SSSIT Computer Education

Besides R.S Brothers Show Room
Kphb – Hyderabad : 9866144861

Java

```
class Demo
{
   int y=222;    //instance field not static

       void method1()   //non static mtd
       {
              int x=111;  //local variable
              System.out.println("Mtd-1 x : "+x);
              System.out.println("Mtd-1 y : "+y);
       }

       void method2()   //non static method
       {      // System.out.println("Mtd-2 x : "+x);  CE
         System.out.println("Mtd-2 y : "+y); }

       public static void main(String args[ ])  //static mtd
       {
      Demo d=new Demo( );
                         d.method1( );
                                     d.method2( );
              System.out.println("Main y : "+d.y);
       }
}

/*
Note: A variable which is declared in a method, then that variable can be used only
in that method, but a variable which declared as a instance [non static variable] then
it can be
Accessed through out class but we must require an Object reference in static
context   */
```

Instance Methods:

The methods which are define with in class without static modifier.

- Every instance method must be referred by an object reference in static context of the same class
- Every instance method must be referred by an Object reference in static or non static context of outside of the class
- Instance methods can perform the operation on both static and non static variables | fields

These methods are classified into 2 types
1.Mutable Methods
> The methods which are used to change the values of the fields are called mutable methods or setter

2.Immutable Methods
> The methods which doesn't change the value of the fields or
The methods which are used read the values from the fields called immutable methods or getter methods or inspectors

```
class Test
{
   int x,y; //instance field non static

   void setData( )   //non static mtd
   { x=111; y=222;  }

   void getData()
   { System.out.println("x is : "+x);
     System.out.println("y is : "+y); }
```

```
  public static void main(String args[])
    { Test t=new Test( );
          t.setData( );
          t.getData( );
   }
}
```

Eg 2:

```
class Biggest
{
       int x,y; //instance field

       void setData(int a,int b) //a,b are formal parameters
       { x=a; y=b; }

       void findBiggest()
       {   if (x>y)
              {  System.out.println("biggest is : "+x); }
         else
              { System.out.println("biggest is : "+y); }
       }

       public static void main(String args[ ])
       {  Biggest b=new Biggest( );
                       b.setData(120,20);
                                       b.findBiggest( );

       }
}
```

Eg 3:

```
class Student
{
        int m1,m2,m3;  //instance fields

        void setStudent(int a,int b,int c)  //instance mtd
        { m1=a; m2=b; m3=c; }

        void getStudent()
        { System.out.println("M1 : "+m1);
          System.out.println("M2 : "+m2);
          System.out.println("M3 : "+m3); }

        boolean  findResult() //instance mtd
        {
                if(m1>34 && m2>34 && m3>34)
                  { return true;  }
          else
                  {  return false; }
         }

        public static void main(String args[])
        { Student s=new Student( );
                        s.setStudent(80,60,50);
                       s.getStudent( );

           if(s.findResult( )) // here s.findResult() -> returns either true or false
                   System.out.println("Pass");
           else
                   System.out.println("Fail");

                }
}
```