

## **Lab Sheet – 2**

### **Title: Implementing Basic OOP Features**

#### **Part - 1**

1. Create a simple class named Person that contains basic information like name, age, gender, etc. Your class should also contain functions/methods for storing and displaying data.
2. Write a program to define a class named Box which has data **length**, **breadth** and **height** and public functions **ReadData()** for reading data members and **Volume()** to calculate volume of box.
3. Write a program which has class **Book** with data members **book\_name**, **ISBN**, **author** & **price** and appropriate function members to read and display data members.
4. Write a program to define a class **Circle** with its data members **pi** and **r** and members functions **getdata()** for initializing data members and **calculate()** for finding area of ac circle. Return result from **calculate()** and display result in **main()** function. Use **pi** as constant.
5. Define a class **Rectangle** with data members: **length** and **breadth**. Initialize its data members with some fixed values (i.e. say 100 and 200 for length and breadth respectively) using a constructor. Write a program to use an object of the class to calculate area of a rectangle.
6. Modify above program to read **length** and **breadth** of a rectangle in **main()** function and supply them in parameterized constructor to initialize its data members.
7. Write a program to demonstrate static constructor.
8. Write programs to show all usage of this keyword.  
**(You must write separate programs for each usage)**
9. Write a program to demonstrate properties.
10. Write a program to calculate sum and product of two numbers using automatic property.
11. Write a C# program to show the usage of **abstract class and methods**.
12. Write a C# program to show the usage of **sealed class**.
13. Write a C# program to demonstrate **method overloading**.
14. Write a C# program to demonstrate **method overriding using virtual method**.
15. Write programs to demonstrate all types of inheritance.  
**(You must write separate programs for each type)**
16. Write a program to calculate area and perimeter of rectangle using **interface**.
17. Write a C# program to demonstrate the use of **“base”** keyword in inheritance.

## Part – 2

18. Write a program to demonstrate **indexers**.
19. Write a program to demonstrate **struct**.
20. Write a program to demonstrate **enum**.
21. Write a program to demonstrate all types of **generic collections**.  
(You must write separate programs for each type)
22. Create a class Polygon with data members: dimension1 and dimension2 and a member function: ReadDimension () to read data members. Derive two classes Rectangle and Triangle from Polygon class with appropriate member function to calculate area of each rectangle and triangle .
23. Create a class Polygon with data members to represent two dimensions and parameterized constructor to initialize data members. Derive two classes Rectangle and Triangle from Polygon class with appropriate member function to calculate area of each rectangle and triangle.
24. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.
25. Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.
26. Create a class named 'Member' having the following members:  
Data members
  - 1 - Name
  - 2 - Age
  - 3 - Phone number
  - 4 - Address
  - 5 – Salary

It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.