break and continue statements are jump statements that are used to bypass some loop statements or finish the loop without verifying the test expression. These statements can be inserted into any loop, including for, while, and do-while loops.

The **break** statement in java is used to terminate from the loop immediately. When a break statement is encountered inside a loop, the loop iteration stops there, and control returns from the loop immediately to the first statement after the loop. Basically, break statements are used in situations when we are not sure about the actual number of iterations for the loop, or we want to terminate the loop based on some condition.

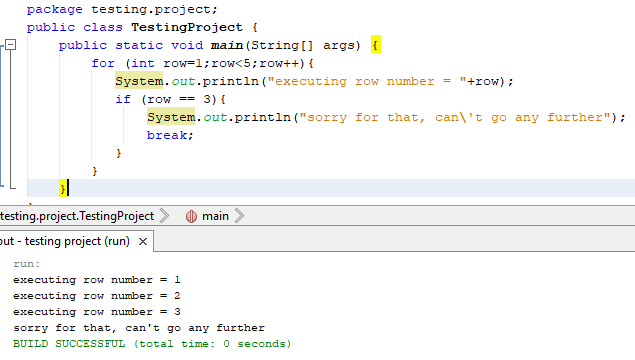


Figure : break statement

The **continue** statement in Java is used to skip the current iteration of a loop. Generally, they are used in the situations when we want to continue the loop but do not want the remaining statement after the continue statement to executed.

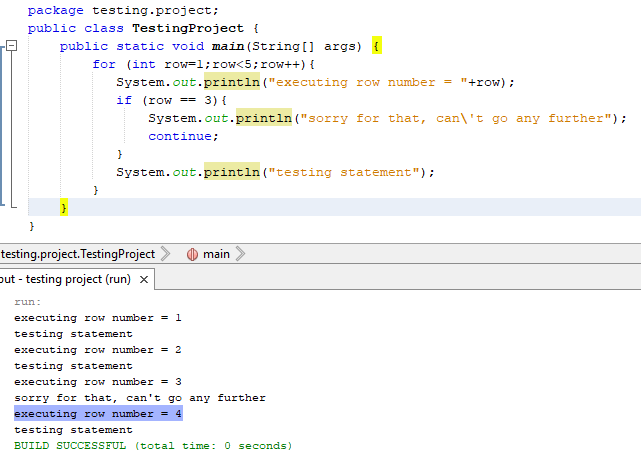


Figure : continue statement

There is no go to statement in Java but still goto is a reserved keyword. If they want to add it to the later versions. For now, the break statements use labels to jump to a specific line of code.

# Java Math Class:

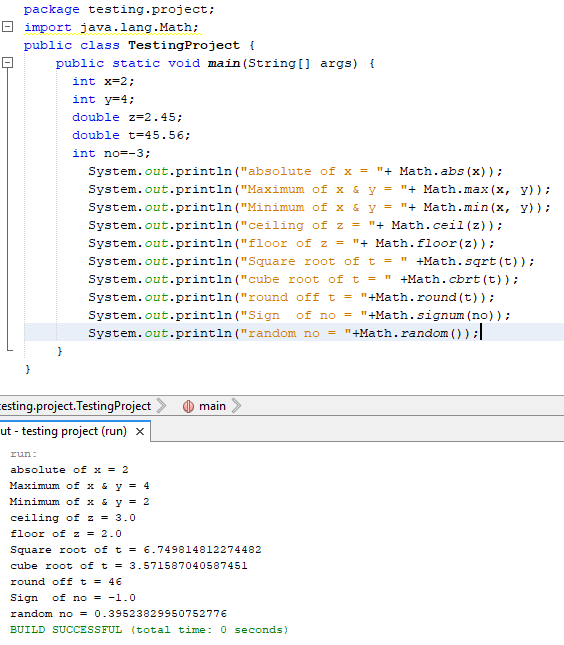
Java Math class provides several methods to work on math calculations like min(), max(), avg(), sin(), cos(), tan(), round(), ceil(), floor(), abs() etc.

