JAVA Lab

Lab Experiment number 1

Roll no. **01** Batch **A**

Aim:

Implement JAVA programs based on the concept of classes, objects and control structures

Theory:

Classes and Objects are basic concepts of Object Oriented Programming which revolve around the real life entities

=> Class :-

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. Modifiers : A class can be public or has default access
2. Class name: The name should begin with an initial letter (capitalized by convention).
3. Superclass: The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
4. Interfaces(if any): A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. Body: The class body surrounded by braces, { }.

=>Objects:-

It is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of :

1. State : It is represented by attributes of an object. It also reflects the properties of an object.
2. Behavior : It is represented by methods of an object. It also reflects the response of an object with other objects.
3. Identity : It gives a unique name to an object and enables one object to interact with other objects.

=>Control statements:

Simple if statement

The if statement determines whether a code should be executed based on the specified condition.

```
if (condition) {
Statement 1; //executed if condition is true
}
Statement 2; //executed irrespective of the condition
```

Output:

If statement!

Hello World!

If..else statement

In this statement, if the condition specified is true, the if block is executed. Otherwise, the else block is executed.

Example:

```
public class Main
{
public static void main(String args[])
{
int a = 15;
if (a > 20)
System.out.println("a is greater than 10");
else
System.out.println("a is less than 10");
System.out.println("Hello World!");
}
}
}
```

Output:

a is less than 10

Hello World!

***

Ladder of If..else

```
if (score >= 90)
    grade = ‘A’;
else if (score >= 80)
    grade = ‘B’;
else if (score >= 70)
```

```
    grade = 'C';
 else if (score >= 60)
    grade = 'D';
 else
    grade = 'F';
```

***

Nested if statement

- An if present inside an if block is known as a nested if block. It is similar to an if..else statement, except they are defined inside another if..else statement.

```
if (condition1) {
        Statement 1; //executed if first condition is true
if (condition2) {
        Statement 2; //executed if second condition is true
}
else {
        Statement 3; //executed if second condition is false }}
```

Switch Statements

1. The switch-expression must yield a value of char, byte, short, or int type and must always be enclosed in parentheses.
2. The keyword break is optional, but it should be used at the end of each case in order to terminate the remainder of the switch statement.
3. If the break statement is not present, the next case statement will be executed.
4. The default case, which is optional, can be used to perform actions when none of the specified cases is true.
5. The order of the cases (including the default case) does not matter. However, it is a good programming style to follow the logical sequence of the cases and place the default case at the end.

```
switch (year) {
        case 7: annualInterestRate = 7.25;
                break;
        case 15: annualInterestRate = 8.50;
                break;
```

```
        case 30: annualInterestRate = 9.0;
                 break;
        default: System.out.println("Wrong number of years, enter 7, 15, or 30");
}
```

***

=>Iteration Statement

Statements that execute a block of code repeatedly until a specified condition is met are known as Iteration or looping statements.

Java provides the user with three types of looping statements:

1. while statement
2. do-while statement
3. For statement

while Loop

Known as the most common loop, the while loop evaluates a certain condition. If the condition is true, the code is executed. This process is continued until the specified condition turns out to be false.

```
int i = 0;
while (i < 100) {
        System.out.println("Welcome to Java!");
        i++;
}
```

```
int i = 5;
while (i <= 15) {
        System.out.print(i);
        i = i+2;
}
```

Output: 5 7 9 11 13 15

do-while loop

The do-while loop is similar to the while loop, the only difference being that the condition in the do-while loop is evaluated after the execution of the loop body. This guarantees that the loop is executed at least once.

```
do { // Loop body;}
 while (continue-condition);
```

Eg.

```
public class Main {
        public static void main(String args[]) {
                int i = 1;
                do {
                        System.out.println(i);
                        i = i+1;
                } while (i <= 5);
        }
}
```

