CIT 103 & CIT 104

Object Oriented Programming

By

Md. Palash Uddin

Lecturer

Dept. of CIT

Hajee Mohammad Danesh Science and Technology University, Dinajpur.

Course Objective

- Objective of this course is to make students familiar with the **concepts of object-oriented programming.**
- Concepts will be reinforced by their implementation in **C++ and Java.**

Books

- Object Oriented Programming with C++
  By E Balagurusamy
- Teach yourself C++
  By Herbert Schildt
- The C++ Programming Language
  By Bjarne Stroustrup
- The Complete Reference C++
  By Herbert Schildt
- Java How to Program
  H.M.Deitel, P.J.Deitel –
- Java Programming
  Joyce Farrell
- And so on…..

What is programming?

- Programming is taking

  A ***problem***

  Find the area of a rectangle

  A set of ***data***

  length

  width

  A set of ***functions***

  area = length * width

- Then, applying functions to data to solve the problem
- **The purpose of a programming is to help express ideas in code.**

  -Bjarne Stroustrup

Programming Paradigm

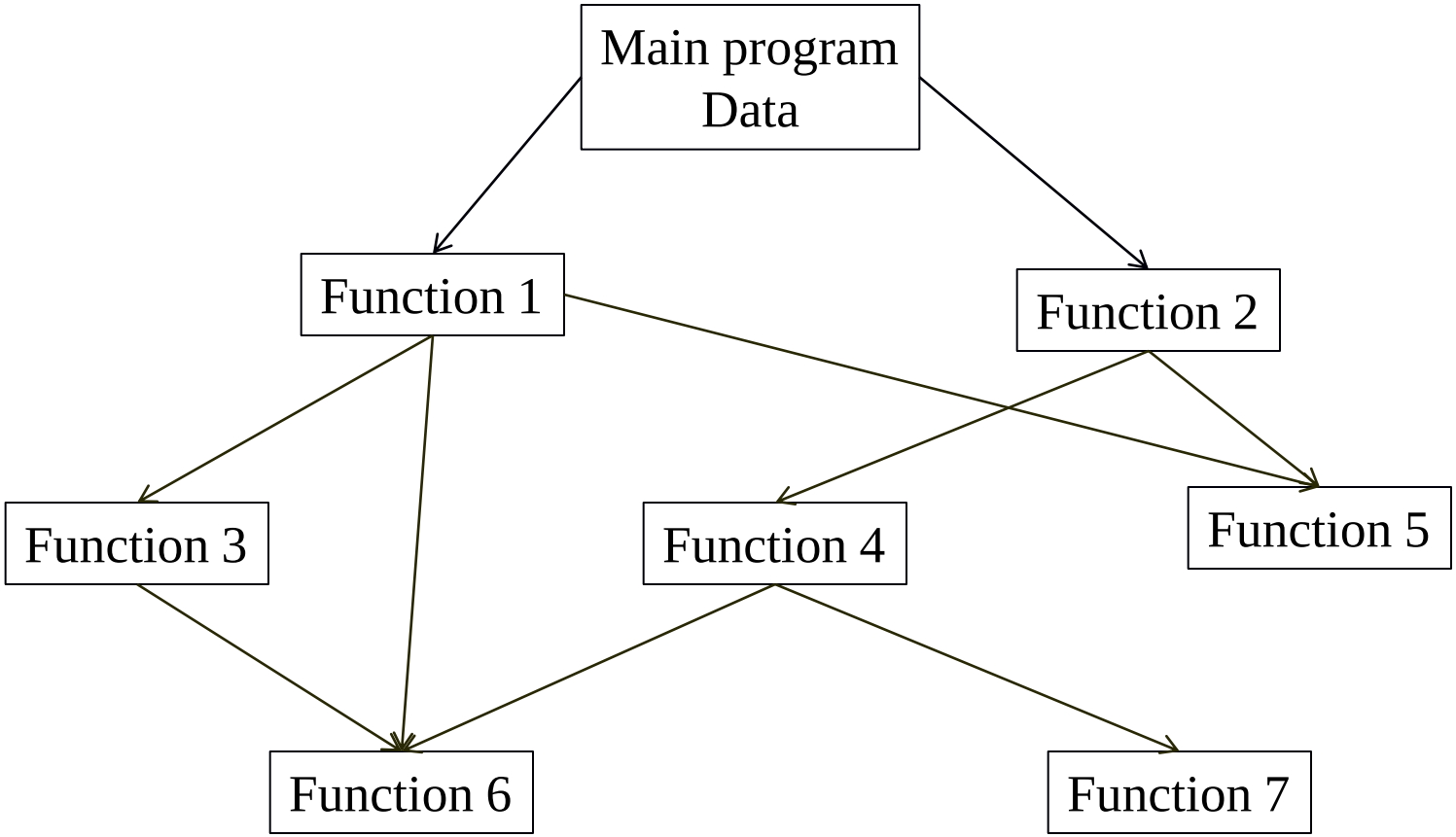
- Procedural (Structure-oriented)
- Object-Oriented Programming
- Functional
- Logic
- Scripting

