TOPIC OF PRESENTATION

DEFINE OOP's CONCEPT

SUBJECT NAME : Object-Oriented Programming
SUBJECT CODE : BCAC301
STUDENT NAME : Chakra Sayan Roy
UNIV ROLL NO. : 22601221209
UNIV REG NO. : 212261001210192
DEPARTMENT : B.C.A
SEMESTER : 3rd

By: Chakra sayan Roy
Date: 25/07/2022

INTRODUCTION

- Object oriented programming is the principle of design and development of programs using modular approach.
  - Object oriented programming approach provides advantages in creation and development of software for real life application.
  - The basic element of object oriented programming is the data.
  - The programs are built by combining data and functions that operate on the data.
  - Some of the OOP's languages are C++, Java, C#, Smalltalk, Perl, and Python.

WHY DO WE NEED OBJECT-ORIENTED PROGRAMMING?

- Object-Oriented Programming was developed because limitations were discovered in earlier approaches to programming.
- To appreciate what OOP does, we need to understand what these limitations are and how they arose from traditional programming languages.

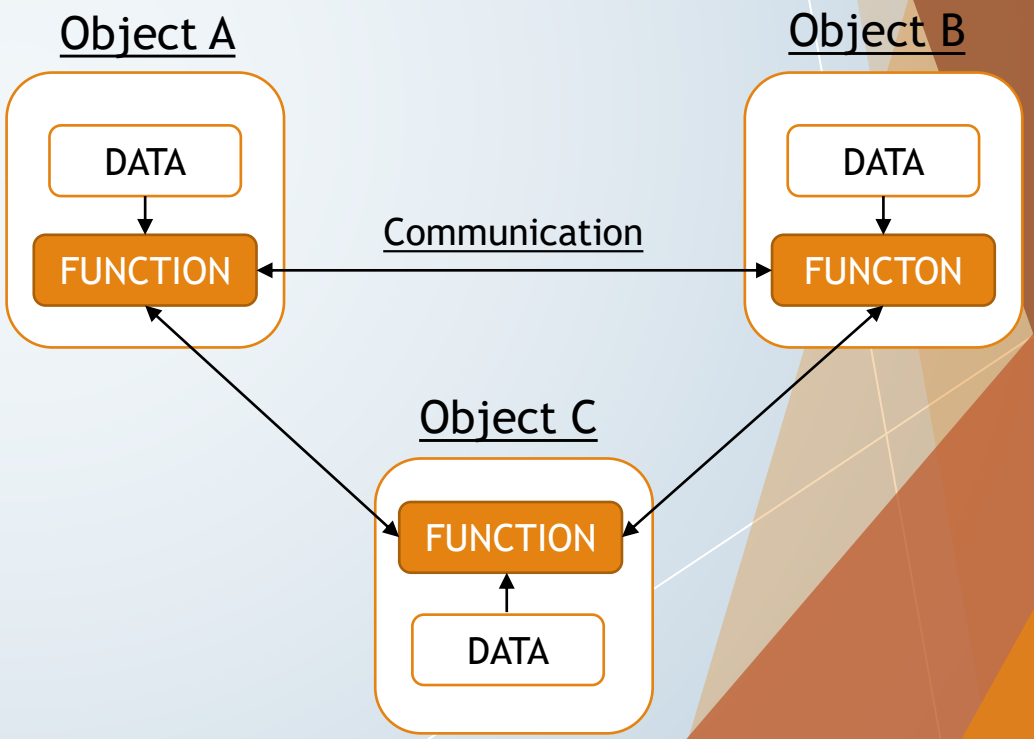
OBJECT-ORIENTED PROGRAMMING

- OOP was introduced to overcome flaws in the procedural approach to programming. Such as lack of reusability & maintainability.
- Fundamental idea behind object-oriented languages is to combine into a single unit both data and the functions that operate on that data. Such a unit is called an object.
- OOP follows bottom-up design technique.
- Class is the major concept that plays important role in this approach. Class is a template that represents a group of objects which share common properties and relationships.
- It ties the data more closely to the functions that operate on it, and protects it from accidental modification from the outside functions.
- Data of an object can be accessed only by the functions associated with that object and Communication of the objects also done through function.

OBJECT-ORIENTED PROGRAMMING

- Characteristics:
  - Emphasis on data rather than procedure.
  - Programs are divided into entities known as objects.
  - Data Structures are designed such that they characterize objects.
  - Functions that operate on data of an object are tied together in data structures.
  - Data is hidden and cannot be accessed by external functions.
  - Objects communicate with each other through functions.
  - New data and functions can be easily added whenever necessary.
  - Follows bottom up design in program design.

- OOP's Paradigm



BASIC CONCEPTS OF OOP's

- Objects: Objects are basically a thing which has existence and also have some properties and behaviors by which we can differentiate one object from an another object. It can be exist physically and logically. Ex:- Apple, Orange, Mango are the objects of class fruit.
- Class: Class is a collection of attributes and methods of similar type of object, which has same type of properties or, common characteristics. Class has no physical existence. Ex:- Planets are members of class of solar system.
- Data Hiding: Our internal data should not go out directly and outside person can't access our internal data directly. By using "private" modifier we can implement data hiding. The main advantage of data hiding is security.
- Data Abstraction: Data Abstraction refers to the process of representing most essential features to outside world, without including background details, or information. We can implement abstraction by using abstract class and interface.

BASIC CONCEPTS OF OOP's

- Data Encapsulation: Encapsulation is the first principle of OOP's. In simple words, binding data members(variables, properties) and member function(methods) into a single unit. And class is the best example of encapsulation.(Encapsulation= Abstraction + Data Hiding).
- Inheritance: The mechanism of deriving a new class from an old class is called inheritance, or derivation. The old class is known as base class and new class is known as derived, or sub class. From the help of inheritance we can reuse old code, or method.
- Polymorphism: In simple terms we can say it means ability to take more than one. In OOP's we preforms function overloading it means we performs different types of operation from same function. It depends on "data type", "sequence" and "number" of parameter, which is passed to the function. Ex:- Single function "printf()" draws different objects.

BASIC CONCEPTS OF OOP's

- Dynamic Binding: This is the process of linking of a function call to the actual code of the function at run time. In dynamic binding, the actual code to be executed is not known to the compiler until run-time. It also known as late binding.
  - For example, complier comes to know at runtime that which function of "sum()"will be call either with two arguments or with three arguments.
- Message Passing: In massage passing objects can communicate with each others by passing message same as people passing message with each other. Objects can send or receive message or information. Message passing involves specifying the name of the object, the name of the function (message) and the information to be sent.
  - For example, consider two classes Product and Order. The object of the Product class can communicate with the object of the Order class by sending a request for placing order.

BENEFITS OF OOP’s

- User can create new data type or users define data type by making class.
- Code can be reuse by using inheritance.
- Data can be hiding from outside world by using encapsulation.
- Operators or functions can be overloaded by using polymorphism, so same functions or operators can be used for multitasking.
- Object oriented system can be easily upgrade from small system to large system.
- It is easy to partition work for same project.
- Message passing techniques make communication easier.
- Software complexity can be easily managed.
- Maintenances cost is less.
- Simple to implement.

THANK YOU