Output : 12345

For statement

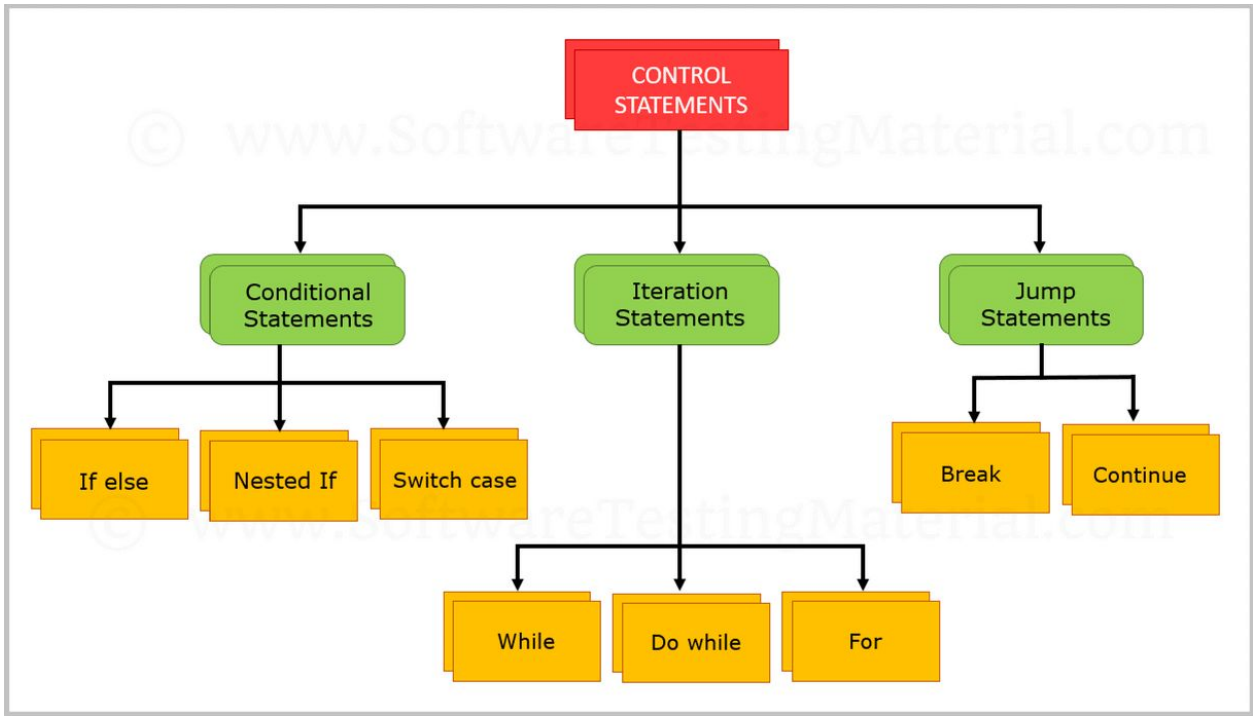
The for loop in java is used to iterate and evaluate a code multiple times. When the number of iterations is known by the user, it is recommended to use the for

loop.
Syntax:

```
for (initialization ; condition ; increment/decrement) {
        Statements;
}
```

Example

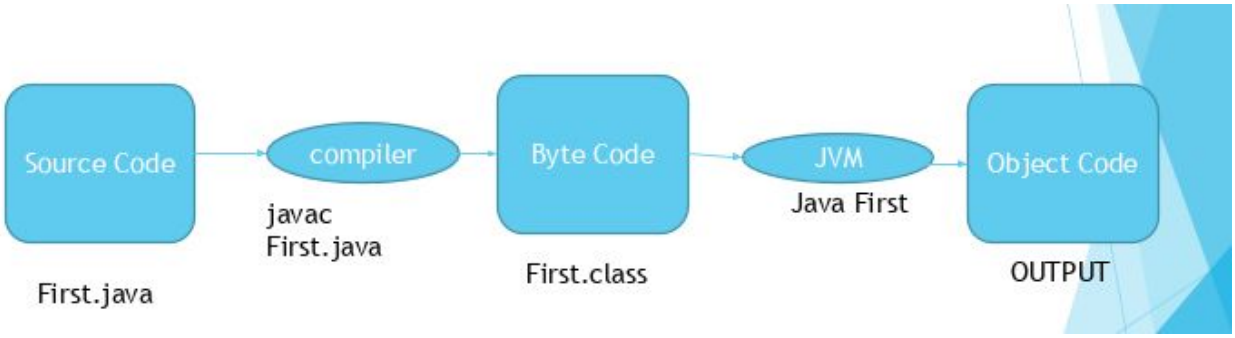
```
int i;
for (i = 0; i<100; i++) {
        System.out.println("Welcome to Java");
}
```