Procedural Concept

- The problem (to be solved) is viewed as **a sequence of things (operation)** to be done.
- **Operations** (actions/instructions/commands) may be **reading, calculating, printing** etc.
- Use **flowchart** to organize these actions
- A **function** (Procedure) is written to accomplish the operations
- A list of **instructions/commands** is in a function
- Thus primary focus on **functions**
- So, it's also called **structure-oriented**, **action-oriented**, **instruction-oriented**.

Procedural Concept

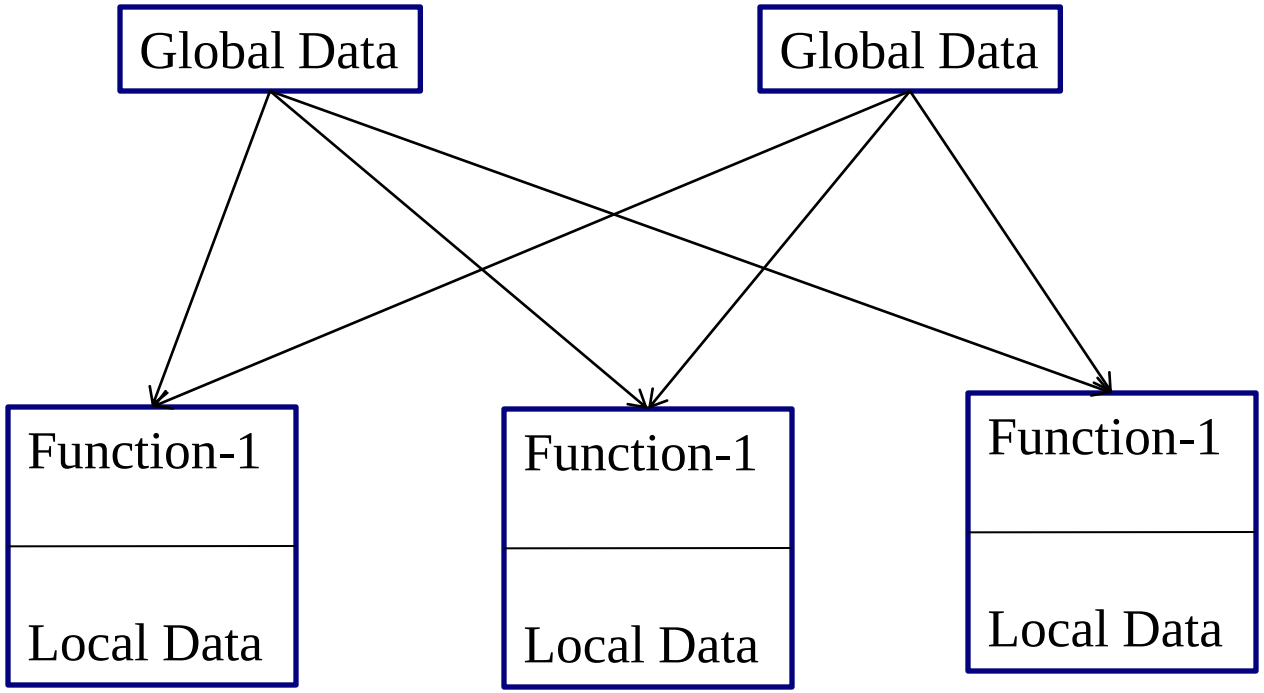
Hierarchical **decomposition** of functions



- The main program coordinates **calls to procedures** and hands over appropriate data as parameters.

Procedural Concept

Relationship of **data and functions**:



- Global data are **more vulnerable** to an inadvertent change by a function

Procedural Concept

- **Procedural Languages**
  - **Fortran, COBOL, Pascal, Basic, C ….**
- **Shortcoming of Structured Programming:**
  - Very **little attention on data**
  - What **happens to the data**? **Data move openly** around the system from **function to function**
  - How are they **affected by the functions**? Data are transformed by the functions from **one form to another**.
  - Most of the **functions share global data**
  - Difficult to identify **what data is used by which function**
  - Does **not model real world problems** very well

Pre-discussion for OOP

- When we need a ***single value*** of data, we use a ***variable*** with the required ***data type***.
  - For example: **int student_id**;
- When we need multiple values of same data type, we use an ***array variable*** with the required ***data type***.
  - For example: int **student_id[100]**;
- When we need multiple logically related values of different types, we use a ***structure variable.***

Pre-discussion for OOP

- **Structure:**
  - *For example:*

```
struct students
        {
                char name[50];
                char address[100];
                char dept_faculty[20];
                int level;
                char semester[3];
                int session;
                float CGPA;
        } student[100];
```

- Structure variable is **not like built-in types**: student[1]=student[2]+student[3]; **impossible**
- Do not permit **functions** and **data hiding** in it.

