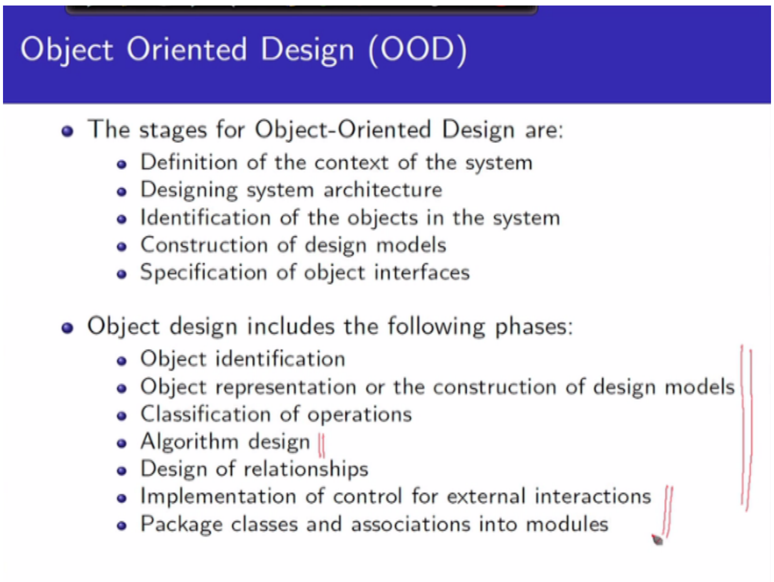
Object-Oriented Analysis and Design
Prof. Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology- Kharagpur

Lecture - 11
Foundations of the Object Model – OOA, OOD and OOP

(Refer Slide Time: 00:24)



Object Oriented Design (OOD)

- The stages for Object-Oriented Design are:
  - Definition of the context of the system
  - Designing system architecture
  - Identification of the objects in the system
  - Construction of design models
  - Specification of object interfaces
- Object design includes the following phases:
  - Object identification
  - Object representation or the construction of design models
  - Classification of operations
  - Algorithm design
  - Design of relationships
  - Implementation of control for external interactions
  - Package classes and associations into modules

Welcome back to module 7 of object-oriented analysis and design. We have introduced a basic foundational concepts of object modelling. We have talked about object oriented analysis and what else required to be done for object oriented design. So I will start here with a quick summary of summary and recapitulation of object oriented design. So the object oriented design comprised typically of a couple of stages, so better that we give here is.

The first thing we need you to define the contexts of the system that is, you remember when we talked about object oriented analysis, the stage was to start with right away start with identifying objects. Because we are just trying to make something, head or tail out of a video introduction or an arbitrary text and diagrams as well, but now we slowly want to bring it to the usability part of the system, so we try to understand what is the context of the system.

So the context of the system if we talk about the leave management system is the fact that we would like to understand what is the size of the organisation? What is the number of kind of number of employees, whether there are branches? What is the typical kind of leave volume that is enjoyed and so on, so forth and now that context is not given probably in the requirements, that is the background of the whole thing.

And you will ask, you will wonder is to why I need to know this context? Why is that important to define this context, this context is very significant influence the kind of design that I do. For example, I will just give you in terms of LMS, I will just give you very simple observations in that. Let us assume that we are doing a LMS for a company which works, which is start up works out of a single room and has about 10 to 20 employees.

So certainly what I can assume? I can assume that have a single desktop PC kind of desktop, where there is a single user system, where we were needs to take the leave, will come and log in and put the request and we were needs to approve a review, will again come and log in and do the task, so and as I, you log in, you will get a message screen which shows any updates in terms of the leave that you have requested or you have approved.

And you just get over with it. But in contrast, if I have an organisation which has about couple of 100 employees but all residing within the same physical network within the assembling maybe residing in 5, 6 different rooms. I will need to think about doing a system architecture which is more distributed certainly I cannot assume that there is one system where everybody comes and you know does the leave operations will assume that.

