# OBJECT

## Creating an object for the class:

```
class_name object_name;          //Create a reference
object_name = new class_name();  //instantiation
```

We can combine above two statement into single statement -

```
class_name object_name = new class_name();
```

## new:

It is important to understand that new allocate memory for an object during run time. The advantage of this approach is that your program can create as many as or as few objects as it needs, during the execution of your program.

## return:

This keyword is used to return the result from the method to the place, from where this method was called. See example2.

Example 1:

```
Class A
{
    int x, y;
    Void get_data(int a, int b)
    {
        x=a;
        y=b;
    }
    Void perform()
    {
        x=x+5;
        y=y-5;
    }
    Void display()
    {
        S.O.P("x= "+x+"y= "+y);
    }
}
```

parameters of the method.

```
Class B
{
    psvm(String [])
    {
        A a1;
        a1=new A();
        a1.getdata(50,60);
        a1.perform();
        a1.display();
    }
}
```

passing arguments to method

```
d:\> javac B.java
        A.class } Byte code
        B.class }
d:\> java B
Output:
        x=55  y=55
```

Example 2:

## Program without return

```
class A
{
   int result;
   void sum(int a, int b)
   {
      result = a+b;
      SOP("The sum= "+result);
   }
}

class B
{
   psvm(String arr[])
   {
     A a1=new A();
     a1.sum(5,10);
   }
}
```

Output:

The Sum= 15

## Program with return

```
class A
{
   int result;
   int sum(int a, int b)
   {
      result = a+b;
      return result;
   }
}

class B
{
   psvm(String arr[])
   {
     A a1 = new A();
     SOP("The sum= "+
          a1.sum(5,10));
   }
}
```

output:

The sum=15