Figure 13: math library functions

|  |  |
| --- | --- |
| Function | Description |
| Math.abs(x) | It will return the Absolute value of the given value. |
| Math.min(x,y) | It returns the Largest of two values |
| Math.max(x,y) | It is used to return the Smallest of two values. |
| Math.round(x) | It is used to round of the decimal numbers to the nearest value. |
| Math.sqrt(x) | It is used to return the square root of a number. |
| Math.cbrt(x) | It is used to return the cube root of a number. |
| Math.pow(x,y) | It returns the value of first argument raised to the power to second argument. |
| Math.signum(x) | It is used to find the sign of a given value. |
| Math.ceil(x) | Compute ceiling of a number (round up forward to nearest decimal place) |
| Math.floor(x) | Compute floor of a number (round off backward to nearest decimal place) |
| Math.random(x) | generate a positive random number between 0 to 1. |
| Math.sin(x) | It is used to return the trigonometric Sine value of a Given double value. |
| Math.log(x) | It returns the natural logarithm of a double value. |

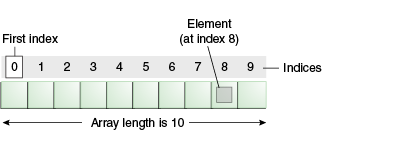
# Arrays

**Syntax to Declare an Array in Java**

1. datatype[] arr; (or)
2. dataType []arr; (or)
3. dataType arr[];

**Instantiation of an Array in Java**

1. arrayRefVar=**new** datatype[size];



# Functions:

**public** **static** **void** sum (int n1, int n2){

int sum = n1 + n2;

    System.out.println("Sum"+ sum);

    }

# LAB TASKS:

# TASK – 01:

Body Mass Index (BMI) Calculator: Body Mass Index (BMI) is a person's weight in kilograms divided by the square of height in meters. Formula given for reference.

Create a BMI calculator application that reads the user’s weight in kilograms and height in meters, then calculates and displays the user’s body mass index. The program should ask the user to input values at run-time. Also, the application should display the following information so the user can evaluate his/her BMI:

Input: weight in Kg, height in meters

Output: given below in tabular form

|  |  |
| --- | --- |
| BMI values | Output |
| Less than 18.5 | Underweight |
| 18.5 – 24.9 | Normal |
| 25 – 29.9 | Overweight |
| 30 or above | Obese |

## TASK – 02:

This programming exercise demonstrates a program that calculates a customer’s bill for a local cable company. There are two types of customers: residential and business. So, billing rates would vary for residential customers and business customers.

* For residential customers, the following rates apply:

Bill processing fee: $4.50

Basic service fee: $20.50

Premium channels: $7.50 per channel.

* For business customers, the following rates apply:

Bill processing fee: $15.00

Basic service fee: $75.00 for first 10 connections, $5.00 for each additional connection

Premium channels: $50.00 per channel for any number of connections

The program should ask the user for an account number (an integer) and a customer

code. Assume that R or r stands for a residential customer, and B or b stands for a

business customer in customer code.

Input: The customer’s account number, customer code, number of premium channels to which the user subscribes, and, in the case of business customers, number of basic service connections.

Output: Customer’s account number and the billing amount.

## TASK – 03:

Take numbers as input from user. Write a Java program to interchange the contents of both variables.

## TASK – 04:

Write a program to check if the number is a prime number or not.

## TASK – 05:

Write a program which calculates the grace marks for a student in Object oriented programming course using switch case. The user should enter the grade obtained by the student and the number of assignments he has failed in.

* If the student gets A grade and the number of assignments, he failed in is greater than 3, then he does not get any grace. If the number of assignments he failed in is less than or equal to 3 then the grace is of 5 marks.
* If the student gets B grade and the number of assignments, he failed in is greater than 2, then he does not get any grace. If the number of assignments he failed in is less than or equal to 2 then the grace is of 4 marks.
* If the student gets C grade and the number of assignments, he failed in is greater than 1, then he does not get any grace. If the number of assignments he failed in is equal to 1, then the grace is of 5.

Display the appropriate output in each case.

## TASK – 06:

Create a menu driven transactions’ processing unit in a bank as long as user enters correct input. Initially, take the username and password as input from user. If that username and password is correct then the user will log in to the system and can perform any transaction as per the menu shown below.

Main Menu \*\* Welcome to Bank of Pakistan\*\*

* + - 1. Deposit Money
      2. Withdraw Amount
      3. Account status

Select your choice: \_\_\_\_

(After completing the selected transaction) Do you want to continue? [y/Y]

(goes to Main Menu, if y/Y is pressed)

Depending upon the user’s choice, perform the transaction and display the remaining account balance along with the owner’s username.

## TASK – 07:

Write a program to print the circumference and area of a circle of radius entered by user by defining your own method.

## TASK – 08:

Write a Java program to add two numbers without using any arithmetic operators

## TASK – 09:

Program to copy all elements of one array into another array & find frequency of each element.

## TASK – 10:

Write a program to print the factorial of a number by defining a method named 'Factorial'.

Factorial of any number n is represented by n! and is equal to 1\*2\*3\*.... \*(n-1) \*n. E.g.-

4! = 1\*2\*3\*4 = 24

3! = 3\*2\*1 = 6

2! = 2\*1 = 2

Also,

1! = 1

0! = 0

## TASK – 11:

# Write a Java method to count all vowels in a string