Last lecture user defined type CLASS was discussed.

For that the following class was defined

```
class Point {

        double x,y;

}
```

For this class we had defined a plus operation, which performed addition of the x and y coordinates of 2 points and returned a new point.

```
Point plus (Point b) {

        Point p = new Point();

        /************************************************************

         If we say only Point P it does not mean anything

         When we say new Point() only then a new object  is created.

         New is used only for user defined data types and not needed for others

        new is a unary operator

        *************************************************************/

        p.x = this.x + b.x;

        py. = this.y + b.y;

        /*************************************************************

        Here p.x/y denotes the x/y co-ordinate of the object p

              this.x/y denotes the x/y co-ordinates of the current object using which

              the method plus is called.

              b.x/y de denotes the x/y co-ordinate of the object b given as parameter.

          **************************************************************/

        return p;

        /*****************************************************************

        Returns the resulting point p after adding the x and y co-ordinates of the 2 points

        *****************************************************************/

}
```

To can use this as following

```
public static void main(String args[]){

        Point a = new Point();

        Point b = new Point();

        a.x = 2;

        a.y = 1;

        b.x=1;

        b.y=2;

        Point sum = a.plus(b);

        System.out.println(sum.x+” ”+sum.y);

}
```

We had also discussed about static functions.

Static functions are independent of the object using which they are called.

So the previous function if described as static will be

```
static Point plus(Point a, Point b){

        Point p = new Point();

        p.x = a.x + b.x;

        p.y = a.y + b.y;

        return p;

}
```

To call this we just need to write

```
        Point sum = plus(a,b);
```

Then we discussed about **constructors**.

A constructor is used to create an object of a class. It is defined like a function which has the **name same as that of the class and has no return type**.

A default constructor is the one which has no parameters. If not defined by user, the compiler automatically generates one.

For the class point the compiler generated constructor will be

```
Point () {

}
```

Generally a constructor is used to attach some value to the variables of the class

So a default constructor for Point will be the one that creates a point object as the origin

```
Point () {

        this.x = 0;

        this.y = 0;

}
```

A non default constructor will take 2 values for the x and y coordinates and create an object of point with the following points.

```
Point(double x, double y) {

        this.x = x;

        this.y =y;

}
```

Now if we want to create a point with values 4 and 5 then we write

```
Point p = new Point (4,5);
```

Now we have defined the constructor with parameters as double so the compiler will upgrade int to double .

If we define another constructor with one parameter as int and second double then this will be called instead on the previous as only one upgradation is required.

Also if we define another constructor with one parameter as int and second as float then this will be called in this case as the level of upgradation from int to float is less than int to double.