OBJECT ORIENTED SYSTEMS OF KNOWLEDGE REPRESENTATIONS

DR. KRISHNENDU GUHA

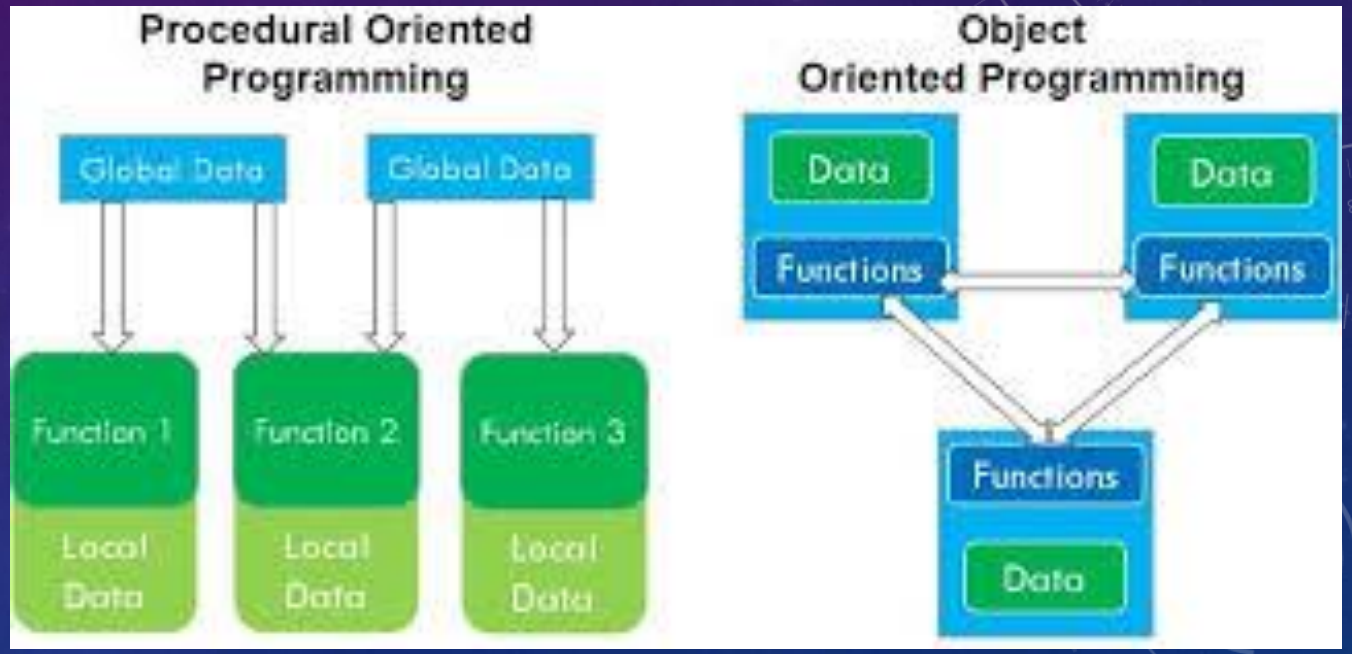
ASSISTANT PROFESSOR (ON CONTRACT)

