
Monsoon 2019

Lecture - 1

# Object Oriented Programming

by

Dr. Rajendra Prasath

Indian Institute of Information Technology

Sri City – 517 646, Andhra Pradesh, India

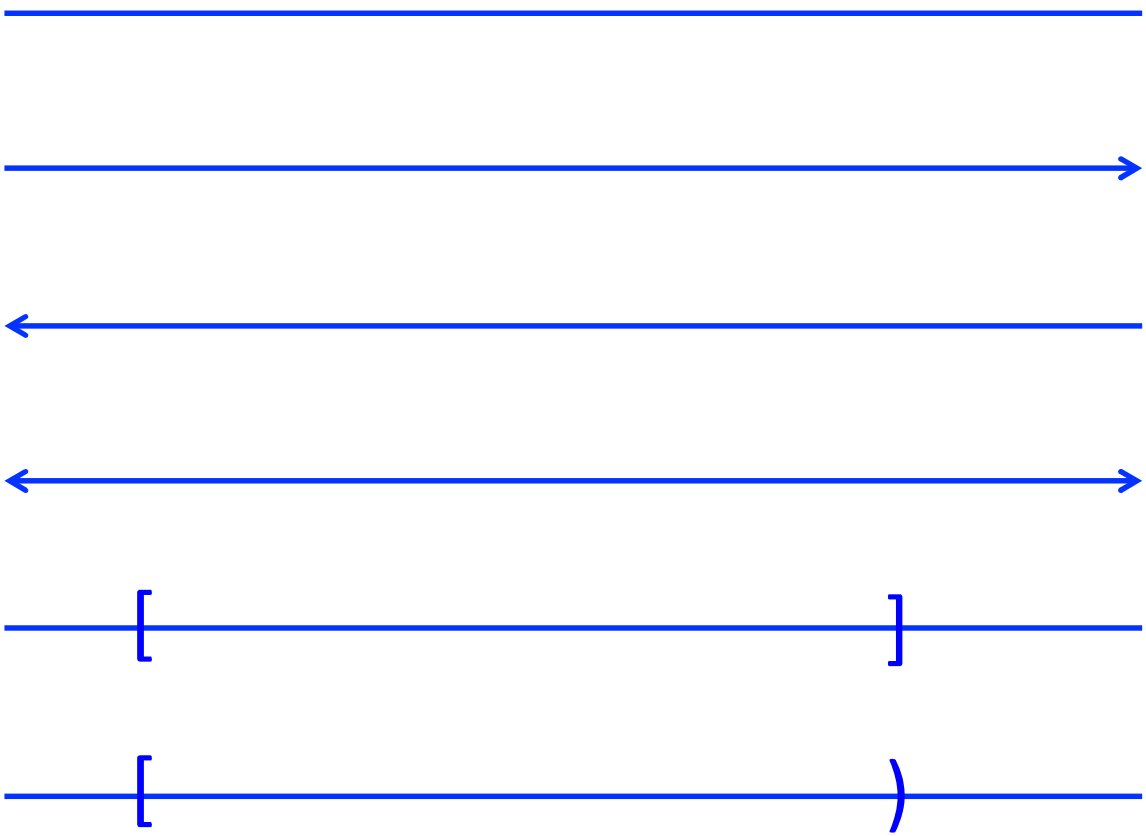
# What is OOP?

- **Object Oriented Programming:**
  - Programming will be the main focus in such a way where everything is an object that interact with one another

- Designing Classes & Objects
  - An incremental, and iterative processes
  - Is it difficult to design right from the first time

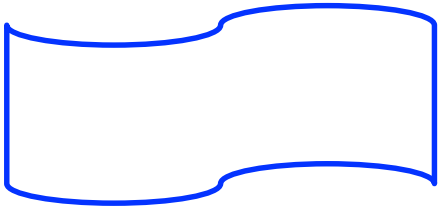
  - Is it easy to make changes?
  - Flexibilities in adapting Object Oriented Design and Analysis
  - Many more facets of OOPs

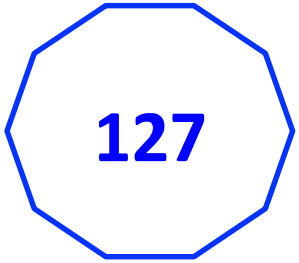
# What is an Object?

