INDEX

Sr.No	Name of Assignment
1.	Classes and object Design a class 'Complex'with data members for real and imaginary part. Provide default and Parameterized constructors. Write a program to perform arithmetic operations of two complex numbers.
2.	Polymorphism Identify commonalities and differences between Publication, Book and Magazine classes. Title, Price, Copies are common instance variables and saleCopy is common method. The differences are, Bookclass has author and order Copies(). Magazine Class has methods orderQty, Currentissue, receiveissue().Write a program to find how many copies of the given books are ordered and display total sale of publication.
3.	Inheritance Design and develop inheritance for a given case study, identify objects and relationships and implement inheritance wherever applicable. Employee class has Emp_name, Emp_id, Address, Mail_id, and Mobile_no as members. Inherit the classes: Programmer, Team Lead, Assistant Project Manager and Project Manager from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.
4.	Dynamic Binding Design a base class shape with two double type values and member functions to input the data and compute_area() for calculating area of shape. Derive two classes: triangle and rectangle. Make compute_area() as abstract function and redefine this function in the derived class to suit their requirements. Write a program that accepts dimensions of triangle/rectangle and display calculated area. Implement dynamic binding for given case study.
5.	Interface Design and develop a context for given case study and implement an interface for Vehicles Consider the example of vehicles like bicycle, car and bike. All Vehicles have common functionalities such as Gear Change, Speed up and apply breaks. Make an interface and put all these common functionalities. Bicycle, Bike, Car classes should be implemented for all these functionalities in their own class in their own way
6.	Exception handling Implement a program to handle Arithmetic exception, Array Index Out of Bounds. The user enters two numbers Num1 and Num2. The division of Num1 and Num2 is displayed. If Num1 and Num2 are not integers, the program would throw a Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception. Display the exception.

7.	Template Implement a generic program using any collection class to count the number of elements in a collection that have a specific property such as even numbers, odd number, prime number and palindromes.
8.	File Handling Implement a program for maintaining a database of student records using Files. Student has Student_id,name,Roll_no, Class, marks and address. Display the data for few students. i) Create Database ii)Display Database iii) Delete Records iv) Update Record v)Search Record
9.	Case Study: Using concepts of Object Oriented programming develop solution for any one application 1) Banking system having following operations : 1. Create an account 2. Deposit money 3. Withdraw money 4. Honor daily withdrawal limit 5. Check the balance 6. Display Account information. 2) Inventory management system having following operations : 1. List of all products 2. Display individual product information 3. Purchase 4. Shipping 5. Balance stock 6. Loss and Profit calculation.
10.	Factory Design Pattern Implement Factory design pattern for the given context. Consider Car building process, which requires many steps from allocating accessories to final makeup. These steps should be written as methods and should be called while creating an instance of a specific car type. Hatchback, Sedan, SUV could be the subclasses of Car class. Car class and its subclasses, CarFactory and TestFactoryPattern should be implemented.
11.	11. Strategy Design Pattern Implement and apply Strategy Design pattern for simple Shopping Cart where three payment strategies are used such as Credit Card, PayPal, BitCoin. Create an interface for strategy pattern and give concrete implementation for payment.