

Programming in Java
Prof. Debasis Samanta
Department of Computer Science Engineering
Indian Institute of Technology, Kharagpur

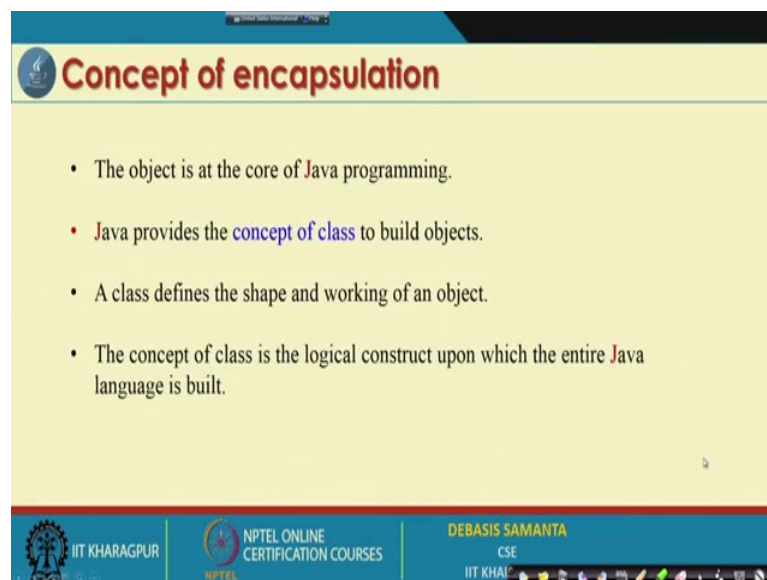
Lecture – 07
Encapsulation

Today we shall discuss about a very important object oriented concept, it is called Encapsulation. So, our today's topic is Encapsulation in Java. As we know the concept of class is a basic things in any object oriented programming. So, we have to write program in Java this means that we have to write a set of classes. All classes then can be used to create objects and this way the object orientation is possible.

So, today we will discuss about the basic concept of class which basically can be achieved with the concept of object oriented principle called encapsulation, and the different elements that is possible in a class.

So, first let us clear the concept of class in Java.

(Refer Slide Time: 01:31)



The slide is titled "Concept of encapsulation" in a red serif font. It contains four bullet points in a black sans-serif font. The footer is a blue bar with white text and logos for IIT Kharagpur, NPTEL, and the speaker's name and department.

- The object is at the core of Java programming.
- Java provides the **concept of class** to build objects.
- A class defines the shape and working of an object.
- The concept of class is the logical construct upon which the entire Java language is built.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE IIT KHALI

As I told you that class is a basic building blocks in any Java program that mean using the class you can write the entire Java programs.

(Refer Slide Time: 01:49)

What is a class?

- A **class** is a group of objects, which have common properties.
- It is a **template** or blueprint from which objects are created.
- It is a logical entity.

A class in Java can contain:

- Fields
- Methods
- Constructors
- Blocks
- Nested class(es) and interface(s)

The diagram on the right shows a vertical capsule shape divided into two main sections. The top section is pink and contains the labels 'Fields', 'Methods', and 'Constructors'. The bottom section is blue and contains the labels 'Blocks', 'Nested Class', and 'Interfaces'.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE IIT KHARAGPUR

So, class basically what exactly a class is? So, a class in fact, is a template; that means, it is basically gives a framework of an object.

Now, how an object can be described? So, in order to describe an object we need many elements for that object. So, the elements which are very important in any objects are called fields, the methods, constructor and sometimes special items called blocks and also class within a class called the nested class. And another important concept also can be included in an object is called the interface.

So, mainly there are 6 different items to be incorporated in a class. Now fields and methods little bit we have familiarity in our last few programs, but other 3 things like constructors and then interface and everything is not known to us at the moment. So, we shall on all this thing slowly. In today we will basically discussed or emphasized on field methods and constructors these 3 things.

(Refer Slide Time: 03:29)

The slide is titled "General structure of a class". It displays a code snippet for a class structure. The code is as follows:

```
class <class-name>{  
    <type> <variable 1>;  
    <type> <variable 2>;  
    <type> <variable 3>;  
    ...  
    <type> <variable n>;  
  
    <type> <method 1>(<parameter-list 1>){  
        // Body of the method 1  
    }  
    <type> <method 2>(<parameter-list 2>){  
        // Body of the method 2  
    }  
    ...  
    <type> <method m>(<parameter-list m>){  
        // Body of the method m  
    }  
}
```

Red curly braces are drawn on the slide to group the variable declarations and the method declarations. A video inset in the bottom right corner shows a man speaking.

The slide footer includes the IIT KHARAGPUR logo, the text "NPTEL ONLINE CERTIFICATION COURSES", and the name "DEBASIS SAMAN" with "CSE" below it.

So, now a basic structure in its simplest form will look like which is shown here. As we see in this fragments of the programs is a basically pseudo program not exactly the program. So, any class can be defined with its own unique name.

So, it is called the class name usually it is given by the user, so user defined name. So here so this is the name of the class that you have you are going to build, so these are name. And then it consists of 2 things which is mentioned here as you see a set of it is called the fields or member elements. All are member elements method also a member elements the particularly it is called fields and other is called the methods.

