| Experiment No.1 |
| --- |
| Basic Program Constructs like Branching and Looping. |
| Date of Performance: 19/7/24 |
| Date of Submission: 26/07/24 |

**Aim: -** Implementation of fundamental programming constructs like branching 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: -**

import java.util.Scanner;

public class Fibonacci {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.print("Enter the number of n for Fibonacci series: ");

int n = sc.nextInt();

if (n <= 0) {

System.out.println("Number of n must be a positive integer.");

} else {

System.out.println("Fibonacci Series up to " + n + " n:");

Fib(n);

}

}

public static void Fib(int n) {

int a=0,b=1;

for (int i = 1; i <= n; i++) {

System.out.print(a + " ");

int c= a+b;

a=b;

b=c;

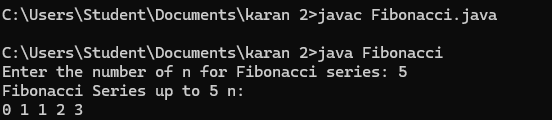
}

System.out.println();

}

}

**Output: -**

****

**Conclusion: -** A while loop checks its condition before executing, ensuring it only runs if true. A do-while loop executes at least once before checking the condition. A for loop is used for a predetermined number of iterations. Break exits the loop immediately, while continue skips the current iteration and proceeds to the next one.