

Object – Oriented System Development using UML. Java and Patterns
Professor. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 01
Introduction

Welcome to this course. In this course, we will discuss about Object Oriented Design using UML, Java and Patterns. My name is Rajib Mall, I will be the instructor for this course.

(Refer Slide Time: 00:33)

About The Instructor

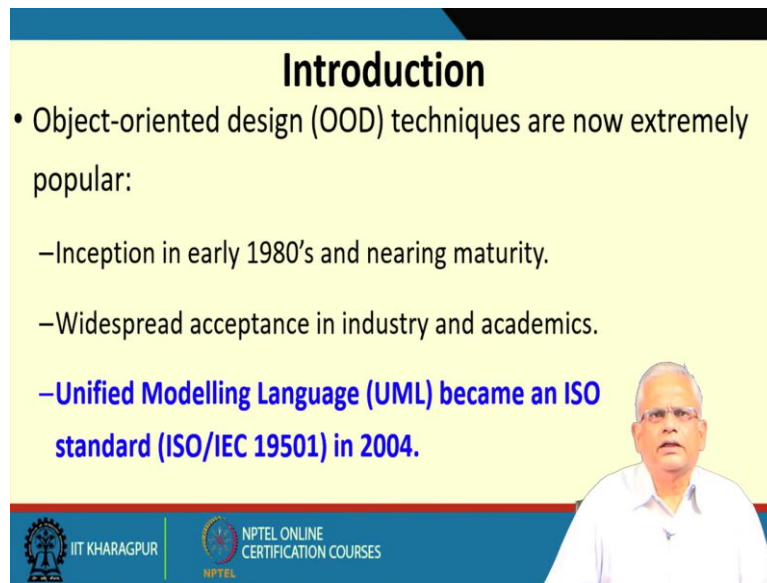
- RAJIB MALL
- B.E. , M.E., Ph.D from Indian Institute of Science, Bangalore
- Worked with Motorola (India)
- Shifted to IIT, Kharagpur in 1994

–Currently Professor

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES


To start with, we need to briefly introduce the instructor myself is Rajib Mall and obtained all my professional degrees, that is Bachelors, Masters and Ph.D. from the Indian Institute of Science, Bangalore. This is the institute, where I studied and then worked for few years with Motorola India and then shifted to IIT, Kharagpur way back in 1994, that is about 26 years back and currently a professor in the CSE department. So, the institute where I work is shown in the above figure.



(Refer Slide Time: 01:22)



Introduction

- Object-oriented design (OOD) techniques are now extremely popular:
 - Inception in early 1980's and nearing maturity.
 - Widespread acceptance in industry and academics.
 - **Unified Modelling Language (UML) became an ISO standard (ISO/IEC 19501) in 2004.**



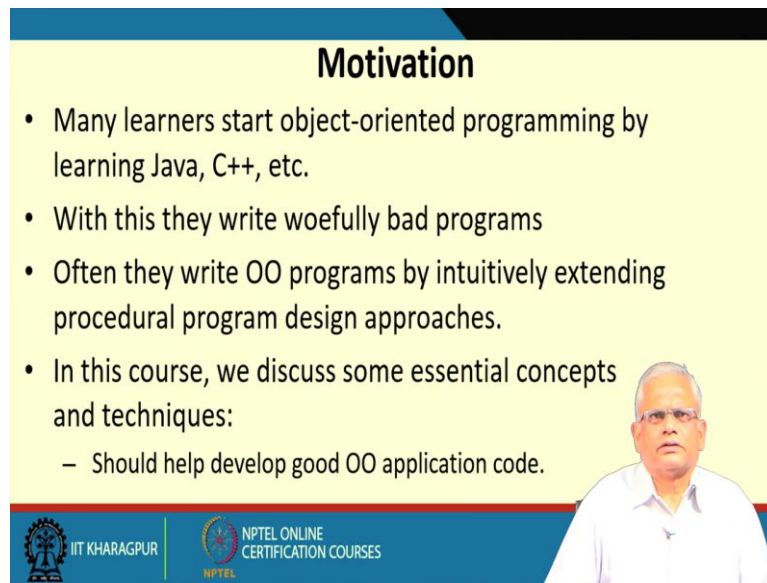
 IIT KHARAGPUR  NPTEL ONLINE CERTIFICATION COURSES

Now let's get started with the course. Object oriented design as you might be knowing it has become extremely popular. Object oriented design started in early 1980 and now it is almost in maturity and in being used in good measure as a matured technology.