Pre-discussion for OOP

- The concepts of object has been introduced to remove the **drawbacks** of structure variable i.e.,
  - ***To make it built-in type***
  - ***To permit functions***
  - ***To support data hiding***
- Informally,

Object=Properties of structure variable + removal of the drawbacks of structure variable + some extra features

and

OOP= programming with **objects**

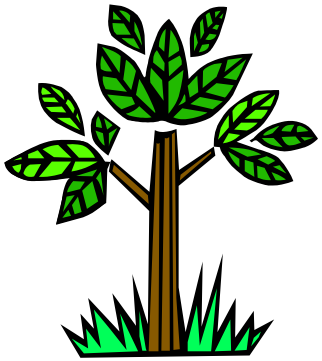
What is Object-Orientation?

- A technique for **system modeling**
- OO model consists of several **interacting objects**

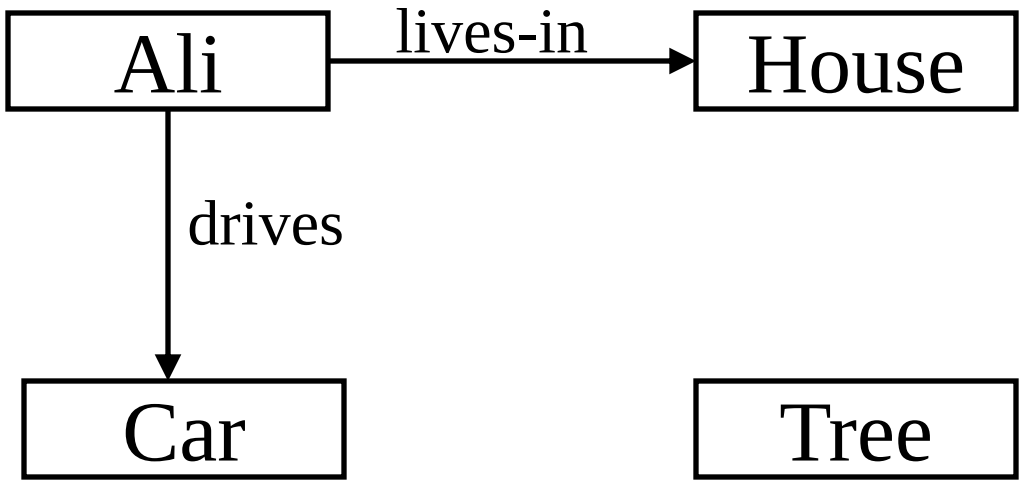
What is a Model?

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

Example of Objects for OO Model



Example – OO Model



- Objects
  - Ali
  - House
  - Car
  - A book
  - A student
  - Tree
- Interactions
  - Ali lives in the house
  - Ali drives the car

Object-Orientation - Advantages

- People think in terms of **objects**
- OO models map to **reality**
- Therefore, OO models are
  - easy to develop
  - easy to understand

What is an Object?

- An ***object*** is an ***abstraction*** that represents **an entity** in the real world which can be **distinctly identified**
- Something tangible (**Ali**, **Car**)
- Something that can be **apprehended intellectually** (Time, Date)

What is an Object?

- An object has:
  - ✓ **State (attributes)**: The *state* of an object consists of a set of *data fields* (also known as *properties*) with their current values.
  - ✓ **Well-defined behavior (operations)**: The *behavior* of an object is defined by a set of methods that describe how to carry out operations
  - ✓ **Unique identity**
- An object is an ***encapsulation*** of both functions and data

Example – Ali is a Tangible Object

- **State (attributes)**
  - Name
  - Age
  - Address
- **Behavior (operations)**
  - Walks
  - Eats
- **Identity**
  - His name or national id no.

