Day-9 Assignment - 30-04-2023

CONTROL FLOW STATEMENTS:

control flow 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.

There are 3 types of control statements in Java:

* Decision Making statements
* loop statements
* Jump(or) Branch statements.

Decision making statements:

Decision making statements execute a pice of code based on some condition.

IF STATEMENT:

* Simple if:

- if statement is used to decide whether a particular block of code will be executed or not based on a certain condition.

- if the condition is true, then the code is executed. otherwise not.

Syntax:

```
if(condition)
  { statement;
  }
```

Example program:

```
package org.controlstatements;
import java.util.Scanner;
public class controlflowstatements {
    public static void main(String[] args
    { Scanner s= new Scanner(System.in);
```

```
System.out.println("Enter your age");
int age = s.nextInt();
if(age >18){
        System.out.println("Eligible for vote");
    }
}
}
```


if-else:.

The if statement is used to execute a block of code based on condition. but if the condition is false and we want to do some other task when the condition is false, how should we do it?

that's where else statement is used. In this, if the condition is true then the code inside the if block is executed otherwise the else block is executed.

Syntax:.

```
if(condition) { truestatement ; }
else { false statement : }
```

ex program:.

```
package org.controlstatements;
import java.util.Scanner;
public class controlflowstatements {
    public static void main (String [] args)
    {Scanner s= new Scanner(System.in);
     System.out.println("Enter your age: ");
     int age = s.nextInt();
     if(age >= 18) {
            System.out.println("Eligible for vote");
     }else { system.out.println("Age criteria
                not met, so not eligible for
        Vote");
    }
}
```

4 else if ladder:-

- if statement if followed by multiple else-if blocks. we can create a decision tree by using these control statements in java.

In which block where the condition is true is executed and rest of the ladder is ignored are not executed.

- if none of the condition is true, the last else block is executed, if present.

Syntax:-

```
if(condition 1)
  { statement 1; }
else if
  { statement 2; }
```

Example:-

```
package org.controlflow.statement;
public class operatorsExample {
  public static void main(String[] args)
    { System.out.println("proofs for vote");
      String proof = "aadhar";
      if(proof = 'voter id')
      { System.out.println("Voter id valid,
        allow to vote");
      }
  else if { proof == "aadhar")
  { System.out.println("Aadhar valid so allow
        for vote"); }
  else if { proof == "ration card")
  { System.out.println("Ration card valid
        so allow for vote");
  } }
}
```

4. Nested - if : (works on logical AND basis)

Nested if refers to an if statement within an if statement, when we write an inner if condition within an outer if condition.

Syntax :

```
if (condition 1)
{ statement 1; // executes when condition is true.
  if (condition 2)
  { statement 2; // executes when condition 2 is true
  } else { statement 2; // executes when condition 2 is false.
  }
}
```

Example program :

```
package org.controlflow.statements;
public class operatorsExample {
 public static void main (String [] args)
 { int age = 54;
   String proof == "aadhar")
 {if(age >= 18) {
                  if (proof == "aadhar")
                  { System.out.println
                  ("both has been checked
                  eligible for vote");
 } } }
```

Switch Statement :

Switch statements are almost similiar to the if-else-if ladder control statements in java. It is a multi branch statement.

- It is a bit easier than if-else-if ladder and also more user-friendly and readable.

The Switch statements have an expression and based on the output of the expression, one or more blocks of code are executed.

-These blocks are called cases. We may also provide a default block of code that can be executed when none of the cases are matched similar to the else block.

Syntax:

```
Switch (expression)
{ case value 1:
      statement1;
      break;
  case value2:
      statemen2;
  break;
  .
  .
  .
case value n:
statement N;
break;
default;
      default statement;
```

Example:

```
package Org.controlstatements;
public class controlflowstatements {

    public static void main (String [ ]
                args)
    { String shit = "m";
      Switch (shit)
      { case "s": System.out.println
                ("Small size shirts
          available here");
          break;
```

```
case "m":
    System.out.println("Medium
    Size Shirts are available");
        break;

case "l": System.out.println("large size
                Shirts are available");
            break;
case "xl" : System.out.println("xtra large size
              shirts available here");
          break;

default : System.out.println("Not available in
                Showroom");
            break;
    }
  }
}
```