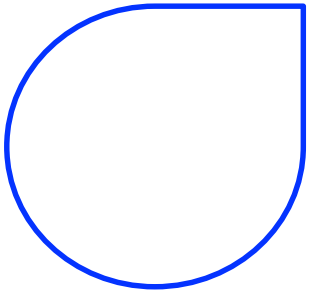


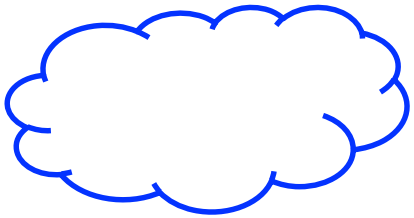
# Objects?



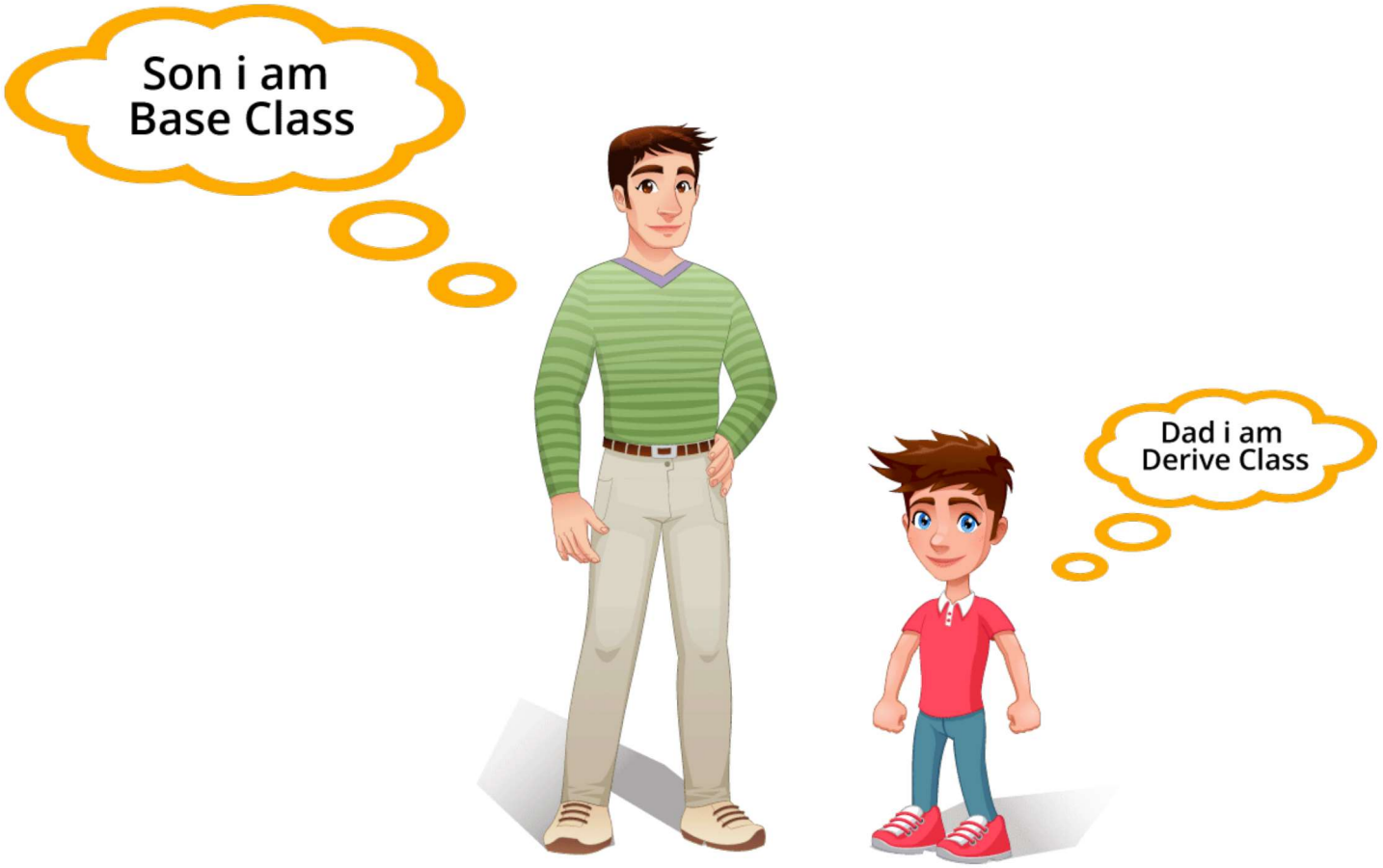






# Objects – Relations?



# Programming Techniques

Different Types:

- a) Unstructured Programming
  - Assembly language programming
- b) Procedural Programming
  - Assembly language, C programming
- c) Object Oriented Programming
  - C++, Java, Smalltalk, C#, Objective C

# Programming Techniques

- Unstructured Programming
  - This consists of just writing the sequence of commands or statements in the main program, which modifies the state maintained in Global Data. Example: Assembly Language programs.
- **Limitations of Unstructured Programming**
  - a) The data is global and code operates on it
  - b) As the size of code increases, maintenance is a problem
  - c) Does not have independent data for processing
  - d) The concept of local variables did not exist
  - e) Reusability of code was not supported

# Programming Techniques (contd.)

- Machine Language is the language which a Central Processing Unit (CPU) of a computer understands and consists only of numbers
  - Written in the form of 0 & 1

- Any computer can directly understand only its own machine language, defined by its hardware design
  - 1011000000000101
  - 1011000000000001
  - 1101110000000111
- Machine Dependent

# Programming Techniques (contd.)

- **Assembly Language:** Similar to machine language, but provides names for numeric instructions present in the machine language, making it easy for the programmer
  - It is a low level language
  - Use Naturally understandable symbols called "Mnemonics"
  - Example:
  - Load 5
  - Load 1
  - Add

