**Question 1: Virtual Function Behavior**

Write a C++ program with a base class Shape that has a virtual function area(). Derive two classes Circle and Rectangle, override the area() function in both, and demonstrate **runtime polymorphism** by storing objects in **a single base class pointer** and calling area().

**Task:**

* Create a base class pointer (Shape\*).
* Point it to a Circle object, call area().
* Then point it to a Rectangle object, call area().
* Show that the correct function is called at runtime.

**Question 2: Pure Virtual Function and Abstract Class**

Create an abstract class Employee with a pure virtual function calculateSalary(). Derive two classes FullTimeEmployee and PartTimeEmployee that implement calculateSalary() differently.

**Task:**

* Implement both derived classes.
* Use base class pointer to call calculateSalary() for both types of employees.
* Show that you **cannot create an object of Employee directly**.

### ****Question 3: Overriding vs Non-Virtual Function****

Create a base class Vehicle with two functions:

1. A **virtual function** displayType()
2. A **non-virtual function** showWheels()

Derive a class Car from Vehicle and override both functions.

**Task:**

* Call both functions using a Vehicle\* pointer pointing to a Car object.
* Show the difference in output between calling a **virtual function** and a **non-virtual function** through base class pointer.