NATIONAL INSTITUTE OF TECHNOLOGY (NIT), JAMSHEDPUR, INDIA

EMAIL: KRISHNENDU.CA@NITJSR.AC.IN

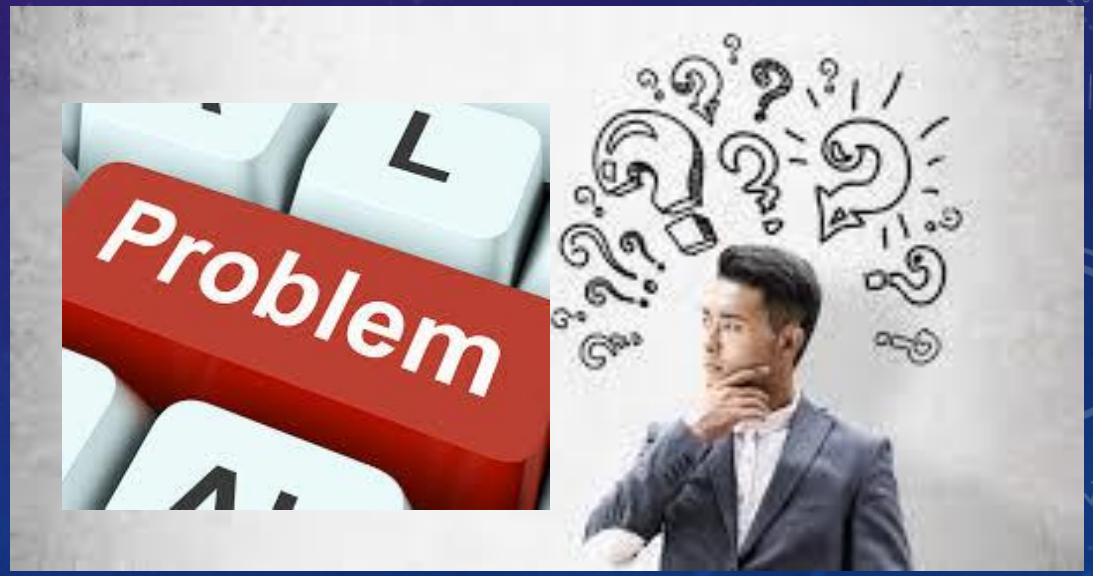
INTRODUCTION

- In procedural programming languages such as Pascal or Fortran, a program consists of a procedural part and a data part
- The procedural part consists of set of program instructions and the data part (the numbers and character strings which are manipulated by instructions)
- Programs typically contain several modules of instructions which perform computations on the same data set



WHAT IS THE PROBLEM?

- When some change is made to the format of the data, every module which uses it must then be modified to accommodate the newly revised format
- This places a heavy burden on the software maintenance process and
- Makes these types of programs more prone to errors



OOS

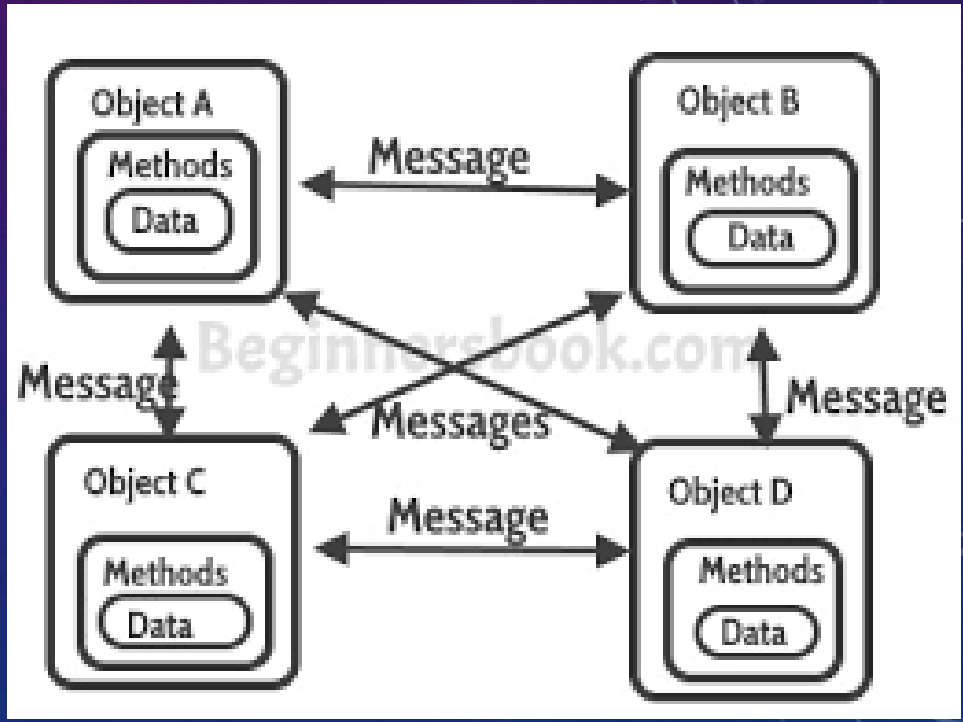
- In an object-oriented system (OOS) the emphasis between data and procedures is reversed
- Data becomes the primary object and procedures are secondary
- It is a well known system design principle used to make systems more modular and robust
- Here, objects are associated with their own procedures and as such are responsible for their own actions
- Thus, when some change is required in the data or procedure, only changed object need be modified
- Other objects are not affected and therefore require no modification
- In object oriented systems there is simplicity in structure because almost everything is an object
- For example, a car can be regarded as an object consisting of many interacting components or sub-objects

- The object paradigm seems to model real-world systems more closely than the procedural programming models where objects (data) and procedures are separated

- In object oriented system objects become individual, self-contained units which exhibit a certain behaviour of their own and interact with other objects only through passing of messages

- Tasks get performed when a message is sent to an object which can perform the task