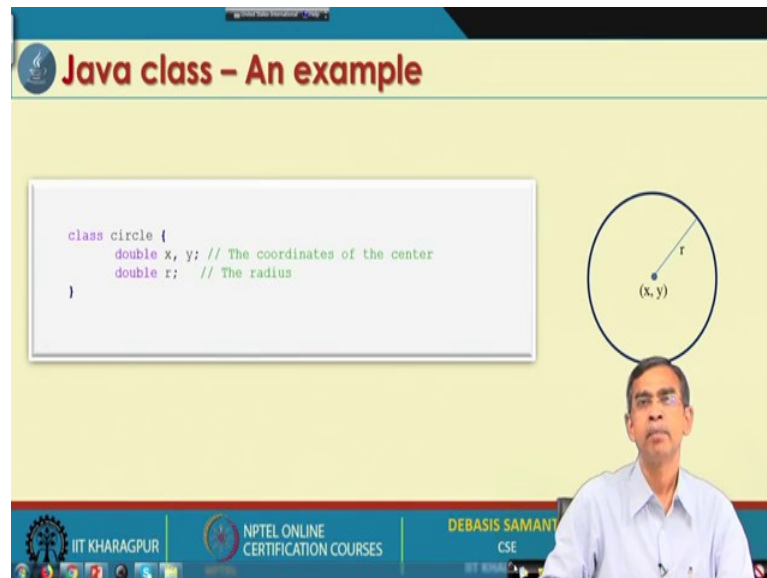
So, fields and methods are 2 very important things in any class. A class can consist of 1 or 0 or more methods and 0 or more fields. There is quite possible that a class does not have any fields also possible that a class does not have any methods, but which is really not useful is a useless thing without any fields are any methods anyway.

So, logically a class have 0 or more member elements fields and methods. So, fields is basically simplest of his form is basically reference variables or some primitive variables that means objects and other name of the other variables that can be define and then methods are basically the operations which are possible to manipulate all the fields variables are there. So, these are basically called data in a class and these are basically called operations in a class. So, these are the 2 things data and operations when they are

punched together it is called encapsulation which you have a little bit learn in our previous lectures.

So, here encapsulation means data and operation are to be put together.

(Refer Slide Time: 05:58)



Java class - An example

```
class circle {  
    double x, y; // The coordinates of the center  
    double r; // The radius  
}
```

The diagram shows a circle with a center point labeled (x, y) and a radius line labeled r .

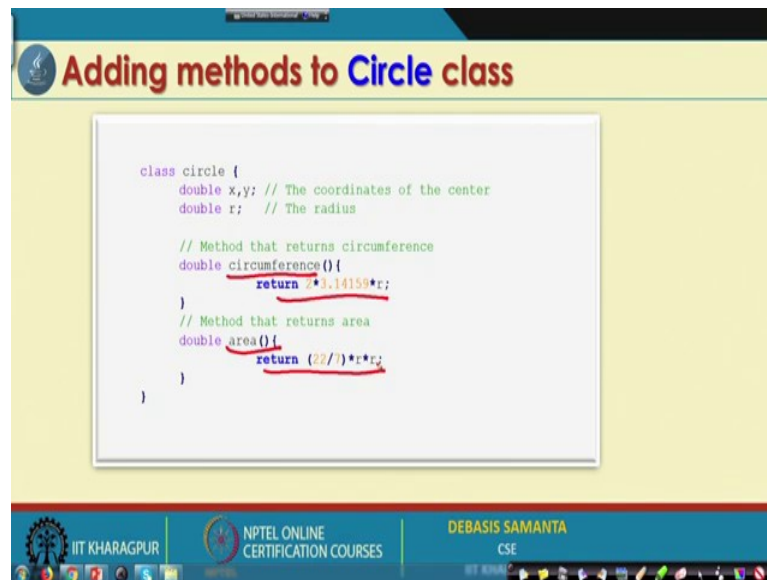
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANT CSE

So, this is a basic structure of the class and let us have a very simple example. Now suppose a circle, a circle is an object. So, there are number of objects all the objects are different, because of its own centre location as well as its radius, the 2 circles are 2 objects, they are having either centre location or radius or both are different.

So, such an object can be defined by its own class. So, here is an example how a class of type circle for you look like this. So, this basically if this is the name of the class a circle in this case the circle has 2 fields, one is declared as float type the radius r and another 2 fields are also declared as float type x and y namely the coordinates of the centre.

So, here as we see a circle is defined in terms of 2 data, centre location x and y and then it is radius. So, this basically completes the definition of a circle. However, in this definition we have not included any method, if we in if we include some methods in it, it will basically makes a sense. Now let us see how the different method that is possible in this definition.

(Refer Slide Time: 07:42)



So, here the circle class that we have discussed just now has been augmented with 2 methods. The name of the methods as we see here one method is called circumference another method is called area. This means these are the 2 operations which can work with if the radius is given to this method. So, they will these 2 methods will be able to calculate the circumference of the circle using this is a formula and area of the circle using the standard formula So, here of course, r is in use; however, other 2 variables mainly x and y is not in use.

So, we can pair some other methods to utilize x, y as well as both all x, y and are there any others details description of this things will take the program little bit lengthy, we just now ignore it. And we will discuss a detail usage of all member elements, all fields in using some other operations or methods. Anyway, so this basically gives one a form of a class called the Circle having 3 different fields x , y ,r, 2 methods namely circumference and area. So, this way we can define the object the class namely Circle here in this case.

