FUNDAMENTALS OF OBJECT ORIENTED PROGRAMMING

Lecture08

Lecture 8: Constructors in C++ - Default and Parameterized

Welcome to lecture 8 Classes and Objects. So we are going to see C++ constructor. So constructor is nothing but a special method or a member function. So this will be automatically invoked when you are creating an object. So assume that.

Suppose you have a class and then you have an object. Let us say student S1. So when I am creating an object, assume that we are going to see how we are going to write this special member function, what is the meaning of this special member function. So here when I create this object or instantiate the object, so the constructor will be automatically called. We will see how we are going to check this.

So also whenever you want to initialize, so when you want to initialize something on the data members, for example I want to give a is equal to 10. Or string is equal to IIT Roorkee. So something like this, right? So in that case, so when I want to initialize the data members, correct? So constructors is being used.

So talking about the constructor, if you look at the name of the class. See for example here student is the name of the class. So the name of the constructor will also be same like class or even a structure. In the procedural oriented programming you might have studied structure. So when you want to create or when you want to have the constructor so it should have a same name.

So the syntax go like this. So you have name of the class and then we will have a list of parameters. It looks like a function. Right? So that is why it is called as a special member function.

Right? That is the reason we are calling this as a special member function. Another one is you will have, in fact, when I am talking about the function, you can have a list of parameters. Correct? Also, so here you do not have any definition.

The first syntax does not have any constructor definition. Whereas in the second syntax, if you look, you have name of the class and then list of parameters begin and then you have constructor definitions. So you can have, we are going to see two types. So we will see what are those. So in fact, so when I talk about this syntax, so you have name of the class, scope resolution operator, again the name of the class which will be like name of the constructor.

One is name of the class, another one is the name of the constructor and then you will have the open braces like a function. You will have a list of parameters and then we will have constructor definition. So this is how we are going to define the constructor. So there are two types of constructors in C++. One is the default constructor, right?

Another one is the parametrized constructor. right, default constructor and the parametrized constructor. So we are going to see what is meant by default constructor and what is meant by the parametrized constructor. So let us see what is the meaning of default constructor. So let us have class employee, right, the name of the class is class employee and here if you look

The name of the class and this is called the constructor. In fact this is the default constructor. Alright. So the name of the class and name of the constructor should be same. And then as usual I mean we all know functions.

Right. So as usual like function you are writing. See out testing default constructor. Some see out statement. Alright.

And then your class is over. Right. In line number 10. So then you are going to main program. So main program, if you look, you are creating two objects, E1 and E2 under employee class, right.

So whenever you are creating the objects, the constructor will be automatically called. So this is what we define. If you look at the two slides back, we define this, right. So constructor is a special member function. which is invoked automatically at the time of object creation.

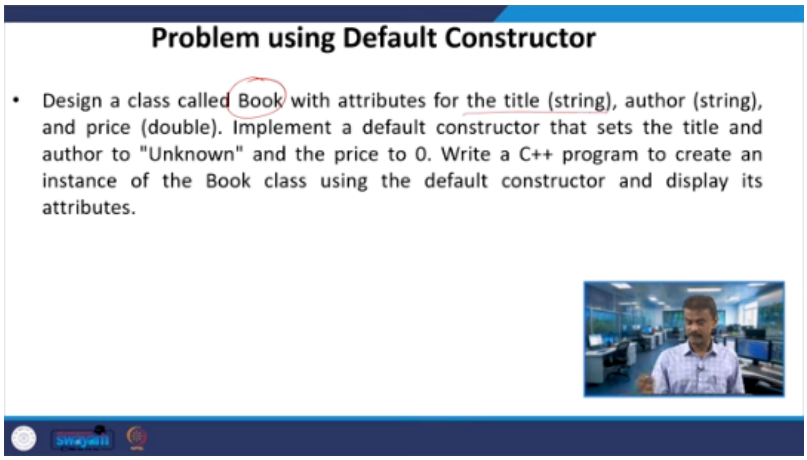
So here I have created objects, right? So first when I line number 13, when I create objects, you have, right, the constructor will be invoked. When I create this, the constructor will be invoked. So that means testing default constructor will be printed first time. And when I create the second object E2, testing default constructor will

Alright. Will be displayed second time. Okay. So you can see the output. So when you run the code you will get the output like this.