It can be operated access from multiple places and it should be possible to apply from anywhere and approve it from anywhere within the company organisation. Think about another enterprise which is kind of across different locations, may be across geographies. When we do that certainly we cannot just think about having working in a system which is on the single network will typically try to look at some web based kind of internet technology based solution which is completely loosely coupled.

The moment we do that, this is a quiet because if the organisation has offices in Kolkata, in Mumbai, in Ahmedabad and Tokyo and in Rome, then certainly the application that will be the best supported as an internet application and the moment we have that we will have a several rounds of other consequences coming in which were not specified in the requirements document in terms of what is the security of the system.

What if this will be broken through? We will need to take context of organisation in multiple other ways. For example, do people of that organisation travel very frequently, is it required? That it should have a view on mobile phones as well. Now all finally we are talking about a leave management system. Now we will say that, now when certainly, when the organisation give you the specification, the every organisation, every customer thinks that they are the whole world, they are the universe.

So they will think that their situation, their context, their office, their employee, their organisation structure is the only one of the kind that can happen in the world. Therefore, when the requirements are given for the leave management system, certainly the customer did not bother to specify all this context, because to the customer there is only one context in the world and that their organisation.

To you, the vendor, software developed engineer, the context could be anything. So it is very critical that the OD has to talk about defining the context and the success of the good design will depend on that is the second stage will significantly depend on, what is the context? And as I said that it could be just a single note solution, your architecture may not need anything at all.

Just a couple of independent modules which are linked together into one single executable, if it is, for an organisation, which is housed in a single room with 10, 20 employees, couple of different rooms, 100 employees and so on, you will need to have a distributor, more distributor architecture, distributed across geographies, lead to internet based architecture. Depending on the number of employees, you will decide, whether how with the, all these leave record and employee record will be stored.

Would you have a back in data base, how strong will be that back in data base so and so far. So all these are the factors that go into doing your system architecture design and then you will start taking clue from the object oriented analysis steps identify the objects in the system, you already have key abstractions and the structure of those key abstractions given from the object oriented analysis face that you have done.

Using those objects and the details of attributes association and so on, we will start constructing the design models that is we build the logical model, will start building the logical model, the physical model, the behavioural model and very clearly specify the different object interfaces, so that you can see that under the object interfaces again will go back to references to the architecture.

Because architecture will decide has to if there are 5 different kinds of objects, where do this objects reside. I am just to relate you back to the LMS system in this context. If you are doing the LMS system, certainly we need to know lot of employee information and those employee information, I am not specific to the leave. For example, you need to note the date of birth of the employee.

Now certainly it is quite logical to understand that if the organisation is asking for a leave management system, then certainly that organisation has 5 other systems relating to their employing management, may be in terms of a task management in terms of their payrolls and salary management and so on so far.

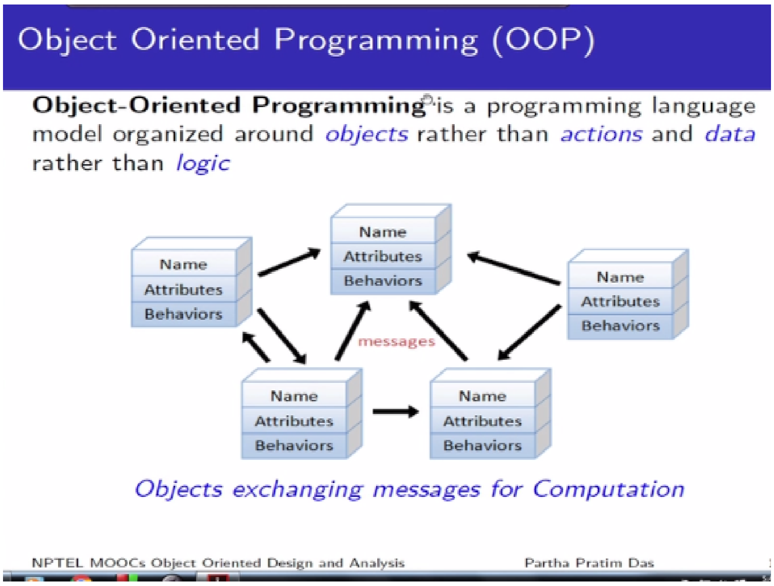
So there are certain parts of the employee related data, employee data which relates to leave, which relates to eligibility of leave, which relates to whether certain leave can be approved, enjoyed by an employee and so on, but a large part of the data is common across other systems that possibly already exist. So those data will not be the exclusive right of your system that you are building, so you can think of modelling employees, but if you model all of these data, which we will typically refer to as master data of employees.