(Refer Slide Time: 09:15)

Declaring object of type Circle class

```
// A program that uses the circle class
// Call this file circledemo1.java
class Circle {
    double x,y; // The coordinates of the center
    double r; // The radius

    // Method that returns circumference
    double circumference(){
        return 2*3.14159*r;
    }

    // Method that returns area
    double area(){
        return (22/7)*r*r;
    }
}
```

```
//The following class declare an object of type Circle
class CircleDemo {
    public static void main(String args[]){
        Circle c = new Circle();
        c.x = 0.0;
        c.y = 0.0;
        c.r = 5.0;
        System.out.println("Circumference" + c.circumference());
        System.out.println("Area" + c.area());
    }
}
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMAN CSE IIT KHARAGPUR

Now, let us see what the uses of this class is. Now once you declare a class Circle, we shall be able to use to declare its objects.

So, suppose this is the definition, suppose this is the definition of your class which we have already discussed it and then we can use it to create objects. So, for these things we can create our main class program. So, here is the idea about the main class program we dip. So, this is the class defining the circle category and here is another class we are defining which our program is.

So, here we are defining the program. The name of the program is the name of the program in this case is circle demo 1. So, we have given the name of r 1 and we have already familiar to the main class declaration by virtue of public static void main String args. So, this is the usual statement to declare a class as a main class using the main method, and here you can see we have we have declared one object. We give the name of the object as c and the new is basically the usual operator who is basically create an objects of type circle.

So, this is the standard sentence that you should follow to create an object of type in this case for example, of type class. So, the object namely c is created here, this object c is created, once c is created we can use its different member elements, namely the different fields and then also its method.

Now here we see the c.x basically implies that it is the element x for the object c likewise c.y, c.r and the values for this fields has been initialized as 0.0 for x 0.0 for y 5.0 for r. So, this basically gives an initialization of an objects why are the centre is located 0 0 and having the radius 5. So, this way an object is now created with its value in it. Once the object is created then we can use its method to do certain operation. For example, in this case we use c dot circumference that means, for the object c we call the method circumference; that means, to return the result of its circumference.

So, whenever we call it, it basically use or use this code to calculate the circumference of a circle whose radius is r. In this case it will calculate the circumference of the circle whose radius is 5. Similarly, in the next statement it will calculate area of the same circle c and the result will be printed using System.out.println().

So, this is a one method, this is a one class call the main class who is basically creates an object of type circle and the name of the object is c and for this object we can access the different elements ,fields and methods. Methods are basically operations on the data which is there in that objects. So, this way we can use a create objects. Now with so, in this example, we have created only 1 objects, but it is quite possible to create the multiple objects having a class definition ready.

(Refer Slide Time: 13:41)

Multiple objects declaration

```
// A program that declares two objects of the Circle class
// Call this file CircleDemo2.java
class Circle {
    double x, y;
    double r;

    double circumference() {
        return 2 * 3.14159 * r;
    }

    double area() {
        return (22/7) * r * r;
    }
}

//The following class declares multiple objects of type Circle
class CircleDemo2 {
    public static void main(String args[]) {
        Circle c1 = new Circle();
        Circle c2 = new Circle();
        // Initialize the circles
        c1.x = 3.0;
        c1.y = 4.0;
        c1.r = 5.0;
        c2.x = -4.0;
        c2.y = -8.0;
        c2.r = 10.0;
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
    }
}
```

DEBASIS SAMANT
CSE

IIT KHARAGPUR
NPTEL ONLINE
CERTIFICATION COURSES

So, in this example we can see how we can create more than one objects using the same concept that we have learned just now. So, here if we see c1 and c2 are the 2 objects

created a using this concept new and that these are the 2 objects of type circle. And so, here we see these the initialization of the first circle having centre at 3.0 and 4.0 and radius 5.0 and this is the initialization of the another circle whose centre is at minus 4.0 minus 8.0 and radius is 10. So, the 2 circles are now built, 2 circles having the 2 centers 2 different radius have been built. So, the 2 circles are created.

Now, once the 2 circles are created, we can call the methods in them. So, for example, as we see in these 2 statements `System.out.println()`, we use the circumference method to return the circumference of the circle c1. In this method we see area of the circle c one in another statement we use the same thing, but for the circle c2. So, we just now learn about how the multiple objects in this case 2 objects can be created and all their member elements can be accessed in from the program.

So, this is the concept about creating multiple objects. And now we will discuss about if we can include more than one classes in a program .So, our next example is towards this .So, we will discuss about multiple classes declaration in a program, in a Java application.

(Refer Slide Time: 15:49)

The slide is titled "Multiple classes in a program". It features two diagrams: a circle with center (x, y) and radius r, and a 3D box with dimensions width, height, and depth. Below these are two code snippets. The first is for a Circle class with attributes x, y, and r, and methods circumference() and area(). The second is for a Box class with attributes width, height, and depth, and methods area() and volume(). A video inset of a man is visible in the bottom right corner of the slide.

```
class Circle {
    double x,y;
    double r;
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return ((/)*r*r;
    }
}

class Box{
    double width;
    double height;
    double depth;
    double area(){
        double a;
        a = (width*height + height*depth + width*depth) * 2;
        return a;
    }
    double volume(){
        double v;
        v = width*height*depth;
        return v;
    }
}
```

So, multiple class for example. Already you have learn about the Circle class, how a Circle class can be created and thereby different objects of it can be used. Now, so this is a circle, now suppose another objects say is a quadrilateral it or a rectangle just we want to create it.

Now, so this is suppose example of cuboid and it is basically define with 3 parameters height, width and depth. Now so this is an object like circle right, it has that 3 member elements and definitely it has its own method. Now let us see how we can define a class for this kind of object Quadrif (Refer Time: 16:44) here in this case. Now this is the class definition for this objects and we give the name of this object as a box and as we see the class Box content 3 data and then it has 2 methods; one is called area and another is called volume. So, a by means of area method if the width height and depth is given to us we shall be able to calculates the area, total surface area of the box and given this width height and depth we will be able to calculate its volume, this is method for calculation.

So, these are the 2 methods and the 3 member 3 fields to define these objects. So, now, 2 different classes are defined. Now this after the definition of these 2 classes. Now let us see how we can utilize this class definition to in our program.

(Refer Slide Time: 17:54)

Multiple class objects

```
// Declaring objects of type Circle and Box
class MulticlassDemo {
    public static void main(String args[]){
        Circle c = new Circle();
        Box b = new Box();
        // Initialize the circles
        c.x = 3.0; c.y = 4.0; c.r = 5.0;
        b.width = 3.0; b.height = 4.0; b.depth = 5.0;
        System.out.println("Circumference Circle" + c.circumference());
        System.out.println("Area Circle" + c.area());
        System.out.println("Area of Box" + b.area());
        System.out.println("Volume of Box" + b.volume());
    }
}
// Save this file as MulticlassDemo.java
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMAN CSE

Now, here is the program who is basically use more than one class in the program. So, the name of this class let it be Multiclassdemo we get this is because, it is basically is a program which use the different classes that we have just notify. And here you see the 2 declaration Circle c =new Circle() this basically creation object whose name is c. In the next statement, we create another objects called b of type Box. So, the 2 objects are created, once the 2 objects are created now definitely we have to give the value to this objects because they are created without any specific value in them.