There is widespread acceptance in industry and academics. In industry many projects get developed in object oriented, both large projects as well as small and embedded projects. And in academics, object orientation has come to play a crucial role, not only development of programs in the computer science discipline, but also in the other disciplines they do use object oriented techniques in good measure.

To start with this course, we will discuss about the Unified Modeling Language which is the language used in documenting object oriented designs and since this has become an ISO standard with the standard number ISO/IEC 19501, which was accorded in the year 2004. The use of UML has become extensive, in industry as well as in academics. UML is the language for documenting design is almost becoming standard everywhere and in this course, we will also be using UML as the language for documenting object oriented designs.

(Refer Slide Time: 03:37)



Motivation

- Many learners start object-oriented programming by learning Java, C++, etc.
- With this they write woefully bad programs
- Often they write OO programs by intuitively extending procedural program design approaches.
- In this course, we discuss some essential concepts and techniques:
 - Should help develop good OO application code.

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

Portrait of a man in a white shirt.

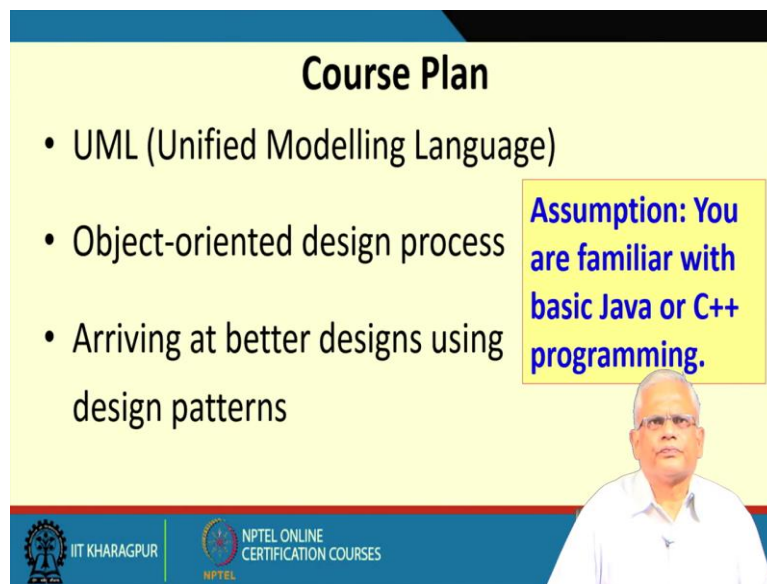
Now in this first lecture, we will form a brief motivation for doing this course. Many of you might have some exposure to Java, C++, etc. Because nowadays, in many engineering colleges Java, C++, etc, which are object oriented languages are taught in the first year or second year of their curriculum.

But the fact is that, they write programs which are not really accepted as object oriented programs, a very poor quality. If we go through this course, we will see that those who have been doing programming, just by reading the Java and C++ books, we will see that there is a big change in their programming, the way they program for a given a problem.

Most of the programmers, who do object oriented programming just by learning Java and C++ do so by only extending their knowledge of procedural program design approaches and as we will see in this course that, that does not do a good job for object oriented software development.

In this course, we will discuss some very essential concepts and techniques which is the objective of this course. At the end of this course, you should be able to develop good object oriented application code. Those who have been doing Java programming, I will encourage them to redo some of their programs they had written earlier, after doing this course and if this course has been done well, then you will see that, there is a big difference in the way you write the program after doing the course and the way you would have written the program earlier.

(Refer Slide Time: 06:17)



The slide is titled "Course Plan" and features a yellow background with a blue header and footer. The main content is a bulleted list of three items: "UML (Unified Modelling Language)", "Object-oriented design process", and "Arriving at better designs using design patterns". To the right of the list is a yellow box with blue text that reads "Assumption: You are familiar with basic Java or C++ programming." Below the list is a portrait of a man with glasses and a white shirt. The footer contains the logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

Course Plan

- UML (Unified Modelling Language)
- Object-oriented design process
- Arriving at better designs using design patterns

Assumption: You are familiar with basic Java or C++ programming.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

In this first lecture, we will just identify the course plan, we will start with the UML, because that is essential language we will use to document designs. While discussing UML, we will also touch upon various basic concepts in object orientation that are necessary to have a good understanding of the object orientated development. After completing the UML part, we will look at the object oriented design process.

The UML is a language, which helps us to document a design, but then how does one come up with the design? That we do by using a design process. We will see a step by step methodology by which we will be able to come up with a solution for a given problem, which will document using you UML and of course during the design process itself, we will make use of the UML to go from one step to another.

