



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

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 a 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



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

- 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 elseif else
```

```
import java.util.Scanner;
```

```
public class Calculator {  
    public static void main(String[] args)  
    {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter the first number ");  
        int a = sc.nextInt();  
        System.out.print("Enter the second number ");  
        int b = sc.nextInt();  
        System.out.print("Enter the operation you want to perform [ + - * / %]");  
        char op = sc.next().charAt(0);  
        solve(a, b, op);  
    }  
    public static int solve(int a, int b, char op)  
    {  
        int ans = 0;  
        if (op == '+')  
        {  
            ans = a + b;  
        }  
        else if (op == '-')  
        {  
            ans = a - b;  
        }  
        else if (op == '*')  
        {
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
ans = a * b;
```

```
}
```

```
else if (op == '%')
```

```
{
```

```
    ans = a % b;
```

```
}
```

```
else if (op == '/')
```

```
{
```

```
    ans = a / b;
```

```
}
```

```
System.out.println("Your answer is " + ans);
```

```
return ans;
```

```
}
```

```
}
```

```
//switch case
```

```
import java.util.*;
```

```
public class Main {
```

```
    public static void main(String[] args)
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int day;
```

```
        System.out.print("Enter the number of day");
```

```
        day = sc.nextInt();
```

```
        String dayString;
```

```
        switch (day) {
```

```
            case 1:
```

```
                dayString = "Monday";
```

```
                break;
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

case 2:

```
dayString = "Tuesday";
```

```
break;
```

case 3:

```
dayString = "Wednesday";
```

```
break;
```

case 4:

```
dayString = "Thursday";
```

```
break;
```

case 5:

```
dayString = "Friday";
```

```
break;
```

case 6:

```
dayString = "Saturday";
```

```
break;
```

case 7:

```
dayString = "Sunday";
```

```
break;
```

default:

```
dayString = "Invalid day";
```

```
}
```

```
System.out.println(dayString);
```

```
}
```

```
}
```

```
//while
```

```
import java.util.*;
```

```
class Main
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
{
public static void countno(int n)
{
    int count =0;
    while(n != 0)
    {
        n = n/10;
        count++;
    }
    System.out.println(count);
}
public static void main(String args [])
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter number:");
    int n = sc.nextInt();
    countno(n);
}
}
```

```
//nested for
import java.util.Scanner;
class Main
{
    public static void squareWall(int s)
    {
        int i,j;
        for(i=1; i<=s; i++)
        {
            for(j=1; j<=s; j++)
            {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

}

```
public static void main(String[] args)
{
    Scanner scn = new Scanner(System.in);
    System.out.print("Enter value of side: ");
    int s = scn.nextInt();
    squareWall(s);

}
}
```

Output-

//if elseif else

```
Enter the first number 4
Enter the second number 3
Enter the operation you want to perform [ + - * / %]-
Your answer is 1
```

```
=== Code Execution Successful ===|
```

//switch case

```
Enter the number of day4
Thursday
```

```
=== Code Execution Successful ===|
```

//while



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
Enter number:1004
```

```
4
```

```
=== Code Execution Successful ===
```

```
//nested for
```

```
Enter value of side: 4
```

```
* * * *
```

```
* * * *
```

```
* * * *
```

```
* * * *
```

```
=== Code Execution Successful ===|
```

Conclusion:

When combined, branching and looping allow for powerful problem-solving capabilities:

- **Complex Logic:** Branching can be used within loops to create complex decision-making processes for each iteration.
- **Dynamic Algorithms:** Many algorithms, such as search and sort, rely on both branching and looping to navigate through data and make decisions efficiently.