Theory Question

27\_jan\_se NAME: Vrinda Mavadhiya

**1.**Explain the key concepts of Object-Oriented Programming (OOP)

OOP is a programming paradigm based on the concept of **objects**. The four main pillars of OOP are:

**Encapsulation**

Wrapping data and functions into a single unit (class).

Access is controlled using access specifiers: private, public, and protected.

**Abstraction**

Hiding complex implementation details and showing only essential features.

Achieved using abstract classes and interfaces.

**Inheritance**

One class (child) can inherit properties and behaviors from another class (parent).

Promotes code reuse.

**Polymorphism**

Same function or operator behaves differently based on the context.

Types: Compile-time (function overloading, operator overloading) and Runtime (virtual functions).

## 2. What are Classes and Objects in C++?

* **Class**: A blueprint for creating objects. It defines data and methods.
* **Object**: An instance of a class.

**Example:**

#include <iostream>

using namespace std;

class Car

{

public:

string brand;

int speed;

void showDetails()

{

cout << "Brand: " << brand << ", Speed: " << speed << " km/h" << endl;

}

};

main()

{

Car myCar; // Object created

myCar.brand = "Toyota";

myCar.speed = 120;

myCar.showDetails();

}

## 3. What is Inheritance in C++? Explain with an Example.

**Inheritance** allows a class (child/derived) to acquire properties and methods of another class (parent/base).

**Example:**

#include <iostream>

using namespace std;

class Animal

{

public:

void eat()

{

cout << "This animal eats food." << endl;

}

};

class Dog : public Animal { // Inheriting from Animal

public:

void bark() {

cout << "The dog barks." << endl;

}

};

main() {

Dog d;

d.eat(); // Inherited from Animal

d.bark(); // Dog's own method;

}

## 4. What is Encapsulation in C++? How is it achieved in classes?

**Encapsulation** is the process of **binding data and methods** that operate on the data into a single unit (class), and **restricting direct access** to some of the object's components.

**Achieved by:**

* Declaring class variables as private
* Using public methods (getters/setters) to access and update them

**Example:**

#include <iostream>

using namespace std;

class Employee

{

private:

int salary; // Private variable

public:

void setSalary(int s)

{

salary = s;

}

int getSalary()

{

return salary;

}

};

main()

{

Employee emp;

emp.setSalary(50000);

cout << "Salary: " << emp.getSalary();

}