As the soul propriety of your system, your design will not work, because you will not be allowed to arbitrarily make changes or even read some of this data, because at the master data. So it depends architecture has to decide where does this master exists? and what is the interface through which you can actually interact with this master and get extract your related information out and that is the whole exercise that this stage has to achieve.

In terms of specifically designing objects certainly these are closer to what you do in object oriented analysis, you identify the object, you decide on a representation, you classified different operations, we will see more of, understand more of this what does classification mean in terms of, who can use these operations, can any object use that operation or may be the object, define some operations which the object itself has to use and so on.

At which stage, you stop doing object oriented decomposition and start focusing on specific logic that needs to be built up, which will lead to algorithm decomposition at the lower stages, you do algorithm design and then you decide on the different external interactions, how do you package the whole thing, these are just; I can understand that many of you may not be really be able to follow what these activities will mean?

(Refer Slide Time: 10:57)



Object Oriented Programming (OOP)

Object-Oriented Programming is a programming language model organized around *objects* rather than *actions* and *data* rather than *logic*

Objects exchanging messages for Computation

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

But I just wanted to mention to give a glimpse of what are the different tasks that may be involved and these are the tasks that we will reveal to you through subsequent discussions of object models, class models and the ML language. So with this we look into the third and the last or the foundational concepts that is object oriented programming. Certainly analysis and design, analysis is primarily taking out the objects, concepts, notions from the requirements which is weak, arbitrary and inconsistent, contradictory.

Analysis is giving us certain consistent set of objects, properties, services behaviour all those. Object oriented design is now putting them in terms of the right context, in terms of the architecture, in terms of the different required system models that we have. With these 2, we will finally have to go and build the system and when you build the system we need to actually right code.

So we need programming languages and programming practices and I will emphasise on both of them together repeatedly, what I mean is it is very important to decide on the right kind of programming language because many of the languages support different kinds of features and depending on what my object oriented or design, analysis and design has revealed what has been the requirements.

The choice of language will be critically dependent on that, just to give you a simple example that If I foresee that my application, my developed system will need to be moved to several different platforms, different operating systems, different kinds of several configurations, system configurations and so on. Then I may tend to use a python kind of language lot more, because it is perfectly portable.

But at the same time I will analyse at their very hard real time constraints, which go below say in the order of below milliseconds, to be met regularly in the system, if it they exist then I cannot use python because python while it is excellent for portability simple easy to code and give up is moderately slow. I should not say moderately slow; it is extremely slow compared to languages like C or C++.

So the choice of object oriented programming platform will depend on the result of analysis, the result of the design that you are making and will also depend on the way you actually are going to design and implement the system. Now it is a common notion that C++ is object oriented language and therefore many of us tend to believe that it is, if I need to do some object oriented programming then I must use C++, while it is good to use C++, it gives you lot of advantages.

