Basic Concepts of OOP

Branch: S6 EC

Faculty: Sreedivya I

- Objects
- Classes
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Dynamic binding
- Message Passing

Objects

- Basic runtime entities in an object oriented system
- They may represent a person, a bank account , a table of data or any item that the program has to handle
- They represent a physical quantity
- During the execution of a program the objects interact by sending messages to each other
- Each object contains data and code to manipulate the data
- Example: Students in a classroom, Employees in a bank

Classes

- Collection of objects of similar type.
- User defined datatypes
- Class is a logical entity
- Objects are variables of the type class.
- Any number of objects can be created for a class.
- Example: Mango, Apple, Orange are members of class FRUITS

  Car, Bus, Train are members of class VEHICLE

Abstraction

- It refers to the act of representing essential features without including the background details or explanations
- That means only necessary details are exposed
- Example: When a key is pressed on the keyboard, the character appears on the screen, but the exact working may not be known to the user

Encapsulation

- Wrapping up of data and functions into a single unit
- Data is not accessible to the outside world
- The functions which are wrapped inside the class can access those data
- Data is made hidden
- Class is the best example of Encapsulation
- Example: To buy a medicine we go to medical store and ask chemist for the medicine. Only chemist has access to medicines. It reduces the risk of you taking any medicine that is not intended for you.

Inheritance

- It is the process by which objects of one class (derived class)acquire the properties of objects of another class(base class)
- Derived class has some common properties of the parent class
- Code reusability –additional features can be added to an existing class without modifying it
- Saves a lot of time in programming and reduce errors
- Increases the quality of work

- Example: Family tree

The features of children are inherited from Parents and grandparents. Similarly the parents inherit the properties from their parents. (Grandparents ,parents ,children)

- Types of Inheritance – Single, Hierarchical, Multiple, Multilevel and Hybrid

Polymorphism

- Ability to take more than one form
- An operation may exhibit different behaviors at different instances
- The behavior depends upon the types of data used in the operation
- The process of making a **single operator** to exhibit different behaviors on different instances is known as **operator overloading**
- The process of using a **single function name** to perform different types of tasks is known as **function overloading**
- **Overriding**- Both the parent class and derived class have the **same method name and arguments** but their functionality is different

- Example:

  1. Addition – If the operands are numbers then the result will be sum.
     If the operands are strings then the result is concatenation
     (Operator overloading)

  2. Drawing – If the problem is to draw then,
     for drawing a circle, function draw(int r) is used
     for drawing a rectangle, function draw(int l, int b) is used
     (function will be same, arguments will be different)

Dynamic Binding

- It refers to the linking of a procedure call to the code to be executed in response to the call.
- Also known as late binding as the code associated with a given procedure call is not known until the time of call at runtime

Message Passing

- Objects communicate with each other through message passing
- Objects can send or receive information
- Basic steps:

1. Creating classes that define objects and their behavior
2. Creating objects from class definition
3. Establishing communication between objects

Benefits of OOP

- Users can create new datatype by making class
- Code Reusability through Inheritance
- Data hiding which ensures data security through encapsulation
- Same functions or operators can be used for multitasking by overloading (Polymorphism)
- Can be used to build large or complex systems
- Message passing techniques makes communication easier
- It is easy to partition work for same project
- Software complexity can be easily managed

Application areas of OOP

- Real time systems
- Simulation and modelling
- Object oriented database
- Neural network and parallel programming
- Office automation system
- Client server system
- AI Expert system