1. Write Simple Hello World program using C#.

2. Create a basic four-function calculator. Take two numbers input from command line along with the operation to be performed. Use switch case.

3. Create a basic four-function calculator. Take two numbers input from console along with the operation to be performed. Make a menu driven program, which will continue to execute according to user choice. Use do-while loop.

4. Write a structure Student with following members:

   string name; bool gender; int age;
   int std; char div; double marks;

   Provide following functionality:

   Default constructor(?)
   Parameterized constructor
   Get and Set methods
   AcceptDetails() method to accept data from console.
   PrintDetails() method to print data to console.

   Check the default access specifier for members of structure.
   Keep all data members private and all member functions public.

5. Create a reference of Student's array in Main. Write a function CreateArray() to allocate the array as per user's requirement. Pass it to a function AcceptInfo() to accept student information. Print the array using PrintInfo() function. Also write a function ReverseArray() that reverse the array into a new array. Reprint the modified array in Main.

   Note: Do not return array from the function. All four functions mentioned above should be static functions in Main()'s class.

6. Write all following classes in a class library "EmplyoeeLib". (Question No 7 to 13)

7. Write class Date with following members:

   int day; int month; int year;

   Provide following functionality:

   Default constructor
   Parameterized constructor
   Properties: Day, Month and Year
   AcceptDate() method to accept data from console.
   PrintDate() method to print data to console.
   bool IsValid(); method to check validity of date.
   string ToString(); method to return data of object in string format.

   Also provide a static member function that returns difference between two date objects in number of years.
   Overload "-" operator to perform the same job.

8. Write a class Person with following members:

   string name; bool gender; Date birth;
   string address;

Provide following functionality:

Default constructor
Parameterized constructor
Properties: Name, Gender, Birth, Address and Age(Read Only)
Accept() method to accept data from console.
Print() method to print data to console.
string ToString(); method to return data of object in string format.

Hint: Class will contain Date object for birth and Age will be calculculated using static method in Date class.

9. Write a class Employee with following members:

    int id; (Auto Generated)
    double salary; string designation;
    enum DepartmentType dept;

    Provide following functionality:

    Default constructor
    Parameterized constructor
    Properties: Id, Salary, Designation, Dept
    Accept() method to accept data from console.
    Print() method to print data to console.
    string ToString(); method to return data of object in string format.

    Hint: class must be inherited from Person class. Use static count for auto generating id.

10. Write a class Manager with following members:

    double bonus;

    Provide following functionality:

    Default constructor
    Parameterized constructor
    Properties: Bonus
    Accept() method to accept data from console.
    Print() method to print data to console.
    string ToString(); method to return data of object in string format.

    Hint: class must be inherited from Employee class so that designation is fixed i.e. “Manager”.

11. Write a class Supervisor with following members:

    int subbordinates; (Number of assistants)

    Provide following functionality:

    Default constructor
    Parameterized constructor
    Properties: Subbordinates
    Accept() method to accept data from console.
    Print() method to print data to console.
    string ToString(); method to return data of object in string format.

Hint: class must be inherited from Employee class so that designation is fixed i.e. “Supervisor”.

12. Write a class WageEmp (Contract basis employee) with following members:
    int hours; int rate; (Per hour basis)

    Provide following functionality:

    Default constructor
    Parameterized constructor
    Properties: Hours, Rate
    Accept() method to accept data from console.
    Print() method to print data to console.
    string ToString(); method to return data of object in string format.

    Hint: class must be inherited from Employee class so that designation is fixed i.e. “Wage”.

13. Write class Company with following members:
    string name; string address;
    LinkedList empList;
    double salaryExpense; (Per month)

    Provide following functionality:

    Default constructor
    Parameterized constructor
    Properties: Name, Address, EmpList, SalaryExpense.
    Accept () method to accept data from console.
    Print () method to print data to console.
    CalculateSalaryExpense() method to calculate salary expense per month.
    public void AddEmployee(Employee e);
    public bool RemoveEmployee(int id);
    public LinkedListNode<Employee> FindEmployee(int id);
    public override string ToString();
    public void PrintEmployees();

14. Create a console application and use the library “EmployeeLib” as private assembly. Console application must be a menu drive program providing facilities of adding employee to company, removing employee, find employee by id, display company info and display all employees.

15. Create a similar console application and use the library “EmployeeLib” as shared assembly.

can use serialization for 16.

16. Modify EmployeeLib assembly to provide serialization support. Change the version number of this assembly and re-deploy it into GAC. In menu driven console application add two menus for serialization and deserialization. Use BinaryFormatter class.

use Linked list for 17

17. Modify Company class to use ArrayList of employees rather that LinkedList. Also provide support for SoapFormation. In menu driven console application add two menus for serialization and deserialization. Use SoapFormatter class.

18. Create a static Math class using Sum, Subtract, Multiply, Divide methods. Call these methods from Main() function using uni-cast delegates as well as multicast delegates.
    Create another Math class using Sum, Subtract, Multiply, Divide methods as instance methods. Call these methods from Main() function using uni-cast delegates as well as multicast delegates.

19. blank

20. Create event “EmpListChanged” in Company class. Event must be fired when employee is added or removed from the company. Event handler in client project should take care of updating salary expense of the company.