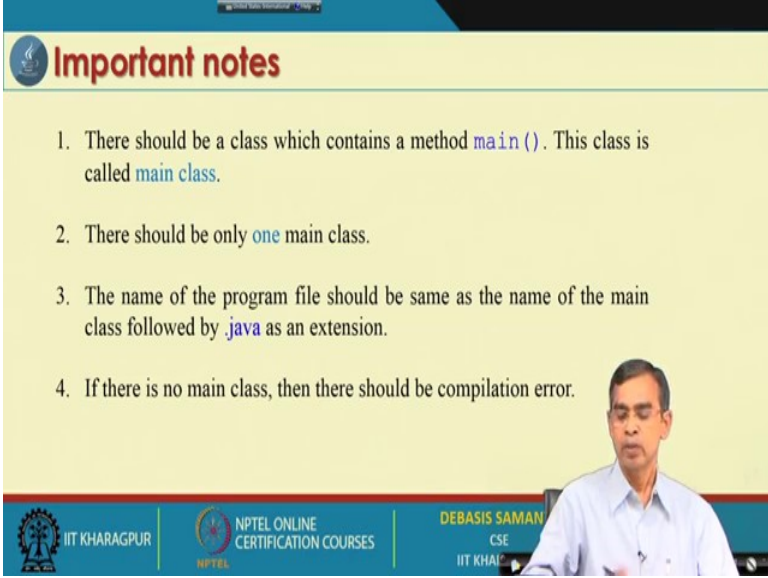
So, we have to initialize objects. So, for the initialization as we see here these are the statement that basically initialize the circle objects for its x, y and r value in. So, you can see c dot x implies that it is a x value for the object c like, likewise the object b is initialized with its 4 different values namely width 3.0 and then height 4.0 and depth 5.0. So, the objects is now initialize with its member elements data and now we can access the operations which are defined for each objects in their corresponding classes.

So, here we can see in this 2 statement, we access the 2 methods which are defined in the class Circle, namely the circumference and area in another statement. We access the method area, but it is defined for the class box and another is volume also define for the class Box. So, the operations in each objects are now accessed once they are objects are created.

And so this way we can utilize the different elements, the different classes which we can define; one thing is that all the classes that we have define we can store in one program, one file in this case, for example, the class Circle and we can store in the program file after this class Circle we can write the program or the quotes for the class Box, they can be placed one by one and then finally, the main class. So, main class is here Multiclass demo can be written and all this thing, 3 classes, the class says for the class definition for Circle class definition for Box. And finally, multiclass demo it is called the main class who is basically creates objects defined by the different classes can be used. And all this things can be stored in a file the name of the file should be same as the name of the main class.

So, in this case MulticlassDemo.java should be the name of the file. So, once all these things put together and you set the file as MulticlassDemo.java, your program is now ready and you can compile it and then run it. And then once you run the different result as per the main method will be executed and you will be able to get the output.

(Refer Slide Time: 21:46)



Important notes

1. There should be a class which contains a method `main()`. This class is called **main class**.
2. There should be only **one** main class.
3. The name of the program file should be same as the name of the main class followed by **.java** as an extension.
4. If there is no main class, then there should be compilation error.

DEBASIS SAMAN
CSE
IIT KHARAGPUR

So, this is a concept we have learned about that how we can create the class and how we can create the different member elements namely the data and operations in it and how the different classes can be used to instantiate the create objects. And finally, using those objects how we can solve many problem. In this case we can create a set of class a class Circles or a set of boxes and for a circle and boxes we can create the different geometrical parameters like area volume pay then circumference accepted etcetera.

Now, so here are few things that are you want to highlights it. So, there are 3 classes in the last example that we have considered, out of these 3 classes 1 class is called the main class, which we have told many time. So, main class is because it includes the main methods. So, if you want to run a program successfully, then what about the classes are there they should be there and there should be one class called the main class which includes the main method. And of course, in one program only one main class that mean one class with main method should be, you cannot have 2 classes and in both the classes again main method it is not allowable.

So, you cannot do that and the main class method should be safe as the same name of the main class name, but extension should be dot Java which is already there. Now if you have any program without any main class. That means, there is no main method in any class then this program cannot be compiled.

So, execution is also not possible, if it is not compiled execution is no I am in question, so that is not possible. So, you should have one program one file; the main class should be there this means that should include the main method.

(Refer Slide Time: 23:42)

Java program without main class

```
class Circle {  
    double x, y;  
    double r;  
    double circumference() {  
        return 2 * 3.14159 * r;  
    }  
    double area() {  
        return (22/7) * r * r;  
    }  
}  
  
class Box {  
    double width;  
    double height;  
    double depth;  
    double area() {  
        double a;  
        a = (width * height + height * depth + width * depth) * 2;  
        return a;  
    }  
    double volume() {  
        double v;  
        v = width * height * depth;  
        return v;  
    }  
}
```

Name the file as **Test.java**.
This program reports compilation error as follows.

