**MODULE: 4 OOPS Concept**

**2. What is OOP? List OOP concepts.**

**Ans.** Object-Oriented Programming (OOP) is a programming paradigm that uses "objects" to design applications and computer programs. Objects are instances of classes, which can encapsulate data and functions that operate on the data. The goal of OOP is to increase the flexibility and maintainability of code by modeling real-world entities and relationships.

 **Class**: A blueprint for creating objects. A class defines a datatype by bundling data and methods that work on the data into one single unit.

 **Object**: An instance of a class. When a class is defined, no memory is allocated until an object of that class is created.

 **Encapsulation**: The bundling of data (attributes) and methods (functions) that operate on the data into a single unit or class. It also restricts direct access to some of the object's components, which is a means of preventing unintended interference and misuse of the methods and data.

 **Abstraction**: The concept of hiding the complex implementation details and showing only the necessary features of the object. This reduces complexity and increases efficiency.

 **Inheritance**: The mechanism by which one class (child class) can inherit the attributes and methods from another class (parent class). It promotes code reuse and establishes a natural hierarchy between classes.

 **Polymorphism**: The ability of different classes to be treated as instances of the same class through inheritance. Polymorphism allows methods to do different things based on the object it is acting upon, even though they share the same name.

 **Method Overloading**: The ability to define multiple methods with the same name but different parameters within the same class. This allows the same method to be used for different types of data input.

 **Method Overriding**: The ability of a subclass to provide a specific implementation of a method that is already defined in its superclass. This allows a subclass to offer specific behavior while maintaining the same method signature.

**3. What is the difference between OOP and POP?**

**Ans.** Object-Oriented Programming (OOP) and Procedural-Oriented Programming (POP) are two fundamental programming paradigms, each with its own approach to software development. Here are the main differences between OOP and POP:

### **Object-Oriented Programming (OOP)**

1. **Structure**: Programs are organized around objects, which are instances of classes. Classes encapsulate data and functions that operate on the data.
2. **Data and Functions**: Data and functions are bundled together as a single unit, or object. This encapsulation hides the internal state of the object from the outside world.
3. **Reusability**: Promotes code reuse through inheritance and polymorphism. New classes can be created based on existing ones, reducing code duplication.
4. **Abstraction**: Allows for abstraction through classes and objects. Details of implementation are hidden, exposing only what is necessary through interfaces and abstract classes.
5. **Modularity**: Objects are modular, meaning they can be developed, tested, and debugged independently.
6. **Example Languages**: Java, C++, Python, C#, Ruby.

### **Procedural-Oriented Programming (POP)**

1. **Structure**: Programs are organized around procedures or functions. The focus is on a sequence of tasks or operations to be performed.
2. **Data and Functions**: Data and functions are separate. Functions operate on data, but the data is not encapsulated within the functions.
3. **Reusability**: Code reuse is achieved through functions and procedures. There is no inherent support for inheritance or polymorphism.
4. **Abstraction**: Provides limited abstraction. Functions can be used to create reusable code blocks, but the separation of data and functions limits the level of abstraction.
5. **Modularity**: Achieved through functions and procedures, which can be written and tested independently. However, the separation of data and functions can make managing larger projects more complex.
6. **Example Languages**: C, Pascal, Fortran, COBOL.

### **Key Differences**

1. **Focus**:
   * OOP focuses on objects that represent real-world entities.
   * POP focuses on procedures and the sequence of steps to perform tasks.
2. **Data Handling**:
   * OOP uses encapsulation to bundle data and methods.
   * POP keeps data and functions separate.
3. **Code Reuse**:
   * OOP supports inheritance and polymorphism for code reuse.
   * POP relies on function calls for code reuse, without built-in support for inheritance.
4. **Abstraction**:
   * OOP provides higher levels of abstraction through classes and objects.
   * POP provides limited abstraction, mainly through functions.
5. **Modularity**:
   * OOP inherently supports modularity through objects.
   * POP achieves modularity through functions, which can be less effective for large-scale systems.