Module 4 :

2.What is OOP? List OOP concepts.

Ans.-> Object-Oriented Programming (OOP) is a programming paradigm that uses objects and classes to structure software in a way that models real-world entities and their interactions. It focuses on using objects, which are instances of classes, to represent data and methods that operate on that data. OOP is designed to increase code reusability, scalability, and maintainability.

In C++, the primary OOP concepts are:

1. **Classes and Objects**:
    - **Class**: A blueprint for creating objects. It defines a type of object according to the data members (attributes) and member functions (methods).
    - **Object**: An instance of a class. It represents a specific entity that has state (attributes) and behavior (methods).
2. **Encapsulation**:
    - Encapsulation is the concept of bundling data (attributes) and methods (functions) that operate on the data into a single unit, i.e., a class. It restricts direct access to some of the object's components, which is a means of preventing unintended interference and misuse.
3. **Abstraction**:
    - Abstraction involves simplifying complex systems by modeling classes appropriate to the problem, reducing complexity by hiding unnecessary details and showing only the essential features of the object.
4. **Inheritance**:
    - Inheritance is the mechanism by which one class (derived class or child class) can inherit the properties and behavior (methods) of another class (base class or parent class). It promotes code reuse and establishes a hierarchical relationship between classes.
5. **Polymorphism**:
    - Polymorphism allows methods to do different things based on the object it is acting upon, even if they share the same name. It is typically achieved through method overriding (runtime polymorphism) and method overloading (compile-time polymorphism).

3.What is the difference between OOP and POP?

Ans-> Object-Oriented Programming (OOP) and Procedural-Oriented Programming (POP) are two different programming paradigms used to structure and develop software. Here are the key differences between them:

Object-Oriented Programming (OOP)

1. **Basic Concept**:
    - OOP is based on the concept of objects and classes. It models real-world entities as objects, which combine both data and behavior.
2. **Structure**:
    - Programs are organized around objects and classes.
    - Each object is an instance of a class and encapsulates data and methods.
3. **Data Handling**:
    - Data and functions are bundled together in the form of objects.
    - Emphasizes data security and hides data through encapsulation.
4. **Key Concepts**:
    - Includes concepts like inheritance, polymorphism, encapsulation, and abstraction.
5. **Reusability**:
    - Promotes code reuse through inheritance and polymorphism.
    - Objects and classes can be reused across different programs.
6. **Data Access**:
    - Data is accessed through methods, maintaining a higher level of security and control.

Procedural-Oriented Programming (POP)

1. **Basic Concept**:
    - POP is based on the concept of procedure calls. It is centered around procedures or functions.
2. **Structure**:
    - Programs are organized as a set of functions or procedures.
    - Functions operate on data passed to them as parameters.
3. **Data Handling**:
    - Data is often passed between functions as parameters.
    - Data and functions are separate entities.
4. **Key Concepts**:
    - Includes concepts like sequence, selection, and iteration (loops).
5. **Reusability**:
    - Code reuse is limited to function calls.
    - Functions can be reused, but the data they operate on is not encapsulated.
6. **Data Access**:
    - Data can be freely accessed and modified by any function, leading to potential security issues.