Error: Main method not found in class Circle, please define the main method as:
public static void main(String[] args)
or a JavaFX application class must extend
javafx.application.Application

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE IIT KHARAGPUR

Now, here is an example say suppose you have included 2 classes Circle and then Box and then save them as you say tem tem dot test.Java. If you want to compile it your compilation error will be the output. So, you will not be able to compile it successfully because in this file test.java, there is no main class. So, this is one important thing that you should consider. For example, if you compile this it will easily give this kind of error on your terminal.

So, in that compilation error; that means, a class file is not created.

(Refer Slide Time: 24:22)

Method with parameters

```
class circle {
    double x,y;
    double r;
    double circumference(){
        return :*1.14159*r;
    }
    double area(){
        return ((r/2)*r;
    }
    void setCircle(double a, double b, double c){
        x = a; // Set center x-coordinate
        y = b; // Set center y-coordinate
        z = c; // Set radius
    }
}
```

```
class CircleDemo3 {
    public static void main(String args[]){
        Circle c1 = new Circle();
        Circle c2 = new Circle();
        // Initialize the circles
        c1.setCircle(3,4,5,0);
        c2.setCircle(4,5,6,0);
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area of circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area of circle 2" + c2.area());
    }
}
```

Name the file as [CircleDemo3.java](#)

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE

Now, so we have discussed about class declaration, the different elements in it; namely data and operations. There are many more operations we can include; in fact, there is no limit. There is no limit on how many data should be put into a class, there is also no limitation how many method should be placed there inside a class but depends on as per your requirement you should include as many data as many methods in the in a class.

Now, let us extend the definition of class circle little bit in a detailed manner. Now whatever the discussion that we have discussed here in this, I just want to include one more method all these things you have already learn previously.

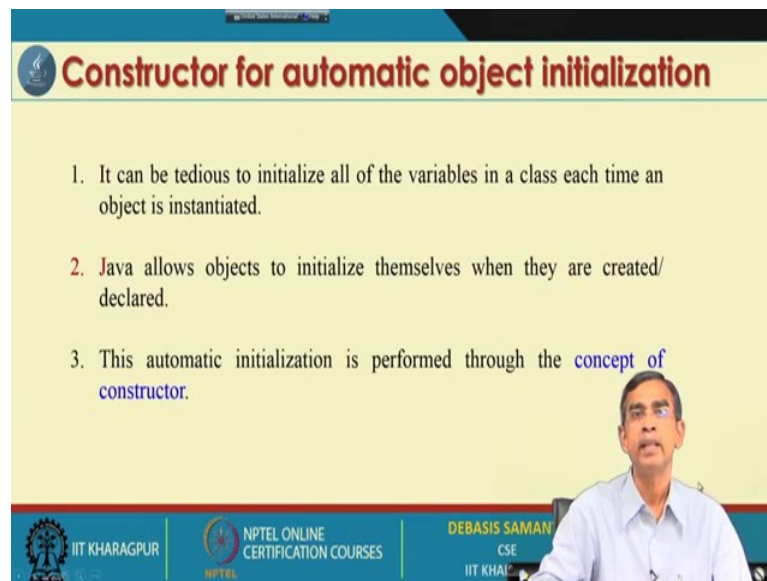
Now, in this class declaration I use one more method in it, the name of the method is set Circle and you see in this method it has 3 parameters. In all the method that we have discussed there is no parameters in them, only they are name of the methods without any arguments, but here we declare 3 parameters namely a, b and c. Now these 3 parameters as you see from the quote this basically the value a will be assigned to x that mean x, y and z will be initialized by these 2 3 variables.

So, setcircle is in fact, is basically to initialize the object initialize an object circle. And here is an example so this example if you see these the main class, the name of the main class is the CircleDemo3 and this create 2 objects c1 and c2 and using the setCircle we can initialize this c1 and c2 objects as the x y 3.0, 4.0 and radius 5 point zero for the first

and minus 4.0, 8.0 and 10.0 for the second and all these things are as usual. Now you can see, the setCircle method is basically initialization attempt.

Now, this initialization in Java is possible in a more pragmatic way. We will discuss about the different way this object can be initialized. The concept is called constructors. So, we will quickly discuss about the concept of constructor now.

(Refer Slide Time: 27:01)



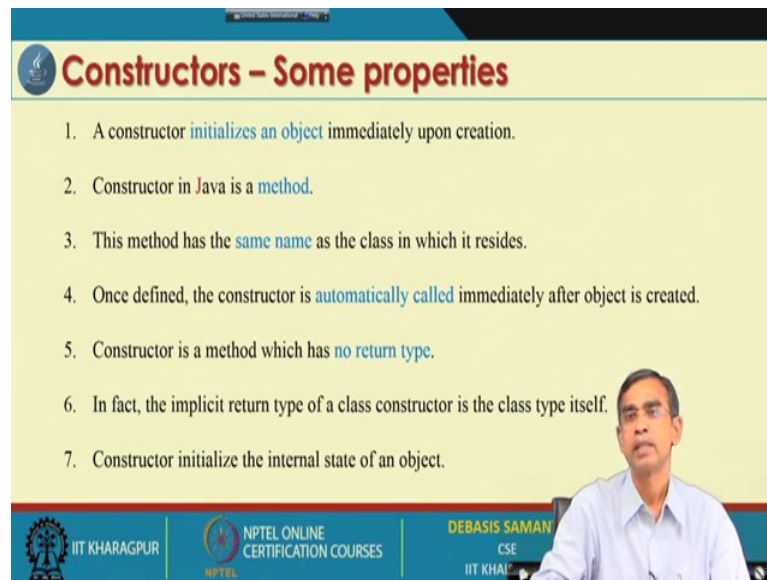
Constructor for automatic object initialization

1. It can be tedious to initialize all of the variables in a class each time an object is instantiated.
2. Java allows objects to initialize themselves when they are created/declared.
3. This automatic initialization is performed through the concept of constructor.