- Assemblers are used as interpreter
- It works directly with microprocessor

# Assembly vs Machine Language

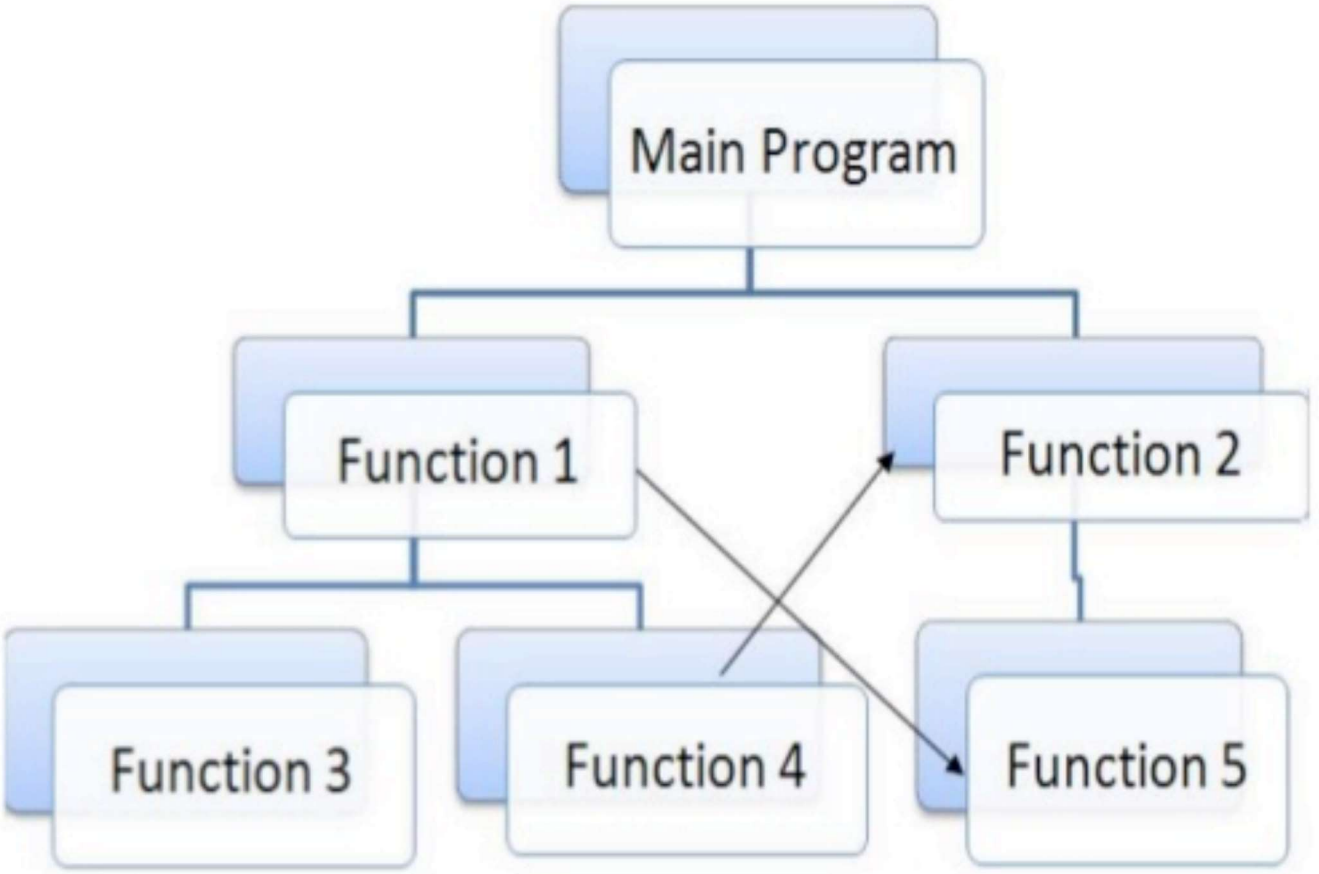
- Assembly Language — Machine Code

| Assembly Language | Machine Code |
|---|---|
| SUB AX, BX | 01010101011 |
| MOV CX , AX | 01010101010 |
| MOV DX , 0 | 11010101010 |

- Assembly language is intermediate between high level language and machine code

# Program Structure

- An Example



# A Real World Example

- A car company needs to update its online inventory system
- Write a program for two similar but separate forms for a website:
  - one form that processes information about cars
  - one that does the same for trucks

- For cars, record the following information:
  - **Color, Engine Size, Transmission Type, Number of doors**
- For trucks, the information will be similar, but slightly different:
  - **Color, Engine Size, Transmission Type, Cab Size, Towing Capacity**

# A Real World Example

- A car company needs to update its online inventory system
- Write a program for two similar but separate forms for a website:
  - one form that processes information about cars
  - one that does the same for trucks

- For cars, record the following information:
  - **Color, Engine Size, Transmission Type, Number of doors**
- For trucks, similar information, but slightly different:
  - **Color, Engine Size, Transmission Type, Cab Size, Towing Capacity**
- For Passenger Bus service:
  - **Color, Engine Size, Transmission Type, Cab Size, Number of passengers**

# Procedural vs OOP

- **Procedural:** We change three different forms:
  - cars, trucks, and buses
  - Here color has be hard coded

- **OOP:** We change the color method in the vehicle class
  - **Why?**
  - car, truck, and bus classes all extend (or inherit from, to put it another way) the vehicle class, they are automatically updated

# Are all cars of same type?

- Generic car to specific makes
  - For example: Rolls-Royce, Ford, Nissan and Mazda
  - **Procedural:** We create a new schema for each make, repeating all of the code for generic car information and adding the code specific to each make
  - **OOP:** We extend the car class with a Ford or Nissan class or a Mazda class and add methods for each set of unique information for that car make.
- **Make Changes:** transmission type
  - **Procedural:** We open and update each schema
  - **OOP:** change transmission Type method in vehicle class and changes reflect in every inherited class