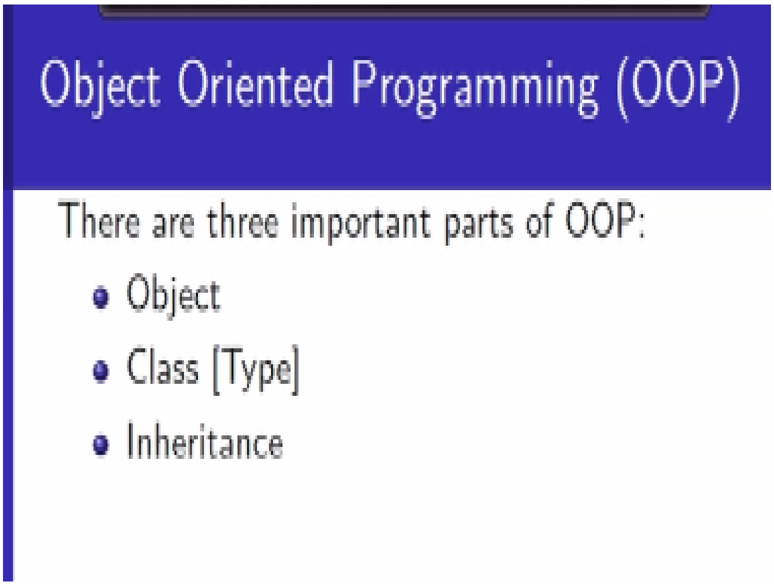
But it is true that many programmers over the years have built many systems with the quietest strict object oriented principle, quite a strict object oriented analysis design paratime by using simple C language, not using C++.Of course you would, I mean you would wonder and ask us to why should someone do that? There are certain reasons for doing that, for example, C is lot more like it.

C is available, C compiler is available for lot many more processes than where C++ compilers available. So when you build small embedded systems, not a big mobile phones, but small systems like a simple voice recorder, a simple answering machine and so on, you may be working with the processor where getting a C++ compiler itself could be a (()) (14:56), I mean the compiler itself, processor is may be so small, then the compiler itself may not run in that processor.

So but it is much easier to get a good C compiler on those, so you program on C compiler, so it is very widely found that several of embedded systems are actually programmed in C, that does not mean that their designs or their implementations practices are not object oriented, object orientation is supported by object oriented programming, but object orientation is more in terms of our thoughts, in terms of our identification, in terms of our analysis and design.

Then the object oriented programming vehicle, the more object oriented, my programming vehicle is; it is easier to program, it is less effort consuming, but we can still program with other platforms as well. So the object oriented programming simply as you can understand, it is organised around objects and rather than looking at actions typically which a more traditional language will do or looking at logic, this looks with objects which basically encapsulate data, we have already seen that.

(Refer Slide Time: 16:32)



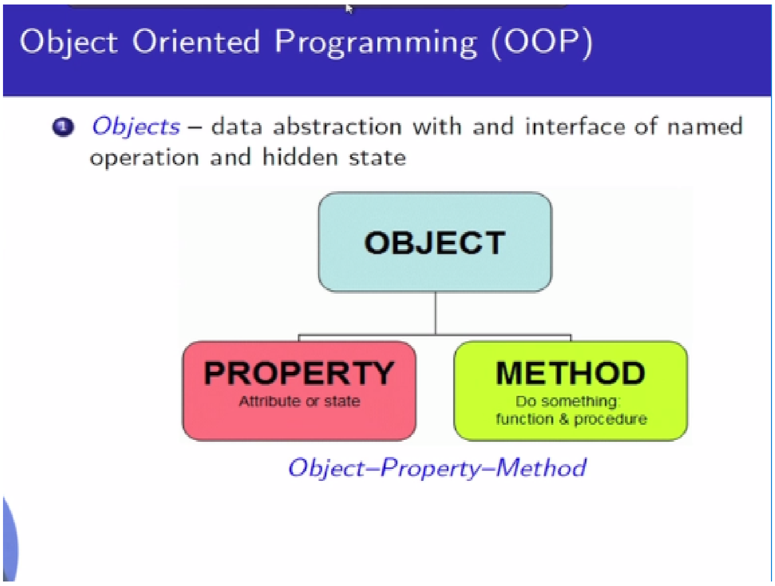
Object Oriented Programming (OOP)

There are three important parts of OOP:

- Object
- Class [Type]
- Inheritance

The diagram here is the variant of what we had discussed earlier in terms of the client server model, so these are different objects and they are sending messages between each other to as achieve the different tasks of that what they need to do. There are 3 important parts of object oriented programming and whenever we say that a language supports object oriented programming will typically look for the history.