- All the details are rightfully hidden from other objects, i.e. ensures privacy



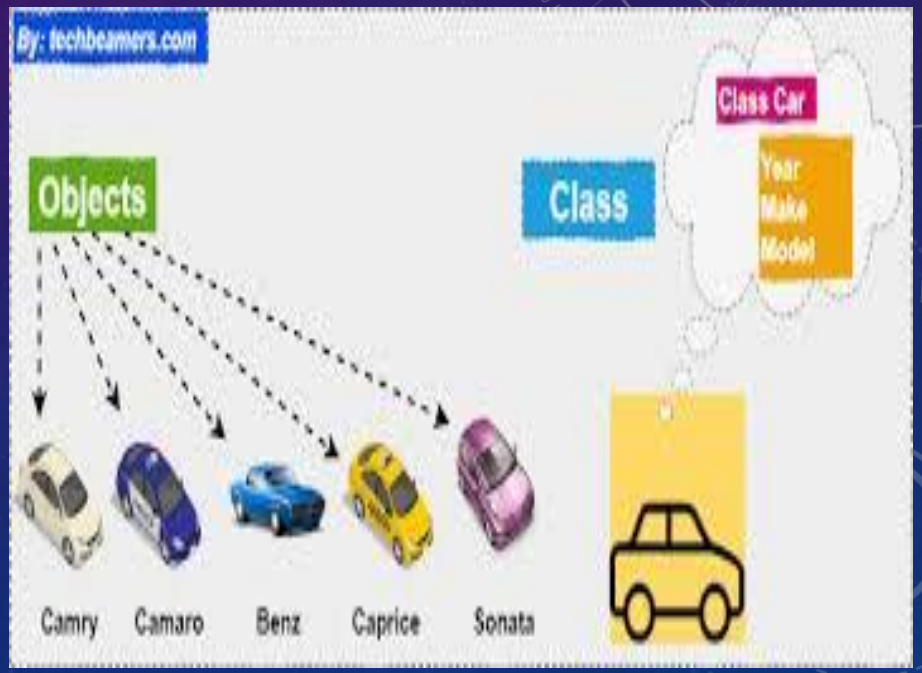
IDEA BEHIND OBJECT ORIENTED SYSTEMS OF KNOWLEDGE REPRESENTATION:

- The basic idea behind an OOS is the notion of classes of objects interacting with each other to accomplish some set of tasks
- The objects have well defined behaviours
- They interact with each other through use of messages
- When a task needs to be performed, an object is passed a message which specifies the task requirements
- The receiving object then takes appropriate action in response to the message and responds by returning a message to the sender
- In performing the required task, the receiver may need assistance from other objects, thereby prompting further messages to be sent
- In general a task may consist of any definable operation

OBJECTS, CLASSES, MESSAGES, METHODS OF OBJECT ORIENTED SYSTEMS:

OBJECTS

- Objects are the basic building blocks in OOS
- All entities except the parts of messages, comments and certain punctuation symbols are objects
- An object consists of a limited amount of memory which contains other objects (data and procedures)
- Objects are characterized by attributes and by the way they behave when messages are sent to them
- All objects belong to some class
- They are created by declaring them as instances of an existing class and instantiating instance variables
- The class to which an object belongs can be determined by sending it the message “Class”



MESSAGES

- Actions are performed in an OOS by sending messages to an object
- This corresponds to a function or procedure call in other languages

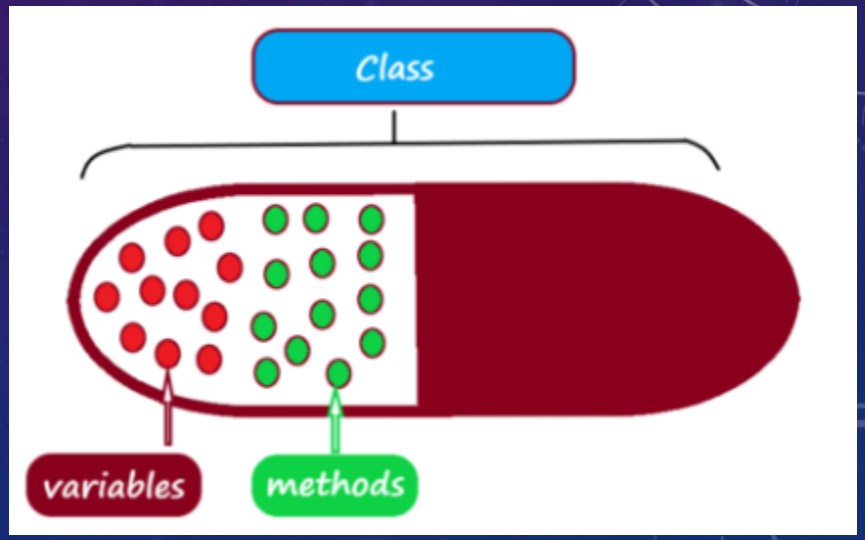
- The messages are formatted strings composed of three parts:
- a receiver object,
- a message selector and
- a sequence of zero or more arguments

