**####Basic Concept Of Oops####**

**(Q 1) WAP to print “Hello World” using C++**

**#include <iostream>**

**int main() {**

**cout << "Hello World" ;**

**return 0;**

**}**

**(Q 2)** **What is OOP? List OOP concepts** .

OOP stands for Object-Oriented Programming. It's a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods or functions).

Here are some key concepts of Object-Oriented Programming:

1. **Class**: A class is a blueprint for creating objects. It defines the attributes and methods that all objects of that type will have.
2. **Object**: An object is an instance of a class. It is a concrete entity that contains data and methods defined in its class.
3. **Encapsulation**: Encapsulation is the bundling of data (attributes) and methods that operate on the data into a single unit or class. It hides the internal state of an object from the outside world and only exposes a controlled interface to interact with the object.
4. **Inheritance**: Inheritance is a mechanism by which a class can inherit properties and behavior from another class, called the superclass or base class. This allows for code reuse and the creation of a hierarchy of classes.
5. **Polymorphism**: Polymorphism means "many forms." It allows objects of different classes to be treated as objects of a common superclass. Polymorphism is achieved through method overriding and method overloading.
6. **Abstraction**: Abstraction is the process of hiding the complex implementation details and showing only the necessary features of an object to the outside world. It helps in reducing programming complexity and managing large codebases.
7. **Message Passing**: Objects in OOP communicate with each other by sending and receiving messages. This is typically done by calling methods on objects.

These concepts together help in creating modular, maintainable, and scalable software systems by promoting code reuse, reducing code duplication, and enhancing code organization and understandability.

**(Q 3) What is the difference between OOP and POP?**

OOP (Object-Oriented Programming) and POP (Procedural-Oriented Programming) are two different programming paradigms, each with its own approach to organizing and structuring code.

Here are the key differences between OOP and POP:

1. **Abstraction and Encapsulation**:
   * OOP emphasizes the use of objects, which encapsulate both data and behavior (methods) within a single unit. Objects hide their internal details and provide a clear interface for interacting with the outside world.
   * POP, on the other hand, primarily focuses on procedures or functions that operate on data. While data and functions can be logically grouped together in modules, the emphasis is less on encapsulation and more on organizing code around procedures.
2. **Data and Behavior**:
   * In OOP, data and behavior are closely tied together within objects. Objects contain both data (attributes) and methods (functions) that operate on that data.
   * In POP, data and behavior are often separate. Data is passed around between functions as parameters, and functions manipulate that data.
3. **Inheritance and Polymorphism**:
   * OOP allows for the concepts of inheritance and polymorphism, where classes can inherit attributes and methods from parent classes and objects of different classes can be treated interchangeably if they share a common interface.
   * POP typically does not have built-in support for inheritance or polymorphism. Code reuse is often achieved through functions and modules.
4. **Code Organization**:
   * OOP promotes organizing code around objects and classes, which can lead to a more modular and organized codebase, especially for large-scale projects.
   * POP organizes code around procedures and functions, which may be suitable for smaller projects or scripts but can become unwieldy for larger applications.
5. **Example Languages**:
   * OOP is associated with languages like C++, Java, Python, and C#.
   * POP is associated with languages like C, Pascal, and early versions of BASIC.

In summary, while both paradigms have their own merits and use cases, OOP provides more advanced features for code organization, reuse, and modularity through the use of objects and classes, whereas POP focuses more on procedures and functions for achieving similar goals.

Top of Form