(Refer Slide Time: 16:51)



Object Oriented Programming (OOP)

1. *Objects* – data abstraction with and interface of named operation and hidden state

OBJECT

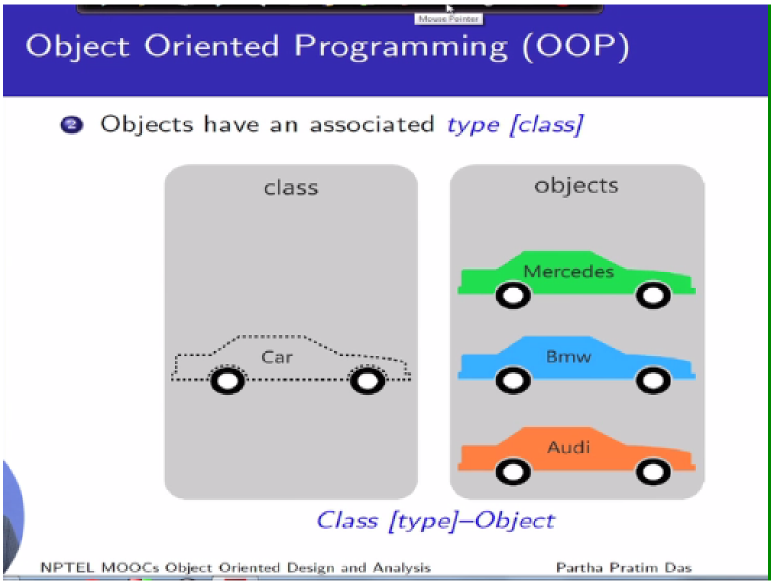
PROPERTY
Attribute or state

METHOD
Do something:
function & procedure

Object–Property–Method

Rather we will mandatorily look for the history objects, classes and inheritance. So first is objects, so any object oriented programming language will support the explicit creation definition of objects with 2 components; one is properties or attributes which are basically data and the values of this data will define the state of the object, so if you talk about a leave, then the start date of the leave is a data, the end data of the leave is the data.

(Refer Slide Time: 17:49)



Object Oriented Programming (OOP)

2. Objects have an associated *type [class]*

class

objects

Class [type]–Object

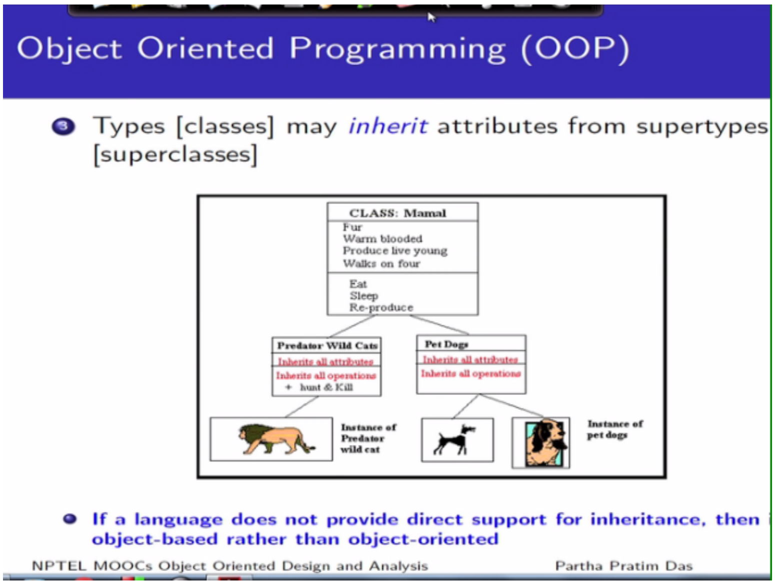
NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

So their values will decide to what is the state of the leave? And the other component is the collection of methods. These are the different services that an object can provide, in terms of a leave object, this will be services like approved leave, revoke leave, enjoy leave and so on. So the objects of the first part that need to be there in an object oriented programming language or system.