The slide is part of an NPTEL lecture. It features a title bar with the NPTEL logo and the title 'Constructor for automatic object initialization'. Below the title is a list of three points explaining the concept of constructors. In the bottom right corner, there is a video inset showing a male lecturer, Debasis Samant, speaking. The bottom of the slide contains logos for IIT Kharagpur, NPTEL Online Certification Courses, and the lecturer's name and affiliation.

So, constructor is basically the concept of automatic initialization and the idea is that whenever an object is created this method will be called automatically. So, it is a way no need to call it explicitly. For example, the method circumference the area we have to call it explicitly for an object, but this method whenever object is created automatically it will be executed.

(Refer Slide Time: 27:22)



Constructors – Some properties

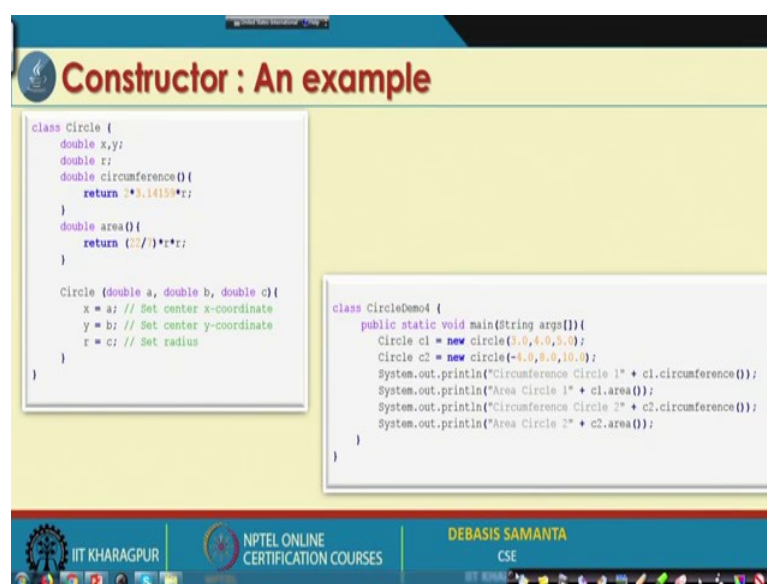
1. A constructor **initializes an object** immediately upon creation.
2. Constructor in Java is a **method**.
3. This method has the **same name** as the class in which it resides.
4. Once defined, the constructor is **automatically called** immediately after object is created.
5. Constructor is a method which has **no return type**.
6. In fact, the implicit return type of a class constructor is the class type itself.
7. Constructor initialize the internal state of an object.

DEBASIS SAMANTA
CSE
IIT KHARAGPUR

And so, for the constructor is concerned, it has few specific things to be considered.

So the name of the constructor should be same as the name of the main class method or the class right. If you want to declare a constructor to initialize an object of a class the name of the constructor should be same as the name of the class and it should not have any return type. And constructor is basically used to initialize the internal member elements in it.

(Refer Slide Time: 27:56)



Constructor : An example

```
class Circle {  
    double x,y;  
    double r;  
    double circumference(){  
        return 2*3.14159*r;  
    }  
    double area(){  
        return (22/7)*r*r;  
    }  
    Circle (double a, double b, double c){  
        x = a; // Set center x-coordinate  
        y = b; // Set center y-coordinate  
        r = c; // Set radius  
    }  
}
```

```
class CircleDemo4 {  
    public static void main(String args[]){  
        Circle c1 = new circle(3.0,4.0,5.0);  
        Circle c2 = new circle(-4.0,8.0,10.0);  
        System.out.println("Circumference Circle 1" + c1.circumference());  
        System.out.println("Area Circle 1" + c1.area());  
        System.out.println("Circumference Circle 2" + c2.circumference());  
        System.out.println("Area Circle 2" + c2.area());  
    }  
}
```

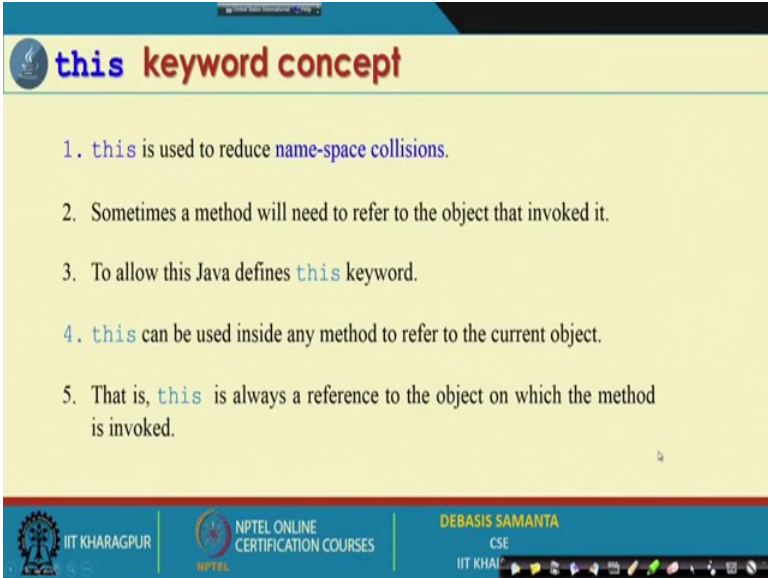
DEBASIS SAMANTA
CSE
IIT KHARAGPUR

Now, let us have an example and this is a simple example as we see this is a class Circle as we have already created earlier and this is a constructor write it basically similar to the setCircle but the method code is like this one. So, this is basically setCircle we have to call it explicitly but this method whenever it is equal we do not have to call it.