So this is what when I am creating an object the constructor will be automatically called. Right. This is how we define the constructor. Right. So this is called the default constructor.

The meaning of default constructor here you do not have any parameters. Alright. So here if you look you do not have any parameters. So when I do not have any parameters, you call this as a default constructor. This is a default constructor.

So now with the help of default constructor, so let us try to solve this problem in object oriented programming. So the problem goes like this. Design a class called book, right. The name of the class is book and then you have the attributes, right, with attributes for the title. Title should be a string.



C++ string and author should be a C++ string and price the double data type. Okay. So now the question is implement a default constructor that sets the title and author to unknown. Right. So title and author initially should be unknown.

You are going to set. Right. With the help of default constructor and the price to be 0. Price is double, should be 0.0, understood. Now, write a C++ program to create an instance of the book class, right?

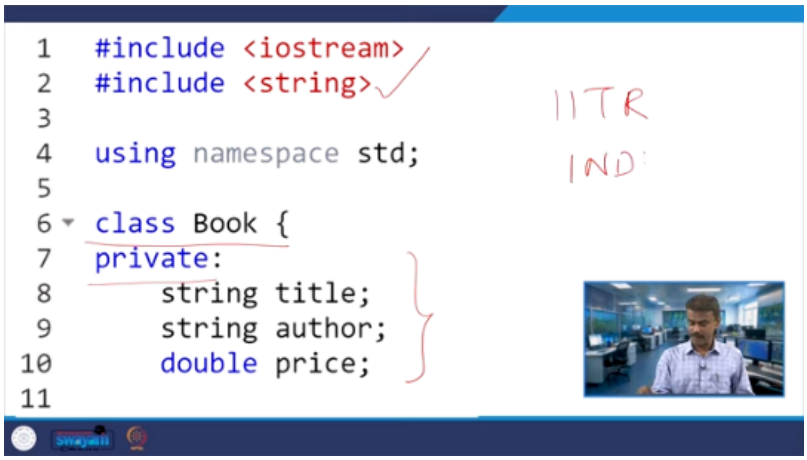
We are going to create the instance of the class book with the help of default constructor and display the attributes. So, whatever the attributes that you have defined in the constructor, that has to be displayed, right? So, this is a simple problem. Let us see how we are going to write the code. So let us define a class book, alright.

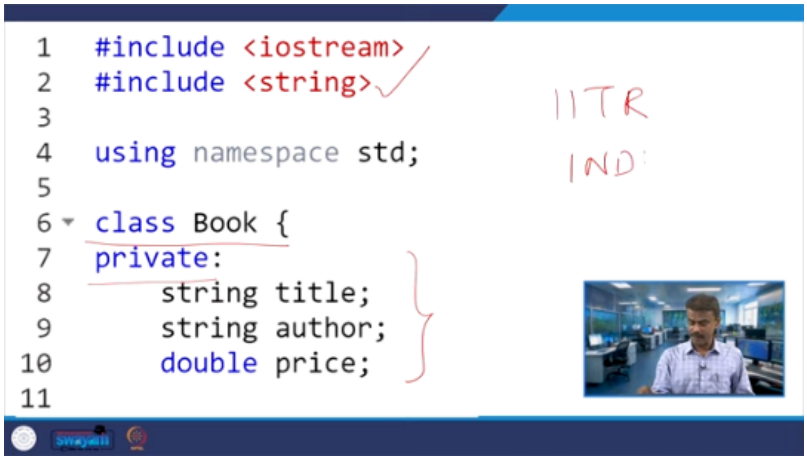
So still we have not seen the access modifier. Right now no need to worry about this keyword. We will define slightly later. So now you have three member data. One is string title.

Second one is string author. Third one is double price, right. Title whose data type is string, right. I have included string. Right.

Once you include this header file, problem solved. Right. You can use the string data type. Right. So string it is like IITR or India.

Right. So this is called as a string. Character array. Right. And then price whose data type is double.





So now under public. Right. Don't worry about the keyword still. We have the constructor. Right.