The second is objects are associated with type, so this is just an illustration from the automotive industry, right here, you can see variety of different cars; Mercedes, BMW, Audi, we could think about several others but in terms of the basic nature, basic behaviour that they support their say they all are used for transportation and we can say that this are cedars, they are the boutique they can be controlled, their air conditions and all that.

They have a lot of commonality of between what their properties are and that constitutes what we say is a class and object oriented programming languages often will refer to the class as their type, a type like integer is a type, which can add, subtract, multiply. String is a type which can contact in it, which can divide a string at a given point and so on. Similarly, a class will give us all the different properties that you can support.

(Refer Slide Time: 19:11)



Object Oriented Programming (OOP)

3. Types [classes] may *inherit* attributes from supertypes [superclasses]

CLASS: Mamal
Fur
Warm blooded
Produce live young
Walks on four
Eat
Sleep
Re-produce

Predator Wild Cats
Inherits all attributes
Inherits all operations
+ hunt & Kill

Pet Dogs
Inherits all attributes
Inherits all operations

Instance of Predator wild cat

Instance of pet dogs

- **If a language does not provide direct support for inheritance, then object-based rather than object-oriented**

NPTEL MOOCs Object Oriented Design and Analysis Partha Pratim Das

It is a fundamental concept of object oriented program. Finally, the third will be the concept of inheritance or specialisation which we will need to explain more, but just if you look into the simple example, here we are talking about a class mammal, some group class. These are its properties, attributes we talked of that is it is far, it is warm blooded, it can produce live young of its kind, it can walk on 4 legs and so on, is a mammal.

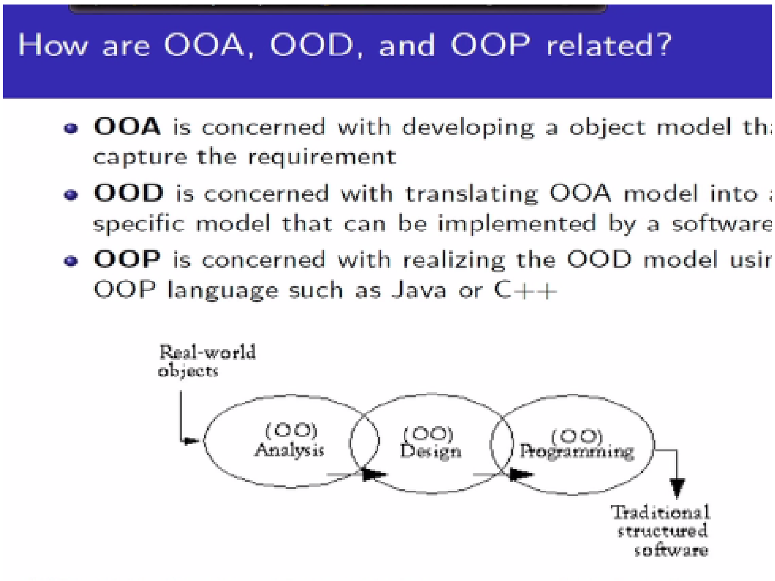
These are the different services, these are the different methods that the class or the object of this type will provide that it can eat, it can sleep, it can reproduce and so on. Now given this basic concept, here we look at one more which you say is a predator wild cats and you say predator wild cats is a mammal, this is an abstraction hierarchy, it has the predator like a lion, inherits all the properties that the mammal has.

The lion can eat, the lion can sleep, a lion can reproduce but it is some more, it has some more, it can hunt, it can kill, so this is what is expressing the concepts interrelated hierarchy where the inheritance can happen that is properties of the super class or the base class which is higher up here, come down to the specialised class the subclass is the basic structure of inheritance, which the object oriented programming languages must support.

