Q :2 WHAT IS OOP ? FEATURES OF OOP

ANS :-



Introduction

OOP IS object oriented programing language

Cpp is on of the worlde’s most popular programing language which is the also support pop with oop

It is built by bjane strouster in 1979

FEATURE OF OOP



1. CLASS
2. OBJECT
3. INHERITENCE

* SINGLE
* MULTIPLE
* MULTILAVAL
* HYBRID
* HERIACHICAL

1. ENCAPSULATION
2. POLYMORPHIISM

* Method overloading
* Method overriding

1. MODIFIRE
2. CLASS

It’s collection of data member and member function

Class without not support oop

1. Object

It’s intctant of class

It use this class so I need class object.

1. Inheritance

Using more than one class is called inheritance

1. Single

Information of one class is shared with another class called single class

1. Multiple

Sharing a property of one class with another class is called a multiplayer class

c) multilevel

Giving information of more than one class to a single class is called multi-level class

d) heriachical

one to main it is called heriachical

E) hybrid

Single and multiple combined shear to property it is called hybrid

1. Encapsulation

It’s raping of data it is called encapsulation

Its automating setup to language

It is also called data binding

4)polymorphism

One name and multiple form

1. Method overloding
2. Method overriding

a) Method overloding

Same name and argument

b)Method overriding

Same name but different argument.

1. Modifier

There are three modifier

* Public
* Private (default)
* Protected - limited acess

Q 3:

WHAT IS DIFFERENT BY OOP AND POP ?

Object-Oriented Programming (OOP) and Procedural-Oriented Programming (POP) are two fundamental programming paradigms, each with its own principles and approaches to software design and development. Here are the key differences between OOP and POP:

**1. Conceptual Differences:**

**OOP (Object-Oriented Programming):**

* **Core Principle:** Based on the concept of "objects" which are instances of classes. Classes define the blueprint for objects.
* **Modularity:** Promotes modularity and reusability through encapsulation.
* **Encapsulation:** Data and methods that operate on the data are bundled together.
* **Abstraction:** Allows for complex reality representation by defining abstract data types.
* **Inheritance:** Supports inheritance, enabling new classes to inherit properties and behaviors from existing classes.
* **Polymorphism:** Allows objects to be treated as instances of their parent class rather than their actual class.
* **Examples:** C++, Java, Python, C#, Ruby.

**POP (Procedural-Oriented Programming):**

* **Core Principle:** Based on the concept of procedure calls (also known as routines, subroutines, or functions).
* **Modularity:** Achieved through the use of procedures or functions, making the code more structured and manageable.
* **Sequential Execution:** Emphasizes a sequence of computational steps to be carried out.
* **Global Data:** Uses global data which can be accessed by any function in the program.
* **Examples:** C, Pascal, Fortran, Basic.

**2. Design Approach:**

**OOP:**

* **Design:** Focuses on the entities involved and their interactions. Design is centered around objects and their relationships.
* **Development:** Objects are created as per the blueprint defined by their respective classes.

**POP:**

* **Design:** Focuses on the sequence of tasks to be performed. The design is centered around functions and procedures.
* **Development:** Functions are defined, and then the program is built using these functions to perform tasks sequentially.

**3. Data Handling:**

**OOP:**

* **Data:** Encapsulated within objects. Data hiding is a key concept.
* **Access:** Through methods (functions) defined in the class.
* **State Management:** Each object maintains its own state.

**POP:**

* **Data:** Typically stored in variables and passed to functions as arguments.
* **Access:** Directly through functions. Functions can access and