***

=>Compilation and Execution of Programs

First we create a word file with any text editor, type the java program and save the file as all files, with extension of .java.
Using command prompt, locate the file.
Use command “javac <file-name.java> to compile
After successful compilation, class file is created
Use command “java <file-name>” to run the java program



***

Programs:

Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Write a Java program to take as input the speed of each racer and print back the speed of qualifying racers. (make use of array)

//code

```
import java.util.*;
class Race {
	public static void main(String args[]) {
		Scanner sc = new Scanner(System.in);
		float speed[], qualified[];
		speed = new float[10];
		qualified = new float[10];
		float avg=0;
		int k=0;
		//input
		System.out.println("Enter speed of bikers :  ");
		for(int i=0 ; i<5 ; i++) {
			speed[i] = sc.nextFloat();
		}
		//find average;
		for (int i=0 ; i<5 ; i++) {
			avg += speed[i];
		}
		avg = avg/5;
		//check qualified
		for (int i=0 ; i<5 ; i++) {
			if (speed[i] > avg) {
				qualified[k++] = speed[i];
			}
		}
		//printing qulified speed
		System.out.println();
```

```
                System.out.println("Qualified bikers with speed are :  ");
                for(int i=0 ; i<qualified.length && qualified[i]!=0 ; i++) {
                        System.out.print(" "+qualified[i]+" ");
                }
        }
}
```

//output

```
E:\Aamir\Sem-3\OOPM\Lab Assignment 1>java Race
Enter speed of bikers:
18.2  20  12.3  17  10.2

Qualified bikers with speed are :
 18.2  20.0  17.0
E:\Aamir\Sem-3\OOPM\Lab Assignment 1>
```

****

```
E:\Aamir\Sem-3\OOPM\Lab Assignment 1>java Race
Enter speed of bikers:
4  12  14.2  10  8

Qualified bikers with speed are :
 12.0  14.2  10.0
E:\Aamir\Sem-3\OOPM\Lab Assignment 1>
```

****

Implement a java program to calculate gross salary & net salary taking the following data. Input: empno, empname, basic process: DA=70% of basic, HRA=30% of basic, CCA=Rs240/-, PF=10% of basic, PT= Rs100/-

//code

```
import java.util.*;

class Salary {
```

```
        public static void main(String args[]) {
                Scanner sc = new Scanner(System.in);
                float basic, da, hra, gross, cca=240, pf, pt=100, deduction, net;
                System.out.println(“Enter employee number :  “);
                int empno = sc.nextInt();
                sc.nextLine();
                System.out.println(“Enter employee name :  “);
                String empname = sc.nextLine();
                System.out.println(“Enter employee’s basic income :  “);
                basic = sc.nextFloat();
                da = 70*basic/100;
                hra = 30*basic/100;
                pf = 10*basic/100;
                gross = basic+da+hra+cca;
                deduction = pf+pt;
                net = gross - deduction;
                System.out.println("Employee with  Employee number: "+empno+"** Name:
"+empname+" ** With basic salary of "+basic+" has");
                System.out.println("Gross Salary  :  "+gross);
                System.out.println("Net Salary  :  "+net);
        }
}



//output
```

```
E:\Aamir\Sem-3\OOPM\Lab Assignment 1>javac Salary.java

E:\Aamir\Sem-3\OOPM\Lab Assignment 1>java Salary
Enter employee number :
1
Enter employee name :
Aamir
Enter employee's basic income :
55000
Employee with  Employee number: 1** Name: Aamir ** With basic salary of 55000.0 has
Gross Salary  :  110240.0
Net Salary  :  104640.0

E:\Aamir\Sem-3\OOPM\Lab Assignment 1>java Salary
Enter employee number :
62
Enter employee name :
Krishna
Enter employee's basic income :
70000
Employee with  Employee number: 62** Name: Krishna ** With basic salary of 70000.0 has
Gross Salary  :  140240.0
Net Salary  :  133140.0
```

Thank you