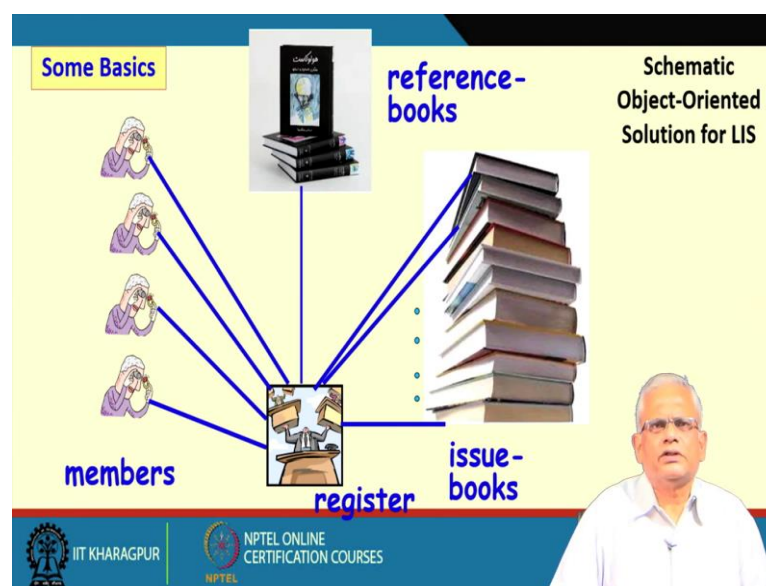
After discussing the object oriented design process, we will attempt to find solution, design solutions to a number of problems, some of the problems, I will work out here and we will give you several exercises and we will encourage you to come up with solutions to those problems.

And also we will discuss about metrics, once you come up with a design, how do you measure the quality of the design? What are the metrics? And after that we will start our discussion about design patterns and here we will first discuss about some very basic principles based on which the design patterns have been founded, we will call it as object oriented principles and then we will discuss about the design patterns and these design

patterns are standard solutions worked out by professionals, these are good solutions and if we know about these solutions, we can use this to improve our designs.

One of the things that we assume here in this course is that you are somewhat familiar with basic Java and C++ programming. We do not want that you are expert in the programming, but then throughout the design process and the design patterns we will give java examples, very small examples, few lines at a time only and if you have some understanding of Java or C++ you will be able to appreciate the code.

(Refer Slide Time: 09:58)



Before we discuss the UML, we will just touch upon some very basic concepts in object orientation, which I hope you did in your object oriented programming using C++ or Java you might already be knowing.

If we try to come up with a program for a library information system. Let us say we have a library of few thousand books and students want to issue the book, the books are of 2 types. The books that can be issued and the books which are only reference cannot be issued and we want a system to be in place, where the students can issue the books, the books will stand in their name, they can return the book, the book will be removed from their name and the issue process will be automatic through a barcode scanner, the books will be issued and returned.

How do we go about designing such a system and write the program? In object orientation, the first thing is to identify, what will be the objects and based on that we identify the classes and then we start writing the class code, but that is for the very preliminary way of writing an

object oriented program. But after completing this course, we will do that better. But then, how does one identify what are the objects here?

If we analyze the library information system, we will find that some of the objects are actually physical entities. For example, some of the books are issue books. Each book is an object and we have an object class, book class from which create the object instances and we have a reference book class and several instances of references books, which are not to be issued out and then there are several members of the library.



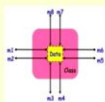
These are all physical entities in the problem description we can identify by analyzing the problem description. But are these all the objects that are needed to solve this problem? Or are there any objects which need to be used, but they are not really physical entities identifiable from the problem descriptions.


Yes, there are some objects which are not really clear from the problem description, we need to synthesize them and one of that is a controller or a register. So, let's just point out here that when you write a program, the classes correspond to objects that are identifiable in the problem domain. But some of the objects are not present in the problem domain, we synthesize them and typical example of these are the controller classes.



As we proceed in this course, it will become clear, how many controller classes to use, under what situation, what are their role, how do they improve the programming and so on.

(Refer Slide Time: 14:13)

Objects

- A system is designed as a set of interacting objects.
- **Objects are often real-world entities:**
 - Examples: an employee, a book etc. 
 - Can also be conceptual objects :
 - Controller, manager, etc. 
- An object consists of data (attributes) and functions (methods) that operate on data. 
- Encapsulation.



