Experiment No.1
Basic programming constructs like branching and looping
Date of Performance:
Date of Submission:

Aim :- To apply programming constructs of decision making and looping.

Objective :- To apply basic programming constructs like Branching and Looping for solving arithmetic problems like calculating factorial of a no entered by user at command prompt .

Theory :-

Programming constructs are basic building blocks that can be used to control computer programs. Most programs are built out of a fairly standard set of programming constructs. For example, to write a useful program, we need to be able to store values in variables, test these values against a condition, or loop through a set of instructions a certain number of times. Some of the basic program constructs include decision making and looping.

Decision Making in programming is similar to decision making in real life. In programming also we face some situations where we want a certain block of code to be executed when some condition is fulfilled. A programming language uses control statements to control the flow of execution of program based on certain conditions. These are used to cause the flow of execution to advance and branch based on changes to the state of a program.

- if
- if-else
- nested-if
- if-else-if
- switch-case
- break, continue

These statements allow you to control the flow of your program's execution based upon conditions known only during run time.

A loop is a programming structure that repeats a sequence of instructions until a specific condition is met. Programmers use loops to cycle through values, add sums of numbers, repeat functions, and many other things. Two of the most common types of loops are the while loop and the for loop.
The different ways of looping in programming languages are

- while
- do-while

- for loop
- Some languages have modified for loops for more convenience eg :- Modified for loop in java.

For and while loop is entry-controlled loops. Do-while is an exit-controlled loop.

Code: -

If statement

```
public class If {
 public static void main(String[] args)
{
  int age=20;
    if(age>18)
       {
        System.out.print("Age is greater than 18");
       }
}
}
```

```
C:\ KOMAL SAPATALE >java If
Age is greater than 18
C:\ KOMAL SAPATALE >
```

Ifelse statement

```
public class IfElse
{
 public static void main(String[] args) {
     int number=13;
     if(number%2==0){
        System.out.println("even number");
       }
      else{
        System.out.println("odd number");
          }
}
}
```

```
C:\ KOMAL SAPATALE >java IfElse
odd number
```

Ifelse if Statement

```
public class IfElseIfExp
{
public static void main(String[] args) {
    int marks=65;
```

```
   if(marks<50){
    System.out.println("fail");
   }
   else if(marks>=50 && marks<60){
     System.out.println("D grade");
   }
   else if(marks>=60 && marks<70){
      System.out.println("C grade");
   }
   else if(marks>=70 && marks<80){
   System.out.println("B grade");
    }
   else if(marks>=80 && marks<90){
    System.out.println("A grade");
   }
   else if(marks>=90 && marks<100){
    System.out.println("A+ grade");
   }
else{
      System.out.println("Invalid!");
    }
 }
}
```

```
C:\ KOMAL SAPATALE >javac IfElseIf.java

C:\ KOMAL SAPATALE >java IfElseIf
C grade
```

LOOPING EXAMPLE

```
public class LoopExample {
    public static void main(String[] args) {
        // Using a while loop
        System.out.println("Using while loop:");
        int i = 1;
        while (i <= 10) {
            System.out.print(i + " ");
            i++;
        }
```

```
System.out.println(); // Print a newline for separation

// Using a do-while
loop
System.out.println("U
sing do-while loop:");
int j = 1;
do {
   Sy
   ste
   m.
   ou
   t.p
   rin
   t(j
   +
   "
   ");
   j+
   +;
} while (j <= 10);
System.out.println(); // Print a newline for separation

// Using a for
loop
System.out.prin
tln("Using for
loop:");for (int
k = 1; k <= 10;
k++) {
   System.out.print(k + " ");
}
System.out.println(); // Print a newline for separation
```

```
    }
}
```

```
C:\KOMAL SAPATALE>javac LoopExample.java

C:\KOMAL SAPATALE>java LoopExample
Using while loop:
1 2 3 4 5 6 7 8 9 10
Using do-while loop:
1 2 3 4 5 6 7 8 9 10
Using for loop:
1 2 3 4 5 6 7 8 9 10
```

Conclusion:

Branching allows you to make decisions in your code by evaluating conditions and executing different code blocks based on the results.It's essential for creating flexible and responsive programs that can adapt to different situations.Common conditional statements include "if," "else if," and "else" statements, as well as switch-case statements.Branching is crucial for implementing logic, error handling, and user interactions in software.