# Procedural vs OOP

✧ Differences:

| Procedural | OOP |
|---|---|
| Program is divided into small parts called functions | Program is divided into parts called objects |
| This category does not have any proper way for hiding data so it is less secure | OOP provides Data Hiding so provides more security |
| Example : C, VB, FORTRAN, Pascal | Examples: C++, JAVA, VB.NET, C#.NET |
| Top down design, Global data focused, Limited code reuse, Complex code | Object focused design, Protected data, Code reuse, Complex design |

# What is an OO Model?

- A model is an abstraction of something
- Purpose is to understand the product before developing it

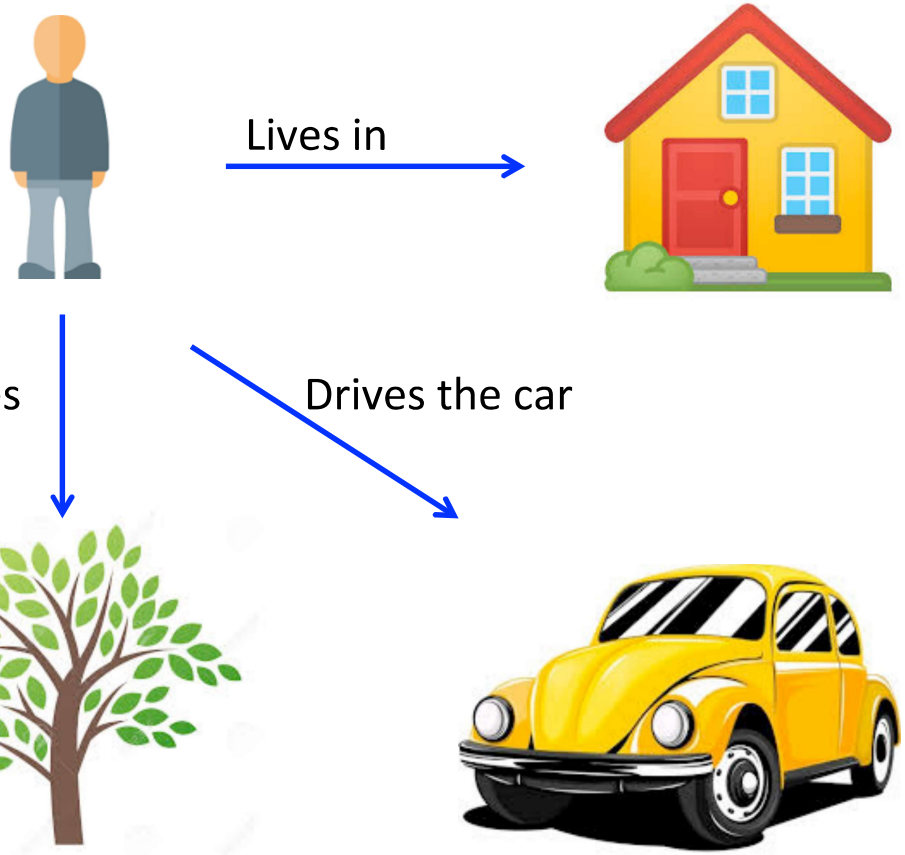
- **Examples**
  - Highway maps
  - Architectural models
  - Mechanical models

# Model – An Example

- Various Objects and their relations

# OO Model

- **Objects:**
  - Person
  - House
  - Car
  - Tree




- **Interaction**
  - Person lives in the house
  - Person Drives the card
  - . . .

