CONSTRUCTOR

Features of Constructor:

(i) It is a method.

(ii) It has same name as the class.

(iii) It is called automatically whenever an object of that class is created.

(iv) A class may contain more than one Constructor. But the Constructor must be defined in such a way that each one of them is identified uniquely.

(v) A Constructor doesn't have any return type.

(vi) A Constructor supports polymorphism and this type of polymorphism is known as static polymorphism or Constructor overloading.

(vii) It mainly contains initialization statement.

(viii) It is only called once per object of that class.

(ix) We can pass parameters from Class.

(x) When an object is called any one of the valid Constructor is called.

Example 1:

```
Class A{
  int x, y;
  A(){
    x=5;
    y=10;
  }
  Void display(){
      x= x+10;
      y= y-5;
      SOP(x +"  "+y);
  }
}

Class B{
   psvm ( String args[])
   {
      A a1 = new A();
      a1. display();
   }
}
```

output: 15 5

Example 2:

```
Class A{
   int x, y;
   A (int p, int q){
     x=p;
     y=q;
  }
  void display(){
     x= x+10;
     y= y-5;
     Sop(x +"   "+y);
  }
}

Class B{
   psvm (String args[])
   {
     A a1=new A(10,20);
     a1. display();
   }
}
```

output: 20 15

Constructor Overloading:

When a class contains more than one constructor then-

(1) No two constructors will be same.

(2) Number of parameters must be different in different constructors.

(3) If the total number of parameters are same then the data type of the parameters must be different.

```
Class A{
   A(int a){
      Sop(a);
   }
   A(int a, int b){
      Sop(a+"  "+b);
   }
}

Class demo{
    psvm(---){
      A a1=new A(5);
      A a2=new A(10,20);
    }
}
```

Output:

5

10 20

```
Class A{
   A(int a){
      Sop(a);
   }
   A(String a){
      Sop(a);
   }
}

Class demo{
   psvm(---){
      A a1=new A(5);
      A a2=new A("Java");
   }
}
```

Output:

5

Java