

Experiment 2: Programs on Basic programming constructs like branching and looping

Theory :

Branching : We know that Instructions those are written in java Language are Executed in Sequence wise or Step Wise as they are Written in Program. or these are Executed in Sequence Order. Decision Making statements are used when we want to execute the statements as according to the user needs. A User can Change the Sequence of the Statements for Execution. Many Times we Want to Execute the Set of Instructions to be Executed in one Situation and other Statements in other Situations For Executing the Statements in Specific Situation there are Some Decision Making Statements or Decision or Control Statements those are provided by the java Language. In The Decision First a Condition is checked if it is true then it Executes the next Instruction otherwise it Executes another Statements.

1) If else Statements : An if statement tests a particular condition; if the condition evaluates to true, a course-of-action is followed i.e. a statement or set-of-statements is executed. Otherwise (if the condition evaluates to false), the course-of-action is ignored and the statements in else block are performed (if any).

Syntax :

```
if(boolean_expression(ie. condition)) {  
    /* statement(s) will execute if the boolean expression is true */  
} else {  
    /* statement(s) will execute if the boolean expression is false */  
}
```

2) Switch Statements : When number of conditions (multiple conditions) occurs in a problem and it is very difficult to solve such type of complex problem with the help of ladder if statement, then there is need of such type of statement which should have different alternatives or different cases to solve the problem in simple and easy way. For this purpose switch statement is used.

Syntax :

```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}
```

Looping : Loops are used to execute a set of statements repeatedly until a particular condition is satisfied. In Java we have three types of basic loops: for, while and do-while.

1) for loop : Java for loop is used to run a block of code for a certain number of times.

Syntax:

```
for (initialExpression; testExpression; updateExpression) {  
    // code block to be executed  
}
```

Here,

1. The initialExpression initializes and/or declares variables and executes only once.
2. The condition is evaluated. If the condition is true, the body of the for loop is executed.
3. The updateExpression updates the value of initialExpression.
4. The condition is evaluated again. The process continues until the condition is false.

2) while loop : The Java while loop is used to iterate a part of the program again and again. If the number of iteration is not known, then we can use while loop.

Syntax :

```
while (test expression) {  
    // code block to be executed  
}
```

3) do while loop : The do...while loop is similar to while loop. However, the body of do...while loop is executed atleast once before the test expression is checked.

Syntax :

```
do {  
    // code block to be executed  
} while(textExpression)
```

A.

Aim : WAP to check if a character is a vowel or not .

Program :

```
import java.io.*;

public class vowels {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter a character :");
        char c = Character.toLowerCase((char)br.read());
        if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u') System.out.println("Vowel");
        else System.out.println("consonant");
    }
}
```

Output :

```
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> java vowels.java
Enter a character :
I
Vowel
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> java vowels.java
Enter a character :
v
consonant
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> java vowels.java
Enter a character :
u
Vowel
```

B.

Aim : WAP to print a two dimensional table of squares of numbers from 1 to 25 using for loop .

Program :

```
public class twodTableSquare {
    public static void main(String[] args) {
        int i;
        for(i=1;i<=25;i++)
        {
            if(i*i<10) System.out.print(i*i+" ");
            else if(i*i<100) System.out.print(i*i+" ");
            else System.out.print(i*i+" ");
            if(i%5==0) System.out.println("");
        }
    }
}
```

Output :

```
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> java twodTableSquare.java
1  4  9  16  25
36 49 64 81 100
121 144 169 196 225
256 289 324 361 400
441 484 529 576 625
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> □
```

C.

Aim : WAP to find number of and sum of all integers greater than 100 and less than 200 that are divisible by 7 .

Program :

```
public class divisibleBy7 {  
    public static void main(String[] args) {  
  
        for(int i=100;i<=200;i++)  
        {  
            if(i%7==0) System.out.println(i);  
        }  
    }  
}
```

Output :

```
\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> java divisibleBy7.java  
105  
112  
119  
126  
133  
140  
147  
154  
161  
168  
175  
182  
189  
196
```

D.

Aim : WAP to print the following pattern,

```
  *
 * * *
* * * * *
 * * *
  *
```

Program :

```
public class pattern1 {
    public static void main(String[] args) {
        for(int i=1;i<=3;i++)
        {
            for(int j=1;j<=3-i;j++)
            {
                System.out.print(" ");
            }
            for(int j=1;j<=i;j++)
            {
                System.out.print("* ");
            }
            for(int j=i;j>=2;j--)
            {
                System.out.print("* ");
            }
            System.out.println("");
        }
        for(int i=2;i>=1;i--)
        {
            for(int j=1;j<=3-i;j++)
            {
                System.out.print(" ");
            }
            for(int j=1;j<=i;j++)
            {
                System.out.print("* ");
            }
            for(int j=i;j>=2;j--)
            {
                System.out.print("* ");
            }
            System.out.println("");
        }
    }
}
```

Output :

```
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> java pattern1.java
  *
 * * *
* * * * *
 * * *
  *

PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs> 
```