If you look at the name of the constructor. Which is same as the name of the class. And then I do not have any parameters. So that means this is called the default constraint. Alright.

So this is the syntax where I can directly. Alright. So either you can do this or I can do like this book. That is one way. Alright.

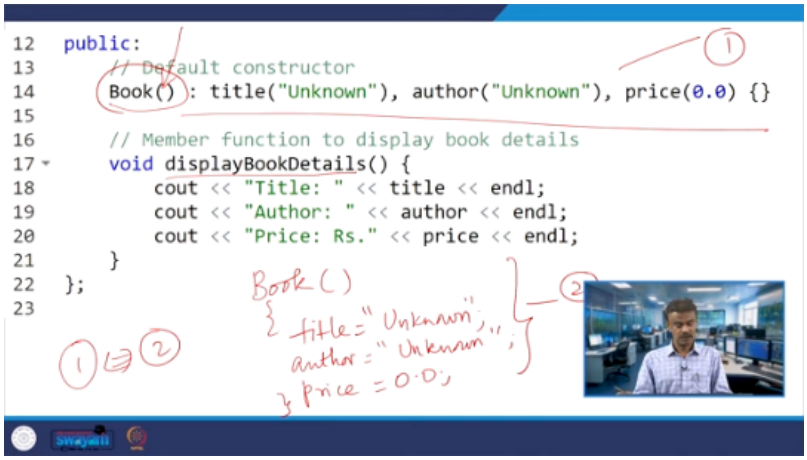
So I have book then followed by the parenthesis. Then I can have title equal to unknown. That is another way. Author is equal to unknown. Right?

Price equal to 0.0. Right? This is a standard way. Whenever you have a function, you write this. Right?

So this is same as I am writing in one line. Right? This is also valid statement. Valid syntax, right? So, the both are same.

So, 1, 2, both are equivalent, right? Both are equivalent. So, either you follow line number 14 or this, okay? You can try, all right? So, book, which is a constructor, title unknown, author unknown and price is 0.0.

Okay, so now we have one more member function called display book details. We have one more member function called display book details. So here you go, see out title, author and price that we are displaying and the class is over. Line number 22, your class is over. So now when I go to the main program, I have created an object my book.



So when I create my book, you know that what will happen? Your default constructor will be called, will be invoked, called or invoked, right? Your default constructor will be called or invoked. So my book is an object, line number 26, underclass book. So this will be invoked, right?

So title will be set. MyBook.title will be unknown. MyBook.author will be unknown. And MyBook.price will be 0.0. So this is what the initialization means.

When we define the constructor, we talked about the attributes or the member data when you want to initialize. So initially I am doing this. Title unknown. Author unknown. And price is initialized to 0.

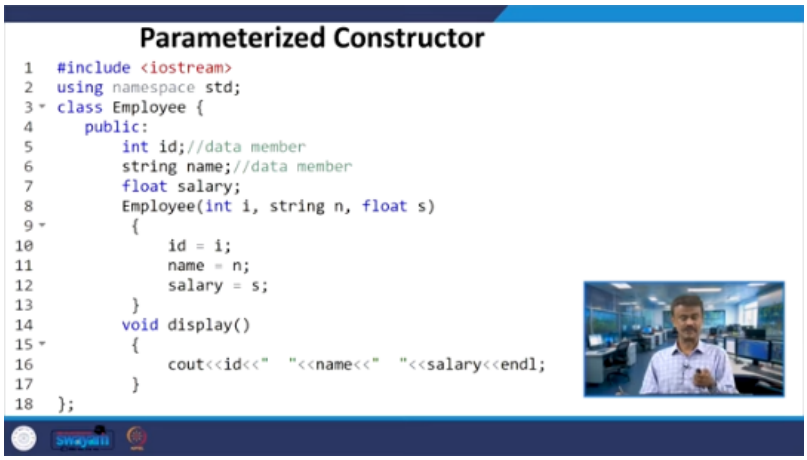
This is the meaning. Price is initialized to 0.0. So now what will happen, your details will be under my book object, my book.title will be unknown, my book.author will be unknown and my book.price is equal to 0.0. That is the meaning. So now I want to display this book details.