Example – Car is a Tangible Object

- **State (attributes)**
  - Color
  - Model
- **Behavior (operations)**
  - Accelerate
  - Start Car
  - Change Gear
- **Identity**
  - Its registration number

Example – Time is an Object Apprehended Intellectually

- **State (attributes)**
  - Hours
  - Minutes
  - Seconds
- **Behavior (operations)**
  - Set Hours
  - Set Minutes
  - Set Seconds
- **Identity**
  - Would have a unique ID in the model

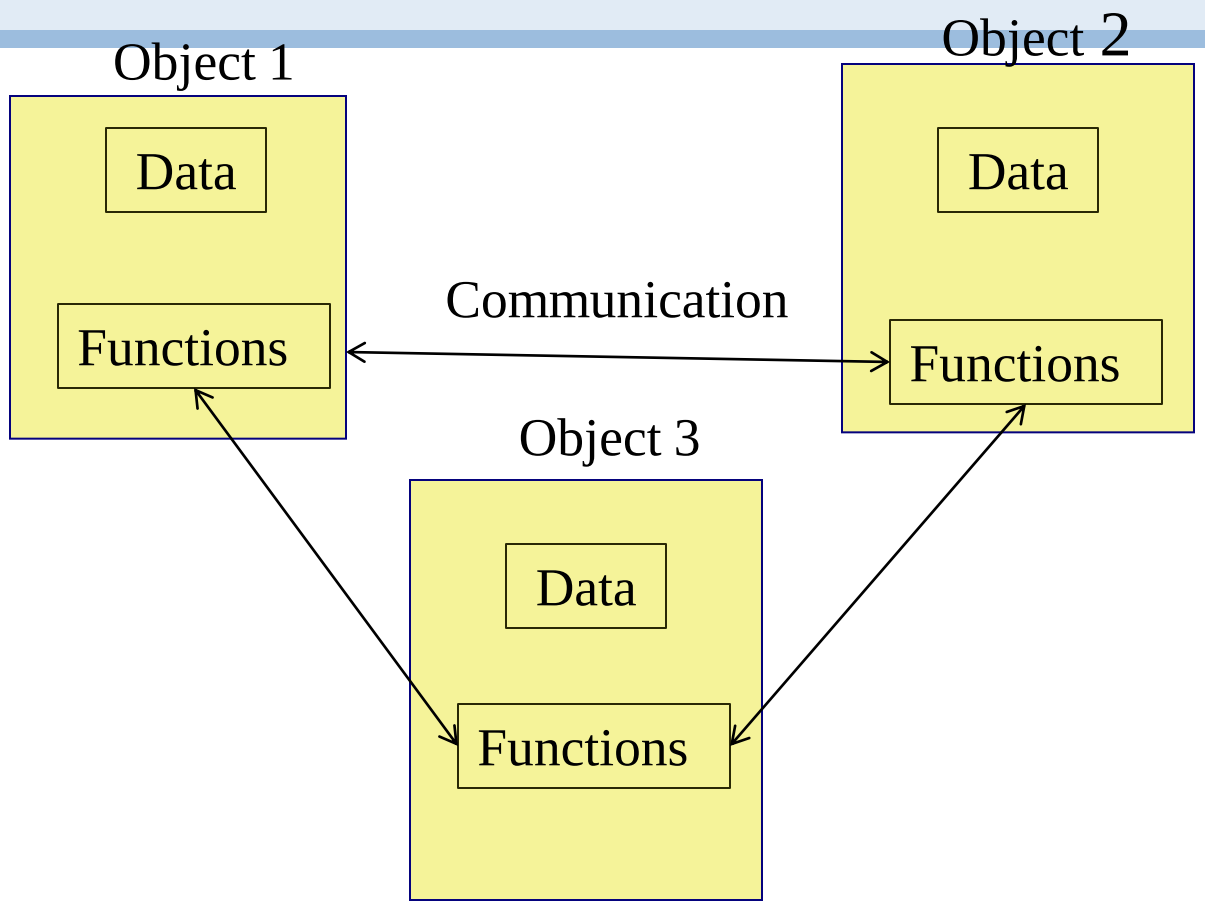
Example – Date is an Object Apprehended Intellectually

- **State (attributes)**
  - Year
  - Day
  - Month
- **behaviour (operations)**
  - Set Year
  - Set Day
  - Set Month
- **Identity**
  - Would have a unique ID in the model

OO Programming Concepts

- Object-oriented programming (OOP):
  - ✓Involves programming using ***objects***
  - ✓Treats data is as a critical element
  - ✓Does not allow data to flow freely around the system
  - ✓Ties data more closely to the functions that operate on it
  - ✓Protect accidental modification from outside functions
  - ✓Allows decomposition of problem into a number of entities (Objects)

Object-Oriented Concept



- Objects of the program interact by **sending messages to each other**
- Data of an object can be accessed only by the functions associated with that object
- Functions of an object can access the functions of others objects

Object-Oriented Concept

- An approach that provides a way of **modularizing programs** by *creating **partitioned memory area** for both data and functions that can be used as **templates** for creating copies of such modules **on demand*** **--**E Balagurusamy

- Objects are the partitioned computer memory area

Object-Oriented Programming

- Basic concepts of OOP:
  - **Objects**
  - Classes
  - Data Hiding
  - Encapsulation
  - Message Passing
  - Data Abstraction
  - Inheritance
  - Polymorphism
  - Dynamic Binding