- The format of a message is given as:
- <object selector> <args>, <args>

METHODS

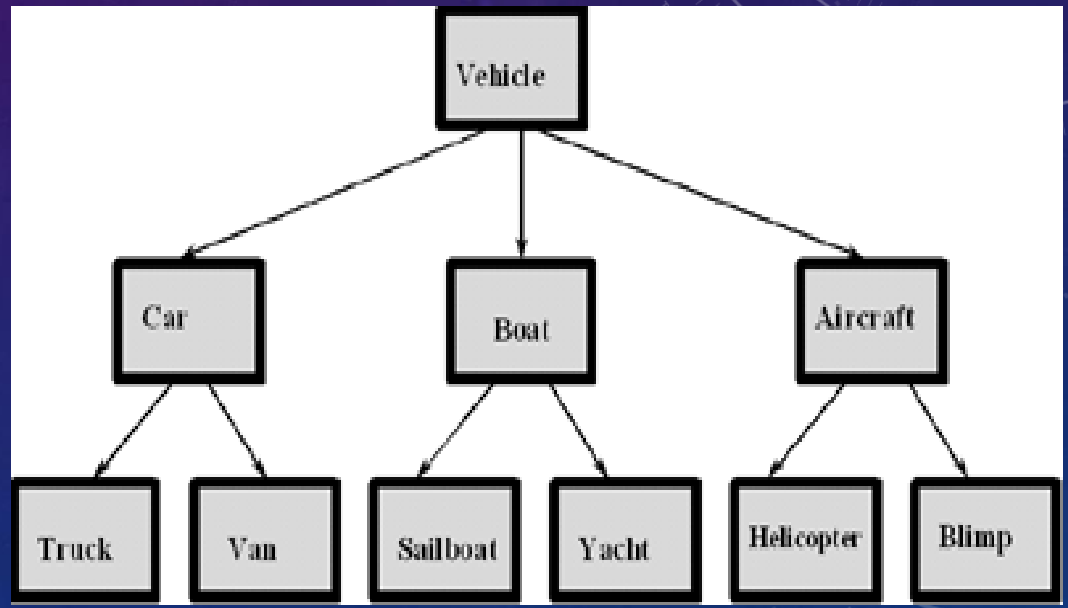
- Procedures are called methods
- They determine the behaviour of an object when a message is sent to an object
- Methods are the algorithms or sequence of instructions executed by an object

- For example
- in order to respond to the message 5+7, the object 5 must initiate a method to find the sum of integer numbers 5 and 7
- On completion of the operation, the method returns the object 12 to the sending object.



CLASSES AND HIERARCHIES

- A class is a general object which defines a set of individual (instance) objects which hare common characteristics
- All objects are instances of some class and classes are sub-classes of some higher class, except for a most general root class
- The root class for an OOS is the class named object.
- When a message is sent to an object, a check is first made to see if the method is for the object itself or its immediate class can perform the required task
- If not, the methods of the nearest superclass are checked
- If they are not adequate, the search process continues up the hierarchy recursively until methods have been found or the end of a chain has been reached
- If the required method is not found, an error message is printed
- This is process of inheritance



- Object oriented systems are useful in modelling many real world situations in which real object can be put in one to one correspondence with program classes and objects
- We may conclude by saying that semantic nets are useful for their applications in monotonic and non-monotonic reasoning
- CDs are more powerful tools for knowledge representation, but have limited use in monotonic systems
- Scripts oriented representation are mainly useful to represent complex scenes, which by other means are too difficult to be realizable

EXAMPLE

- Let for example, we have a robot planning program and we want to know if we move a table into another room, what other objects also change location

- In (Table1, Living room)
- Made-of (Table1, Wood)
- On (Vase1, Table1)
- Made of (Vase1, Glass)
- On (Vase2, Table2)
- On (Lamp1, Table1)

- (Living room1:
  - Contains:
    - (Table 1:
      - Made of: Wood
      - Has-on: (Vase1:
        - Made of: Glass)
      - (Lamp1: …))
    - (Table 2:
      - Has-on: (Vase 2:…)))


Thank
You
Obrigado
Merci
Asante
Grazie
Kiitos
Maake
Welalin
Spasibo
Vinaka
Arigato
Matondo
Juspaxar
Mamana
Multumesc
Dank Je
Nirringrazzjak
Ua Tsaug Rau Koj
Cam on ban
Mochchakkeram
Kia Ora
Obrigado
Chokrane
Danikon