So my book will invoke display book details member function. So when it is invoking, so this is a display book detail. Alright, so my book is invoking, so title, author and price you know, so unknown, unknown and 0.0 will be printed, alright, so this will be printed. So you can see when you run this code, you will get the output, so book details line number 29 and then if you look title unknown, author unknown, price rupees 0, okay. So this is the application of the constructor.

So I have initialized. I have defined or declared the variables. You call it as a member data. And then when I want to initialize using constructor, so we have done this line number. Either you can use line number 14 or here I have given the second one.

So you can use any one. And then we have seen the output. So now this is all about default constructor. So the next concept is parametrized constructor. So the default constructor you can see no parameters.

For example we have seen two default constructors. The last example we had seen book. So under book you do not have any parameters. Correct. So now when you look the parametrized constructor.



So let us consider the class employee. Let us consider the class employee. So we know that when I am going to talk about constructor, the name should be same. Where do you find the name should be same? Where do you find name is same?

On line number 8, employee. Then immediately you can find out this is the constructor. So you have int id, member data or data member, string name, float salary. These are all the data members.

And then you have constructor. So this constructor has three parameters, right? This constructor has three parameters. Int i, string in, n and float s, right? i whose data type is integer, n is a string and s is the float, right?

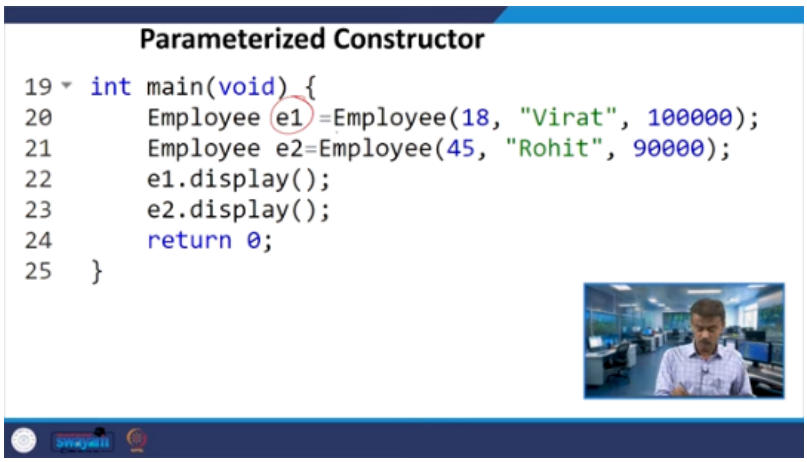
So here I am setting, right? Getter and setter function we have seen. Similarly, i will be initialized to ID because I am using this. N will be name and S will be salary.

Then I will have another member function called void display. This is in fact the special member function. So which is called the constructor and you have

parameter, three parameters. So therefore it is called the parametrized constructor. So here you have void display, right?

Here you have void display. So cout ID Name and salary. Right. So we are printing ID, name and salary.

So now you have int main. I am creating an object E1. Right. And when I am creating an object E1, you can look at the right hand side. You have employee of 18, Virat and 100,000.



Alright, 18 Virat and 100,000. So, this is a special member function, constructor. Alright, so this will be invoked and ID, name and salary, whatever I passed. So, what are the things I passed? 18 Virat and 100,000 we passed.

So, now for E1, the object E1, ID will be 18, name will be Virat and salary will be 100,000. So similarly when you are creating the second object E2 right hand side if you look this is a constructor you are passing 45 Rohit and 90000 right you are passing 45 Rohit and 90000 right. So for E2 the E2 dot id is 45 E2 dot name is Rohit and E2 dot salary is 90000 right so now Here you have E1 dot display, right? So that means the object E1 is invoking display, right?

The object E1 which is invoking display. So when it is invoking display, you can see that we are printing ID, name and salary. So E1 dot ID, right? 18. E1 dot name is Virat and E1 dot salary is 100,000.