# Advantages

- Think in terms of objects
  - OO models map to reality
  - OO models are: easy to develop and understand
- Objects are modeled on real world entities
  - This enables modeling complex systems of real world into manageable software solutions
- Advantage of OOPS
  - It provides data hiding
  - It provides data encapsulation
  - It provides data abstraction
  - It provides reusability of code
  - It provides easy code maintenance

# Assignments / Penalties

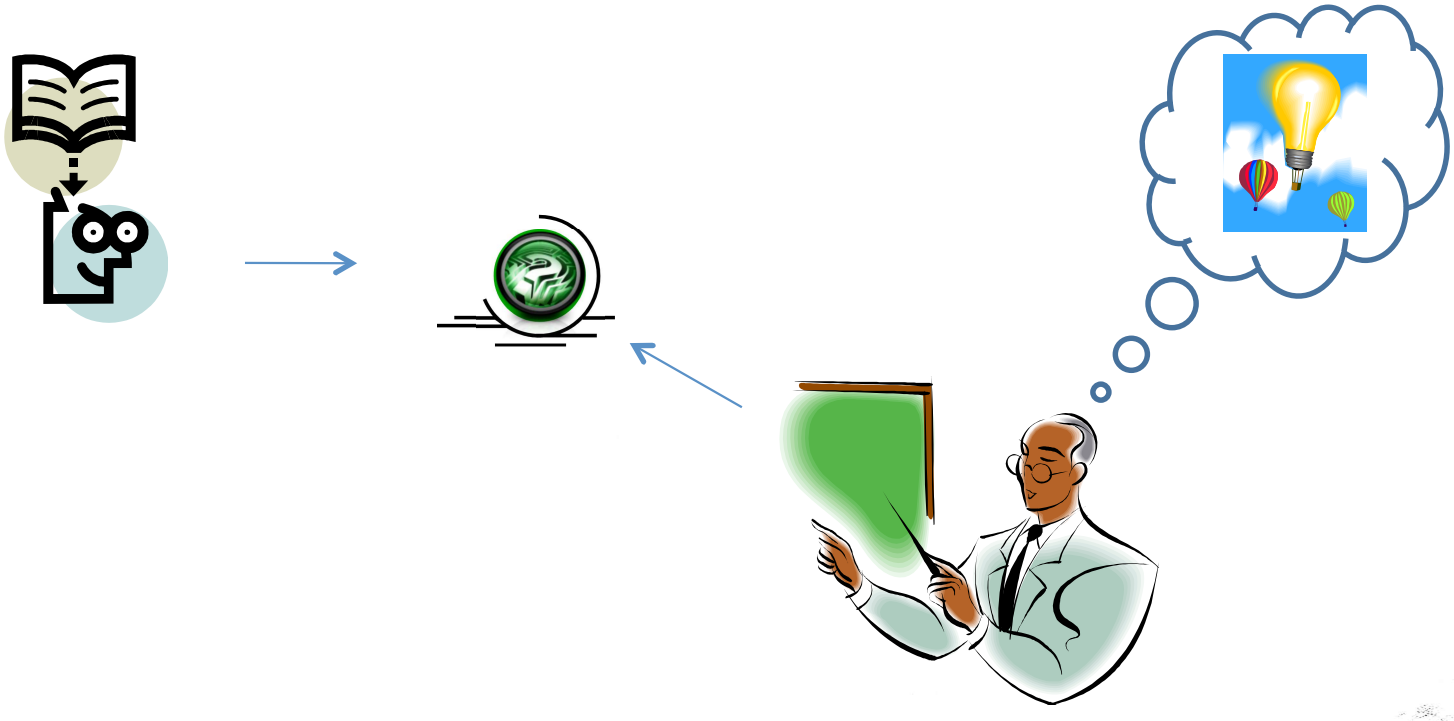
- Every Student is expected to complete the assignments and strictly follow a fair Academic Code of Conduct to avoid severe penalties

- Penalties would be heavy for those who involve in:
  - **Copy and Pasting** the code
  - **Plagiarism** (copied from your neighbor or friend – in this case, both will get "0" marks for that specific take home assignments)
  - If the candidate is **unable to explain his own solution**, it would be considered as a "copied case" !!
  - **Any other unfair means** of completing the assignments

# Assistance

- You may post your questions to me at any time
- You may meet me in person on available time or with an appointment
- You may leave me an email any time
  (email is the best way to reach me faster)

# Thanks …



… Questions ???