Every object oriented solution is designed in the form of a set of interacting objects. We can create classes and we can instantiate objects from the classes and these objects interact with themselves to produce the required behavior of the system. But then, how do we identify the interaction? The way they interact. Before knowing the nitty gritty, let us not worry too much about how do they interact.

Later as we proceed, we will see that how do we design the interaction among the objects. We have already seen that some of the objects are real world entities. They are identifiable by reading the problem description. Examples are an employee, a member, a book, these are entities, identifiable reading the problem description.

But then we also need some conceptual objects, these are not really, we cannot identify such objects in the problem domain, just by reading the problem description. We need to synthesize them and examples of that are controller manager etc, which might be implicit. So, the problems are, the objects are 2 types one correspond to real world entities, the other are conceptual objects, not readily identified in the problem description. But we need them, they play one of the most crucial role in any solution.

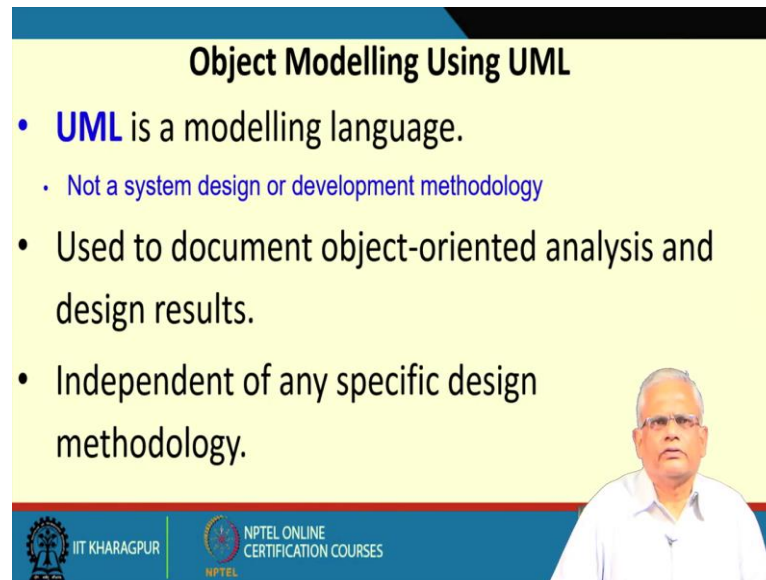
As we proceed in this course, we will see how are these conceptual objects identified and put in place. Every object has its data members, some are public data members and some are private. These are called as the attributes and then every object has its functions, which are called as methods and these methods operate on the data and the data can be operated only through the methods.

An external entity cannot directly access the data of another object. If it wants these data to be changes or to be accessed, it needs to take help of the available methods. The methods have the sole right on operating on the data and this principle is called as encapsulation. The data is encapsulated within the methods. Other objects cannot directly access the data they are somewhat isolated.

The data is isolated from other objects, they can only be operated through the methods and these helps in achieving loose coupling. Encapsulation helps achieving loose coupling among different objects, it is a good programming principle, helps in debugging in development, understanding and so on. So, this is a very important principle in object orientation very fundamental principle and anybody learning object orientation need to have learnt encapsulation.

But here in this course, we are assuming that you already have some knowledge of object orientation. So, just highlighting some of the basic principles without really spending much time on this. The primary focus of this course is, if you know some aspects of object orientation and object oriented programming, how does one go about designing and design well using patterns and we take several Java examples.

(Refer Slide Time: 19:02)



Object Modelling Using UML

- **UML** is a modelling language.
 - Not a system design or development methodology
- Used to document object-oriented analysis and design results.
- Independent of any specific design methodology.

The slide features a portrait of a man in a white shirt in the bottom right corner. At the bottom, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

As I mentioned, to be able to do object oriented design, we need to first learn UML. UML stands for Unified Modeling Language and please note it is a language. Using a language, we can document something. It is not a design methodology, UML by itself does not let you design, if you know the design, if you can come up with the design you can document using UML.

By simply learning UML, you do not become a designer. You need to know the design process to be able to become designer, but then you can document your design using UML. UML is used to document the analysis and design results and one important advantage of the UML is that, it is independent of any specific design methodology.

We will discuss a design processor methodology as part of this course, but then several variations of design processes methodology exist, new design processes and methodology may come up. But you can still use UML, because it makes no assumptions regarding the design methodology or process. So, that is a very important advantage of UML that once you learn UML, whatever be the design methodology be used you can use UML.