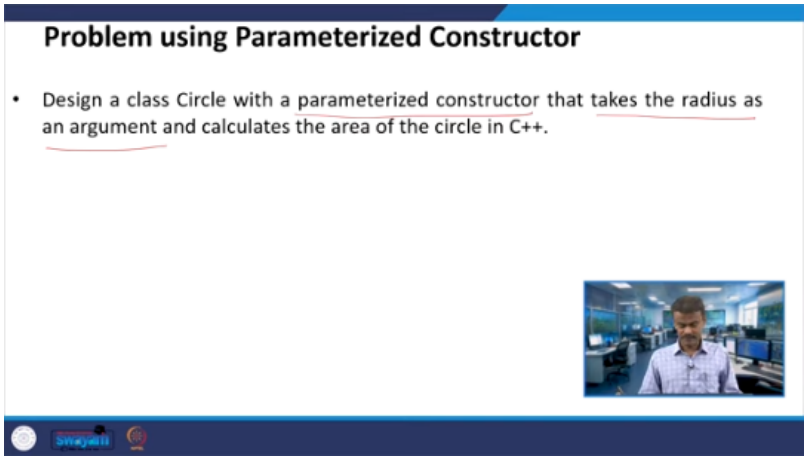
Right. So this will be printed. And similarly when E2 is invoking display E2 dot ID is 45 E2 dot name is Rohit and E2 dot salary is 90,000. Okay. So this will be printed.

So when I run the code I will get the output 18 Virat and 100,000 45 Rohit and 90,000. So the main idea here is I have the class employee, the same name I have in line number 8. So, this I call it as a special member function, right. So, that means this is nothing but a constructor and I have three parameters over here, int i, string n, float s. Therefore, I call this as parametrized constructor, okay.

This is a concept of parametrized constructor. So, the main idea is this is useful to initialize the value. So I, so whatever I am passing, so that will be initialized to ID, name and cell. So now you can manipulate inside, if you are writing another function or you are doing some other program, so we can manipulate. That we will look at it, right.

So this is all about your parametrized constructor. Maybe we can take one example, one problem. Design a class circle with a parametrized constructor that takes the radius as an argument, okay. So you will have radius as argument. So we are going to work it out for the parametrized construct and calculates the area of the circle in C++, right.

This is not a difficult problem but we are going to use the parametrized constructor, right and then how we are going to pass the argument in the parametrized constructor. So that we are going to check here. So here you go. You will have class circle. So we have class circle and then I use one of the member data radius.



So you have radius as the member data. So now I have constructor. So here you have the constructor. So in fact I have one argument R, double R. So therefore it

is called parametric constructor. And the meaning of this is with colon the meaning of this is radius is equal to r right.

You can write other way round circle double r right circle double r open the parenthesis radius equal to r. So whichever you are comfortable you do that. Both the meanings are same. So this is 1 and this is 2. Both 1 and 2 are same. Alright.

So when you want to become the advanced programmer, I mean this is a better syntax. The advanced syntax kind of. So those who are the beginner in OOPS, I mean to say object oriented programming structure, you can use this also initially. Alright. So now we will see another function.

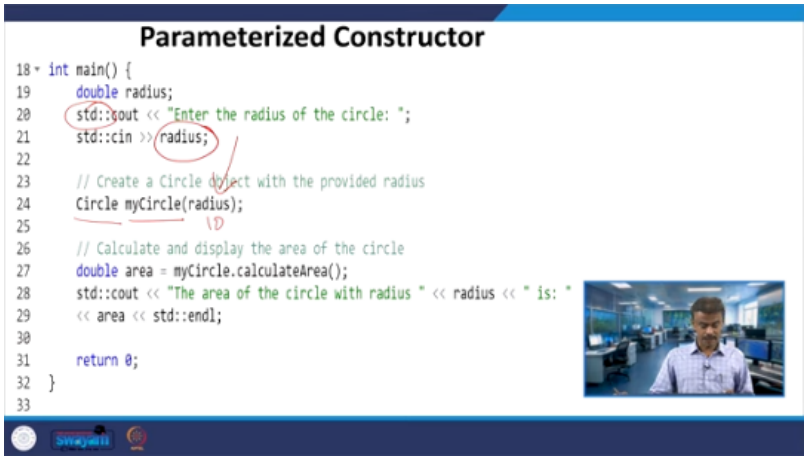
Alright. So now we will see another function. right, which is another member function, double calculate area, right, so this will return pi, this is the value of pi into power of R, that means radius comma 2, so it is nothing but radius into radius, right, the power function that I am calling with the help of C math, right, the power function that I am calling with the help of C math, and this is nothing but pi r square okay pi into radius square right pi r square so the class is over here here you go this is parameter is constructor alright and then I have one member function called calculate area correct and here you go the main program alright in the main program Since I have not used namespace std, using namespace std, I am using std double colon, right, std scope resolution operator.