For example, if you look into this other side, again the properties come down here, here we are talking about another concept of pet dogs which inherit everything but certainly unlike lamb, this pet dogs will not add its own properties or own methods like hunt and kill, may be you can say you will have a property called cuddle, you can cuddle your pet dog right. So this inheritance is an inherent requirement of an object oriented programming language.

Now there are some programming language, some programming systems which just support objects and classes that is objects and types. But do not support inheritance those languages will typically referred to them as object based language, but not object oriented language, just more a matter of terminology but most of the languages that you will get to use in todays programming while are actually object oriented programming languages and this is what you will be using.

(Refer Slide Time: 22:10)



How are OOA, OOD, and OOP related?

- **OOA** is concerned with developing a object model th: capture the requirement
- **OOD** is concerned with translating OOA model into : specific model that can be implemented by a software
- **OOP** is concerned with realizing the OOD model usir OOP language such as Java or C++

Real-world objects

(OO) Analysis

(OO) Design

(OO) Programming

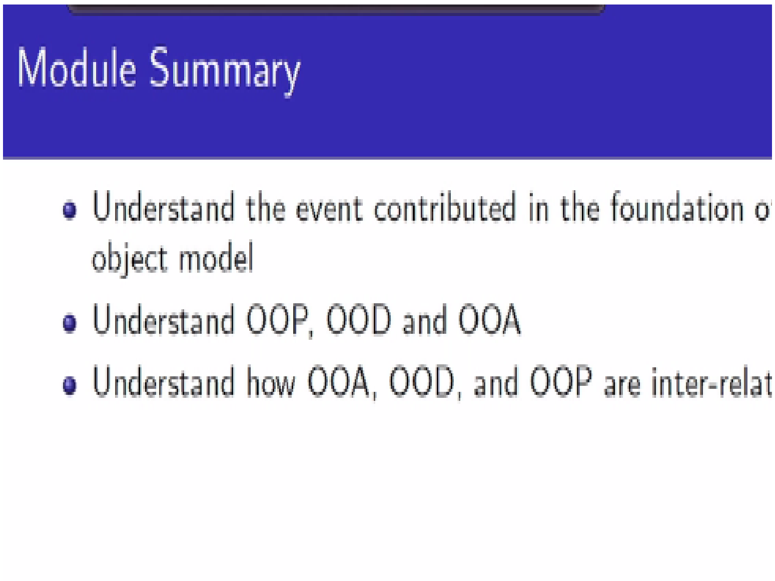
Traditional structured software

So of all these 3, now we have seen how we start with object oriented analysis to can the requirements, to generate the basic object oriented concepts of abstractions of classes and objects, their structure and inter relationship, how object oriented design basically translates the object oriented analysis into models that are specific and required for the system keeping the exact contents of the system and several other non-functional parameters like performance and cast and also into mind.

Then finally the whole process of object oriented design also will make choice about what kind of programming language and platform we will use and object oriented programming will be engaged in actually realising these design of classes, objects, interactions and constraints on the actual software system that we are building into, so which will use some language like java, C++ or python, something like that.

While this particular course, we did not really put any prerequisite of knowing this object oriented languages because we appreciate that some of you may be new to the whole concept but I would suggest that in addition to understanding this object notions and analysis and design techniques, please start picking up at least one of the languages, read Java, read C++ or pythons.

(Refer Slide Time: 24:31)



Module Summary

- Understand the event contributed in the foundation o object model
- Understand OOP, OOD and OOA
- Understand how OOA, OOD, and OOP are inter-relat

So that you can practice some programming when we come to actually realising the designs in terms of the object oriented programming systems so that you will get a confidence and a more concrete idea in terms of how really all these abstract notions of object class structure hierarchy and all those finally get realised in terms of the concrete software program. So to summarise, we have a here try to understand the events that have contributed in the foundation of the object model.

Specifically taken a look into object oriented analysis, object oriented design and object oriented programming, summarising with their inter relationship to make good object based systems. In the next module onwards, we will start focussing on specific aspects of object models and slowly get into their identification and use in the object oriented system.