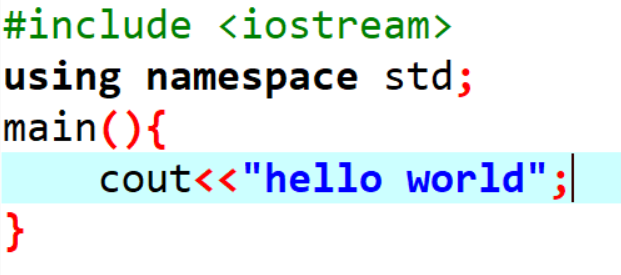
**MODULE: 4**

**OOPS Concept**

**Basic Concepts of OOP**

1. **WAP to print “Hello World” using C++**

**Ans:**

****

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

**Ans**: OOP stands for Object-Oriented Programming. It's a programming paradigm that revolves around the concept of "objects," which can contain data in the form of fields (attributes or properties), and code in the form of procedures (methods or functions). Here are some key concepts of OOP:

1. Class: A blueprint or template for creating objects. It defines the properties and behaviors common to all objects of the class.

2. Object: An instance of a class. It is a concrete realization of the class blueprint, with its own state (attributes) and behavior (methods).

3. Encapsulation: The bundling of data (attributes) and methods (functions) that operate on the data into a single unit (object). It allows data hiding, protecting the internal state of an object from direct access and modification by external code.

4. Inheritance: A mechanism that allows a class (subclass or derived class) to inherit properties and behaviors from another class (superclass or base class). It promotes code reusability and establishes a hierarchical relationship between classes.

5. Polymorphism: The ability of objects of different classes to be treated as objects of a common superclass. It allows methods to behave differently based on the object they are called on, enabling flexibility and extensibility in code design.

6. Abstraction: The process of simplifying complex systems by focusing on the essential characteristics while hiding unnecessary details. In OOP, abstraction is achieved through abstract classes and interfaces, providing a blueprint for subclasses to implement.

7. Modularity: The subdivision of a program into separate, interchangeable components (modules), each responsible for a specific aspect of functionality. In OOP, classes and objects promote modularity by encapsulating related data and behavior into cohesive units.

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

**Ans:**OOP (Object-Oriented Programming) and POP (Procedural-Oriented Programming) are two different programming paradigms with distinct approaches to software design and development. Here are some key differences between the two:

1. Focus:

- OOP focuses on organizing code into objects, which encapsulate data and behavior. It emphasizes modeling real-world entities and their interactions.

- POP focuses on organizing code into procedures or functions, which manipulate data. It emphasizes sequential execution of instructions and is often considered more linear in nature.

2. Data and Behavior:

- In OOP, data (attributes or properties) and behavior (methods or functions) are encapsulated within objects. Objects communicate with each other through message passing, and data hiding ensures the internal state of an object is protected.

- In POP, data is separate from the functions that operate on it. Functions manipulate data stored in variables, and there is typically less emphasis on data hiding or encapsulation.

3. Code Reusability:

- OOP promotes code reusability through concepts like inheritance, where classes can inherit properties and behaviors from other classes. This enables the creation of hierarchies of related classes and facilitates code sharing.

- POP relies more on procedural decomposition, breaking down a problem into smaller procedures or functions. While functions can be reused, the focus is generally on breaking down tasks into smaller, manageable units rather than creating reusable components.

4. Complexity Management:

- OOP provides mechanisms such as encapsulation, abstraction, and polymorphism to manage complexity by modeling real-world entities in a more natural way. This can lead to more modular and maintainable code.

- POP tends to be simpler in structure and may be more suitable for smaller, straightforward programs. However, as programs grow larger and more complex, managing code becomes more challenging in a procedural paradigm.

5. Examples:

- Examples of languages that follow OOP principles include Java, C++, Python, and C#.

- Examples of languages that follow POP principles include C, Pascal, and early versions of BASIC.