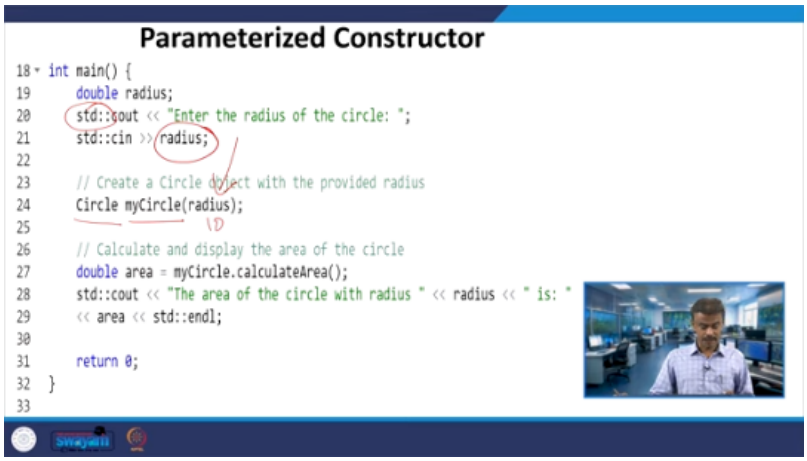
So if you are using namespace std like we had seen in the previous classes or previous program itself, you do not need to use, okay. So enter the radius of the circle. So let us say we are entering the radius, let us say 10, okay. So you are passing the circle, you are creating an object, object is my circle. Alright.

And then you are passing this radius. Whatever radius you have taken the input from the user, you are passing this. So circle, my circle radius. Alright. So assume that.

Suppose I am passing, let us say radius value is 10. I am giving the input 10. Alright. So my circle with 10. Correct.

So what will happen? My circle is an object under circle. So this will invoke the default, right? In fact, this will be invoking parametrized constructor, not a default constructor because you have the argument, right?





You are passing the value, right? So this will invoke the parametrized constructor. So when it is invoking the parametrized constructor, so the radius will be equal to r. So r, what you are passing? We are passing, let us say 10, right? Now radius will be equal to 10.

Correct. Under my circle the radius will be equal to 10. So now I have function. Right. Now this object my circle.

Right. This object my circle will invoke calculate area. Right. So this object my circle will invoke calculate area. Right.

So here you have the area of the circle with radius that will be the print statement. The radius you have passed, let us say I am passing 10, assume that it is 10. So, which is area? So, that means when it is calling this calculated area, right, calculate area member function, right? So, calculate area member function is pi into r star r, right?

So, this is a value of pi, right? So, we will have pi r square. So, r already, right, 10, right? Alright, so that means this is nothing but pi into the radius squared, right, so this is what we are expecting, so here radius will be equal to r, alright, from this meaning radius is equal to r, suppose I pass 10, radius will be equal to 10 now, alright, so pi star 10 into 10, so whatever be the value that will be returned, alright, so when it is returned That will be stored in area.

Alright. So pi star r star r. Whatever be the value that you are giving. So that will be stored in area. So the value will be displayed. So now when we are giving an input.

Let us assume that we are giving r is equal to 10 as an input. Alright. So what will happen? So here my circle. The radius will take the value 10.

Alright. So that means I am passing the value 10. Alright. So here you go, right, the radius, here it is 10 and when you look at the parametrist constructor, the R value will be 10, right? So that means radius is equal to R. So radius will take the value 10, correct?

And then when this my circle, the object is invoking calculate area, right? right so calculate area you have 3.14 right into r into r right the radius into radius correct so that means pi into radius square so pi r square so that will be calculated so when I give 10 as a input I will get this as a output so you can check alright so when you are giving 10 as a input So the area of the circle will be 314.159, right? So this is how the parameterized constructor works, right? So we had seen two examples.

