**Q1WAP to print “Hello World” using C++**

ANS-

#include <iostream>

using namespace std

int main() {

cout << "Hello World" << endl;

}

**Q2 What is OOP? List OOP concepts**

ANS-

Object-Oriented Programming (OOP) is a programming paradigm that is 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). It focuses on organizing software design around objects rather than functions or logic.

Key Concepts of OOP:

1. Class:
   * A blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that the objects created from the class will have.
   * Example: class Car { ... };
2. Object:
   * An instance of a class. Once a class is defined, you can create objects of that class, each having its own set of attributes and behaviors.
   * Example: Car myCar;
3. Encapsulation:
   * The bundling of data (attributes) and methods (functions) that operate on the data into a single unit or class.
   * It also hides the internal state of the object and requires all interaction to be performed through an object's methods.
   * Example: Making a property private and exposing it via getter and setter methods.
4. Abstraction:
   * Hiding the complex implementation details of an object and exposing only the essential features.
   * This allows the user to focus on interactions with the object, rather than on the intricate workings of how it operates.
   * Example: A Car class might expose methods like start() and stop(), but internally it handles how the car starts and stops.
5. Inheritance:
   * A mechanism that allows a new class (called the subclass or derived class) to inherit properties and methods from an existing class (called the superclass or base class).
   * This allows for code reuse and the creation of hierarchical relationships.
   * Example: A SportsCar class inheriting from the Car class.
6. Polymorphism:
   * The ability to take many forms. Polymorphism allows methods to do different things based on the object they are acting upon.
   * There are two main types:
     + Method Overloading: The ability to define multiple methods with the same name but different parameters.
     + Method Overriding: The ability to redefine a method in a subclass that was already defined in the superclass.
   * Example: A method draw() can behave differently depending on whether it’s called on a Circle object or a Rectangle object.
7. Composition:
   * A concept where one object contains another object as a part of its state, rather than inheriting from it.
   * It's often used as an alternative to inheritance, promoting greater flexibility.
   * Example: A Car class may have an instance of an Engine class inside it.

**Q3What is the difference between OOP and POP?**

**ANS-**

The main difference between **Object-Oriented Programming (OOP)** and **Procedure-Oriented Programming (POP)** lies in the way the software design and programming tasks are approached. While both are programming paradigms, they have fundamental differences in structure, methodology, and focus.

* **OOP** is a more modern and structured way to program, particularly useful for large-scale, complex applications.
* **POP** is simpler and more straightforward, but less flexible for large or complex projects because it doesn't focus on data encapsulation or modularity as OOP does.

OOP is generally preferred for its ability to handle large, complex systems, while POP may still be used in situations where simplicity and performance are the primary goals