Now, let us see how we can use it in our program and this is a one program who is basically use the constructor and here you can see here. So here we can create an object c1, c2 and when we call it then we call who is the parameters. So, 3.0, 4.0, 5.0 are the parameters which has been used to call it; that means, the circle object is created, we pass the values, the constructor will take this value and pass it to the c1 objects and then c1 will be initialized.

So, this is the initialization and one more thing there is little bit mistakes in the typing, so, all these are the capital C, you should note it anyway. So, these are Circle class for example, here is a capital is a typing mistake anyway. So, the objects are created using the constructor in this case and the rest of the things are the previously discuss concept as we have already have.

(Refer Slide Time: 29:25)



this keyword concept

1. `this` is used to reduce name-space collisions.
2. Sometimes a method will need to refer to the object that invoked it.
3. To allow this Java defines `this` keyword.
4. `this` can be used inside any method to refer to the current object.
5. That is, `this` is always a reference to the object on which the method is invoked.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE IIT KHARAGPUR

Now, so the constructor that we have created and now there is a special use of particular keyword is called this keyword. We just want to use it this is a special keyword to resolve certain name collision.

(Refer Slide Time: 29:33)

Constructor : An example

```
class Circle {
    double x,y;
    double r;
    double circumference() {
        return 2*3.14159*r;
    }
    double area() {
        return (3.14159)*r*r;
    }
    void setCircle(double a, double b, double c) {
        x = a; // Set center x-coordinate
        y = b; // Set center y-coordinate
        r = c; // Set radius
    }
}
```

```
class Circle {
    double x,y;
    double r;
    double circumference() {
        return 2*3.14159*r;
    }
    double area() {
        return (3.14159)*r*r;
    }
    Circle(double x, double y, double r) {
        this.x = x; // Set center x-coordinate
        this.y = y; // Set center y-coordinate
        this.r = r; // Set radius
    }
}
```

```
class CircleDemo5 {
    public static void main(String args[]) {
        Circle c1 = new Circle();
        c1.setCircle(1.5, 4.5, 5.9);
        Circle c2 = new Circle(-4.5, 6.5, 10.9);
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
    }
}
```

DEBASIS SAMANTI
CSE

I can discuss about an example, so that I can discuss it. So, in this case a this is the simple things that we have discuss setCircle and here is the constructor that we have already discussed earlier. And suppose this constructor is defined, who is the parameter like this one. Now this means that the x value will go to this x value, but this x and this x should be should not be the same actually.

So, the in order to make it distinguish, so what we can this that we can use this keyword. So, this dot it is means that this x, here this dot y means that this y and this dot r means that this r. So, by means of using this we can specific specifically mentioned that this is the member elements belong to this class itself.

So, this means it is belong to this class the current class. So, this way we can use it.

(Refer Slide Time: 30:27)

Multiple constructors

1. Sometimes, it is necessary to initialize an object in a number of ways.
2. Java allows his using the concept of **constructor overloading**.
3. In other words, **Java** allows to declare one or more constructor method with **different lists of parameters** and **different method definition**.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE

Now, so then rest of things are the same and the concept is called multiple construct. So, we have discussed about the constructor concept, which basically used to automatically initialize an objects. Now we will discuss that in a class even you can define more than one constructor. So, more than one constructor is basically helps the Java programmer to initialize an objects in a multiple ways. This concept is called the multiple constructor or called the constructor overloading.

(Refer Slide Time: 30:58)

Constructor overloading : An example

```
class Circle {
    double x,y;
    double r;
    Circle (double x, double y, double r){
        this.x = x; this.y = y; this.r = r;
    }
    Circle (double r){
        x = 0; y=0; this.r = r;
    }
    Circle (Circle c){
        x = c.x; y = c.y; r = c.r;
    }
    Circle () {
        x = 0.0; y = 0.0; r = 1.0;
    }
    double circumference(){
        return 2*3.14159*r;
    }
    double area(){
        return ((r/2)*r)*3.14159;
    }
}
```

```
class CircleDemo6 {
    public static void main(String args[]){
        Circle c1 = new Circle(1.0,4.0,5.0);
        Circle c2 = new Circle(5.0);
        Circle c3 = new Circle(c1);
        Circle c4 = new Circle();
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
        System.out.println("Circumference Circle 3" + c3.circumference());
        System.out.println("Area Circle 3" + c3.area());
        System.out.println("Circumference Circle 4" + c4.circumference());
        System.out.println("Area Circle 4" + c4.area());
    }
}
```

Name the file as **CircleDemo6.java**

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE

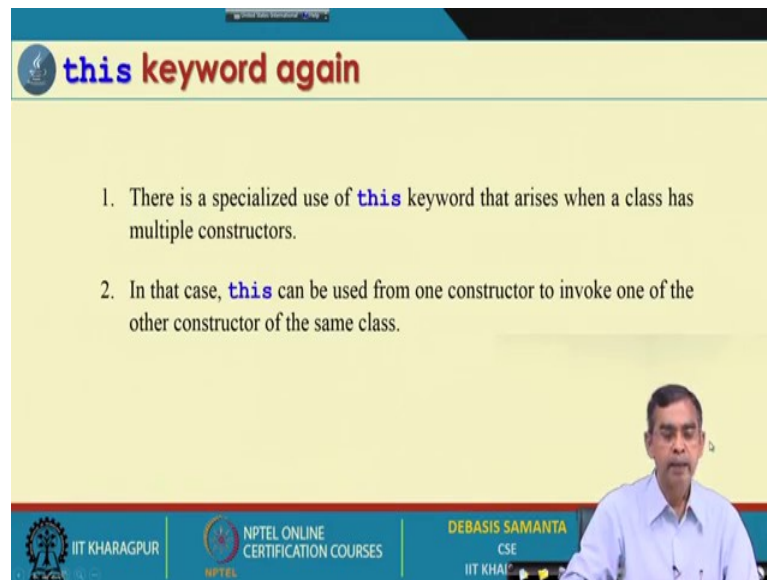
Now, let us have a quick example about the constructor overloading concept. In this let us look at this program. Here we can see this is a one constructor we have defined; this is a one constructor we have defined. So, constructor number 1 and this constructor has the 3 input as a parameter and initialize the elements in the object like this way and in this constructor we only pass the radius r, but others r default value are 0 0. So, this is the second way of constructor and here we initialize in terms of another objects c. So, this is a way of constructor 3 and here it is also default constructor without any argument v.