In the previous example, we had passed three parameters and here you are having one parameter, right? So here you go the radius, right? One argument and we know this is a special member function where the name of the class and name of the constructor will be same. That is why it is called a special member function.

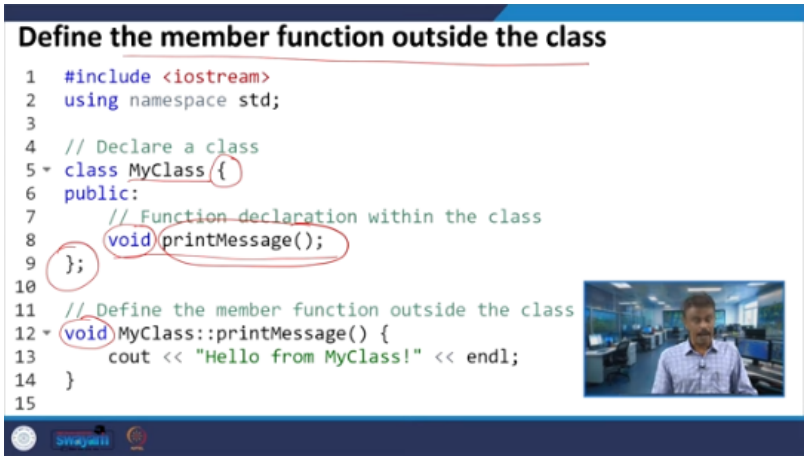
Alright? So we had seen two examples of parameterized constructors. So in one case we had three arguments and in this example we have one argument. So the next program we are going to see is it possible to define the member function outside the class? Right?

The answer is yes. So here you go. I have a class Name of the class is my class. Right?

I have a member function which is print message. So here you can see I have only declared. Right? Here I have only declared print message. Right?

And this is the name of the class. Class is beginning over here and it is ending over here. Right? So now can I define this particular function, member function outside the class? Yes, you can do that.

Right? But you have to follow certain syntax. Right? So here you have the return type void, the same return type. And then you should have name of the class.



Right? So you should have return type. Right? Return type. name of the class right name of the class followed by scope resolution operator two colons when you put it is a scope resolution operator and then you can use the name of the member function right.

So for example print message right member function so whatever be the function. And then you use as usual. Right. So whatever you want to write in this member function. So it goes.

Right. So exactly we are writing. So you have a return type void. And name of the class is my class. And then you have scope resolution operator.

And then you have name of the function. And inside the function whatever you want. Right. You can do that. For example here we have only one cout statement.

Cout allow from my class. Right. So now we will see the main program. Class is over here itself and I am defining outside the class, right. So now if you look at the main program my obj, right which is the object whose class is my class, right my obj under my class.

So this my obj object is invoking print message. So you have a dot operator invoking the print message. So print message will be invoked. So that means you have defined outside the class. So what it is?

You have one see out statement. Hello from my class will be printed. So this we are expecting that as a output. So if you run the code, I will get hello from my class. So the literal meaning over here is, so this is nothing but whatever you want to write inside the class, you are as such writing outside the class.

So, for example usually what will you do void print message right and then you will write cout right allow from my class right. So, this is what you or we have already done right. So, now as such what I want to write I am not writing inside the class. The question is, can I write outside the class? Yes, you can do that.

But we have to follow the syntax or the procedure properly. All right. So when I run this code, I will get the output. So whatever you have written in this print message, all right, in the number function, so that will be printed. All right.

So in today's lecture, we talked about constructor. So whenever you are creating or instantiating an object, the constructor will be automatically called. So one constructor we had seen, default constructor. Another one is parametric constructor. The major difference is in the case of default constructor, you do not have any parameters.

You are not passing any values. All right? In fact, no argument constructor. The second one is with arguments, right? One example we have seen three arguments.

Another example we have seen one argument. So that means you should have more than one argument, right? Or greater than or equal to one argument you should have. So then you call such type of constructor is a parametrized constructor, right? And after this we had seen this program where so is it possible to use the member function outside the class and we had seen this particular

example and yes we can use but we have to follow the syntax that you are seeing in line number 12.

So with this I am concluding this particular lecture. Thank you.