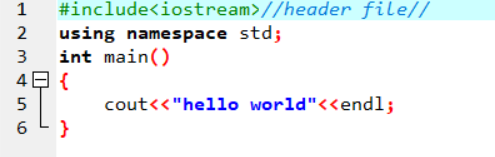
MODULE 4.1 -C++

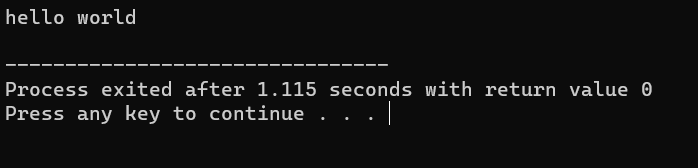
Question 1: WAP to print “Hello World” using C++.

Answer:

Code-



Output-



Question 2: What is OOP? List OOP concepts.

Answer:

Object-Oriented Programming or OOPs refers to languages that use objects in programming. Object-oriented programming aims to implement real-world entities.

**OOPs Concepts:**

* + Class
  + Objects
  + Data Abstraction
  + Encapsulation
  + Inheritance
  + Polymorphism

Question 3: What is the difference between OOP and POP?

Answer:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Object-Oriented Programming (C++)** | **Procedural Programming (c language)** |
| Key Focus | Object and their interactions | Procedures and functions |
| Data Management | Data and behaviour are encapsulated in objects | Data and behaviour are separate entities |
| Abstraction | Encourages the use of abstract classes and interfaces | Does not emphasize abstraction |
| Inheritance | Supports inheritance, allowing classes to inherit properties and methods | Does not support inheritance |
| Polymorphism | Allows objects of different types to be treated as the same type | Does not provide inherent polymorphism |
| Code Reusability | High level of code reusability through inheritance and composition | Relies on functions and subroutines for code reusability |
| Code Organization | Follows a modular approach, with objects as self-contained modules | Relies on functions and procedures for code organization |
| Flexibility | Provides flexibility through polymorphism and dynamic binding | Relies on structured programming, offering less flexibility |
| Complexity Management | Encourages managing complexity through encapsulation and abstraction | Manages complexity through modular code organization and stepwise design |
| Real-World Modelling | Well-suited for modelling real-world entities and their interactions | May not align well with real-world modelling, focusing more on processes |