So, this means that an object can be created without passing any value, with only one value who is passing an object of types Circle itself and sometimes passing 3 different value. So, these are the 3 different way; 4 different way rather the way an object can be initialized. Now here is an example you can see, how you use multiple constructor to initialize the objects in different way.

So, this is the initialization of objects passing 3 values to it. Their initialization using passing only one as a radius and these basically if c1 is known this one; so this mean c3 and c1 are basically the 2 objects having the different or same member elements called x, y and r and here is the c4 is a another object initialize creation without with a default value default constructor; that means, all values are 0 0 and radius is 1 in this case.

Now, we can see how using the constructor or the overloading constructor concept, we can initialize the object in a different way. So, this concept is called the constructor overloading and then all the different object can be accessed their methods can be accessed as usual in the previous example that we have discussed about it. So, this is a concept of constructor overloading.

(Refer Slide Time: 33:11)



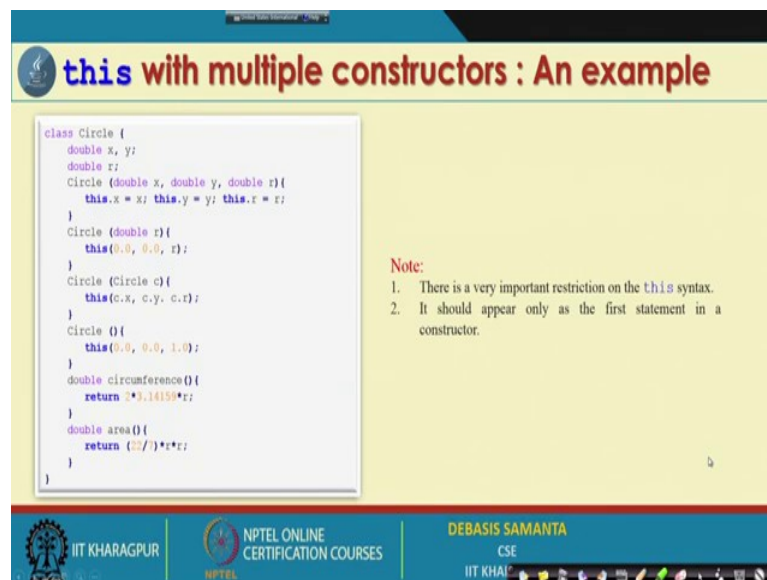
this keyword again

1. There is a specialized use of **this** keyword that arises when a class has multiple constructors.
2. In that case, **this** can be used from one constructor to invoke one of the other constructor of the same class.

DEBASIS SAMANTA
CSE
IIT KHARAGPUR

And there is again one important use of the keyword this. I just want to discuss this with an example.

(Refer Slide Time: 33:18)



this with multiple constructors : An example

```
class Circle {  
    double x, y;  
    double r;  
    Circle (double x, double y, double r){  
        this.x = x; this.y = y; this.r = r;  
    }  
    Circle (double r){  
        this(0.0, 0.0, r);  
    }  
    Circle (Circle c){  
        this(c.x, c.y, c.r);  
    }  
    Circle () {  
        this(0.0, 0.0, 1.0);  
    }  
    double circumference(){  
        return 2*3.14159*r;  
    }  
    double area(){  
        return (3.14159)*r*r;  
    }  
}
```

Note:

1. There is a very important restriction on the **this** syntax.
2. It should appear only as the first statement in a constructor.

DEBASIS SAMANTA
CSE
IIT KHARAGPUR

So, you can see this example here. So, this constructor can be used to reference the method itself as I told you. For example, this is the constructor Circle that we have discussed Circle and we pass this x and y. We have already learned about that this dot x equals to x and everything like that.

Now, there are few more use of these which is highlighted here illustrated here, I will discuss about. Now here if you see this and then the input parameter that mean this in this case the constructor of this class. So, here basically constructor of discussing the first constructor in this constructor; and we call this constructor with the value this 0 0 and r but it takes only input as r as a constructor. So, this constructor Circle will called which in turn, call this constructor to initialize it in terms of 3 different values.

Similarly, this is another constructor which use again this operator here these. And then it basically, takes the value from this Circle c and initialized this as this one and this is another example by is a default constructor without passing any parameter this one.

So, this is another example of this. So, this is basically use for the 2 purpose to resolve the name space; that means, in the same name or variable if it is used then by using this dot the we can, so resolve that who which name basically it implies. So, this is the idea about the constructor and multiple constructor rather.

(Refer Slide Time: 34:58)

? Questions to think...

- How Java developer makes the language simple compared to other language?
- Are there any specific things in Java, which a programmer should aware about those?

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | DEBASIS SAMANTA CSE IIT KHARAGPUR

Obviously there are few more questions that will be discuss in our next class. So, we have mention earlier that Java is most simple language. So, why it is simple compared to other languages like and then another thing is that there are many detail things is things need to be learned prior to experience our good I mean a programming. So, we will discuss about the specific some few important things in Java, which needs to be learned very carefully. So, in our next lecture we should plan it.

Thank you very much.