**Control statement**

These are the statement which helps in controlling the flow of our code.



**Decision or Branch statement**

* These are statements allows us to create programs where decisions are taken based on expression specified in statement.

**Classified into three types-**

1. if statement
2. if else statement
3. nested if statement
4. switch statement

**If is a statement block Which contains set of code Which is dependent on expression**

1. **if statement**

**Syntax:**

if(condition){

}

**Assignment- WAP to verify that the person age is eligible for DL.**

**Note:** If there is no braces then after if(condition) block only first executable statement is a part of if block after that statement all are independent which are not part ..? Cof if block.

1. **IF …… else**





Assignment: WAP to find largest number between two numbers.

**Multiple if …. else statement (Nested if)**

**Assignment: WAP to find largest, mean and smallest number.**

|  |  |
| --- | --- |
| class Demo{  psvm(String args[]){  int a=10, b=20, c=30;  if(a>b && a>c){  sop(“a is greater”);  }  if(b>a && b>c){  sop(“b is greater”);  }  if(c>a && c>b){  sop(“a is greater”);  }  }  } | class Demo{  psvm(String args[]){  int a=10, b=20, c=30;  if(a>b && a>c){  sop(“a is greater”);  }  else if(b>a && b>c){  sop(“b is greater”);  }  else(c>a && c>b){  sop(“a is greater”);  }  }  } |

**Note:**

1. An if statement can be used without an else statement.
2. Multiple if …. else statement can be used in a program.
3. Once an if …. else statement causes an action in a program, then the remaining if …. else statements will be ignored.
4. **Nested if case statement**
5. **Switch case statement**

The switch case statement is used to select an action from a given set of actions, based on a specified expression.

**Syntax:**

switch(expression/variable)

{

case value1: statement1;

break;

case value1: statement1;

break;

case value1: statement1;

break;

[default: default\_statement;]

}

The expression/variable in the preceding code. Snippet can be any expression depicting a char, byte, short, int or enum variable.

(In JDK 1.7 and 1.8, we can also use string in values)

Assignment: Red(stop), Green(go), Amber(caution)

**Loop statement**

There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially. The first statement in a function is executed first, followed by the second, and so on.

A **loop** statement allows us to execute a statement or group of statements multiple times.

**Classified into three types-**

1. for loop
2. while loop
3. do…. while loop
4. **for loop**

Syntax:

**for(Initialization; expression/Condition; Updating){**

**Body**

**}**

It is a loop which executes a sequence of statements multiple times.

for…. Loop is initialized 1st and then the Boolean expression is checked. If the expression is evaluates to true, then the for block is executed, otherwise the loop terminates. If the for block executed, then the increment or decrement expression is updated to continue the loop.



Assignment: WAP to print 10 numbers.

**Enhanced for loop (For each loop):**

The for …. Each loop is used to iterate over arrays and collection. The for…. each loop has only two parts, unlike the traditional which has three parts.

The two parts of the for…. each loop are variable and array/collection.

**Syntax:**

**for(Declaration of variable : Collection/Array){**

**Body**

**}**

Example

Class ForEachDemo{

public static void main(String args[]){

int marks[] = {70,90,60,30,75};

for(int i:marks){

System.out.println(i);

}

}

}

**While**



Assignment: WAP to print all even numbers between 1-20.

class Demo{

public static void main(String args[]){

int i=2;

while(i<=20){

System.out.println(i);

i+=2;

}

}

}

**do while loop**

do…. while loop works similar to while loop, but the difference is do…. while loop block executes at least once irrespectively whether the expression evaluates to true or false.



**Note:** while loop, is an **entry check loop** and do while, is an **exit check loop**