(Refer Slide Time: 21:08)

Object Modelling Using UML

Caution: Learning UML can make you no more an expert OO designer, than learning English alphabet and grammar can make you an expert story writer...

For designing, need to learn object-oriented design methodology, patterns...

The slide features a yellow background with green text. At the bottom right, there is a small inset image of a man with glasses and a white shirt. The bottom of the slide has a blue footer with logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

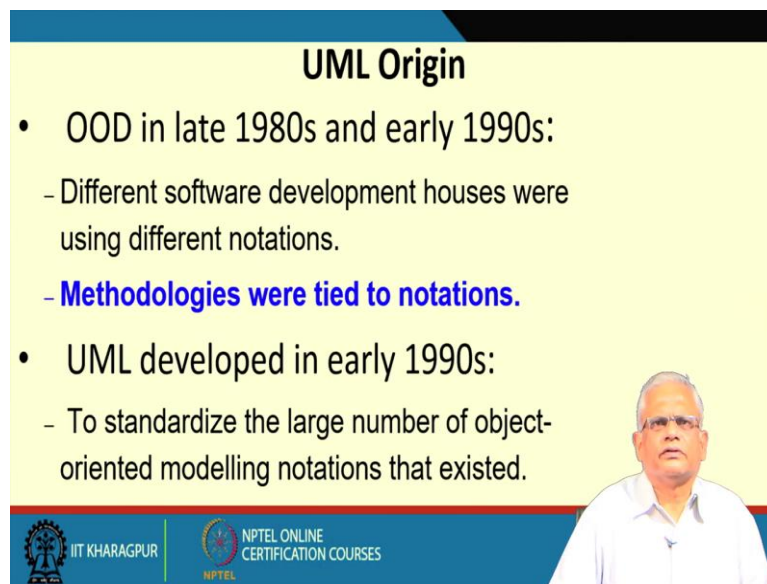
But then, as a caution, please remember that do not expect too much from UML, it is just a language. You must remember, that learning UML can make you no more an expert object oriented designer than learning English alphabet and grammar can make you an expert story writer. A school student, may be in the first grade or second grade, who learns English alphabets and grammar rules can write correct sentences.

But then, becoming a successful story writer is another thing. You need to have a plot, you need to know the story and so on and then only your English language of grammar and alphabets can help to just document your story, knowing the English alphabet and grammar does not make you an expert story writer.

But it helps to express your story, if you know the story. The same thing here that UML if you know well and you come up with a design process you come up with a design by using some design process then you can document your design. So, just knowing UML does not make you an expert designer for coming up with good designs, you need to learn the object oriented design methodology and the design patterns, which are part of this course, we will discuss that after discussing UML.

So, at the end of the course, you should be able to use UML, should be able to use the design process to come up with the design and should be able to improve your design by using the design patterns.

(Refer Slide Time: 23:27)



UML Origin

- OOD in late 1980s and early 1990s:
 - Different software development houses were using different notations.
 - **Methodologies were tied to notations.**
- UML developed in early 1990s:
 - To standardize the large number of object-oriented modelling notations that existed.

The slide features a portrait of a man in a white shirt on the right side. At the bottom, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

Before discussing the nitty gritty of UML, let's see how UML started. The object orientation and the object oriented design all started in a very basic state in 1980s and then progressed in 90s, many design processes were proposed by researchers and the thing is that, these design processes are design methodologies, using which given a problem you can come up with the design solution. Each of them had their own design notations.

For example, if you want to document a class then one design methodology will document a class using some notation, another design methodology may use in another notation. The interaction among classes like the inheritance among classes, the other relations like aggregation composition etc. used different notations.

The methodologies that were being used were many, about a dozen prominent methodologies existed and each one had different notations. This made it very difficult because, even in any small development house or a large development house you go in, you find that in the development projects, different development projects, different methodologies and different notations were used.

This made it very difficult, because if you want to get the solution from another team to reuse their solution, becomes very difficult because they had used different notations and the team that are developing a project were using a different notation and across different companies it became still worse.

Different books gave examples using different notations, it was becoming difficult, as student learning one design methodology found it difficult to get accustomed to another design methodology and the notations. To overcome these problems, there was an attempt for standardization. UML was developed in the early 90s to standardize the large number of object oriented modeling notations that existed and it was attempted to make it independent of the methodologies.

The earlier notations were tied to specific methodologies and that was actually creating the problem. So, one of the objectives behind UML was not only to standardize the notations, but to make it independent of the large numbers of object oriented design techniques that existed. With this basic introduction we will stop and we will continue from this point in the next lecture. Thank you.