Lab 5

Objective: To familiarise oneself with virtual function, polymorphism, and miscellaneous

Questions

1. Write a C++ program that demonstrates the use of virtual functions. Create a base class Shape with a member function area() that is declared as a virtual function. Derive two classes, Rectangle and Circle, from the Shape class. Override the area() function in both derived classes to calculate and return the area of the rectangle and the circle, respectively.

   In the main function, create pointers of type Shape* and assign them to objects of Rectangle and Circle. Use the base class pointer to call the overridden area() function and display the areas of both shapes.

2. Write a C++ program that demonstrates the use of abstract classes and pure virtual functions. Create an abstract base class Employee with the following:
   a. Data members:
      i. name (string)
      ii. id (integer)
      iii. salary (float)
   b. A pure virtual function, calculateSalary(), to calculate the salary
3. Create a C++ program to demonstrate the use of virtual destructors and virtual base classes in an inheritance hierarchy.
   a. Create an abstract base class Shape with:
      i. A data member for storing the colour of the shape.
      ii. A pure virtual function area() to calculate the area of the shape.
      iii. A virtual destructor.
   b. Derive two classes from Shape:
      i. Rectangle: Inherits from Shape, adds data members for width and height, and implements the area() function to calculate the area as width * height.
      ii. Circle: Inherits from Shape, adds a data member for the radius, and implements the area() function to calculate the area as π * radius^2.
   c. Create a virtual base class, Printable, that contains a virtual function print() to print the shape's colour. Both Rectangle and Circle should inherit from Printable (as a virtual base class) along with Shape.
   d. In the main() function:

i. Create pointers of type Shape* and Printable*.
ii. Assign them to objects of Rectangle and Circle.
iii. Call the area() function and the print() function.
iv. Ensure proper cleanup by utilising the virtual destructor to avoid memory leaks when deleting derived class objects through base class pointers.