

Mechi Multiple Campus

(Tribhuvan University)

Bhadrapur, Jhapa



Lab Report of DotNet Technology (CACS-302)

Faculty of Humanities & Social Sciences

Tribhuvan University

Kritipur, Nepal

Submitted By

Name: Santosh Bhandari

Roll No: 58

Submitted To

Mechi Multiple Campus

Department of Bachelor in Computer

Bhadrapur, Jhapa, Nepal

Table Of Contents

S.N.	Title	Page. No.
1.	LABSHEET - 1	1 - 19
2.	LABSHEET - 2	20 - 37
3	LABSHEET - 3	38 - 44

LAB SHEET - 1

1. Write C# Program to calculate factorial of a number using value type.

Using System;

class Factorial {

 static void Main()

 Console.WriteLine("Enter a Number :");

 int n = Convert.ToInt32(Console.ReadLine());

 int fact = 1;

 for (int i = 1; i <= n; i++) {

 fact *= i;

 Console.WriteLine("Factorial of " + n + " is " + fact);

}

Output

Enter a Number :

5

Factorial of 5 is 120.

2. Write a C# Program to Calculate Sum of array Using reference type.

Using System;

class SumOfArray {

 static void Sum(ref int[] arr) {

 int sum = 0;

 foreach (int i in arr)

 sum += i;

 Console.WriteLine("Sum = " + sum);

 static void Main()

 int[] arr = { 5, 26, 23, 1, 26 };

 Sum(ref arr);

}

Output

Sum = 81.

3. Write a C# program to illustrate call by value and call by reference.

using System;

class CallByValueAndReference

static void ByValue(int a)

Console.WriteLine("By Value = {0}", a);
a=10;

}

static void ByReference(ref int a)

Console.WriteLine("By Reference = {0}", a);
a=10;

}

static void Main()

int a=5;

ByValue(a);

Console.WriteLine("After By Value : {0}", a),

ByReference(ref a);

Console.WriteLine("After By Reference : {0}", a),

3

Output

By Value = 5

After By Value : 5

By Reference = 5

After By Reference : 10

4. Write a C# program to illustrate different parameters like ref, out, params, optional, etc.

using System;

class DifferentParameters

static void Reference(ref int a)

Console.WriteLine("By Reference : {0}", a);

static void OutPara(int a, int b, out int sum)

sum=a+b;

static void Params(params int[] arr)

Console.WriteLine("Params : ");

foreach(int i in arr) {

```

    Console.WriteLine("\t"+i);
}

static void Optional (int a, int b=10){
    Console.WriteLine("InSum : {0}", a+b);
}

static void Main(){
    int a=10, sum=0;
    Reference(ref a);
    OutParam(10,20,out sum);
    Console.WriteLine("Sum = {0}",sum);
    Params(1,2,3,4,5);
    Optional(10);
}

```

Output

By Reference : 10

Sum : 30

Params : 1 2 3 4 5

Sum : 20

5. Write a C# program to input 10 numbers in array and find largest number.

```

using System;
class FindLargest{
    static int Largest(int[] arr){
        int max=arr[0];
        for(int i; i<arr.Length; i++){
            if(arr[i]>max){
                max=arr[i];
            }
        }
        return max;
    }

    static void Main(){
        int[] arr={10,22,13,44,50,18,142,15,10,153};
        Console.WriteLine("Largest : {0}", Largest(arr));
    }
}

```

Output

Largest : 155

6. Write a C# program to calculate sum of all elements of rectangular array.

using System;

class SumOfArray{

static void Main(){

int[,] arr = new int[3,4] {{1,2,3,4},{5,6,7,8},{9,10,11,12}};

int sum = 0;

foreach(int i in arr)

sum += i;

Console.WriteLine("Sum : {0}", sum);

}

}

Output

Sum : 78

7. Write a C# program to calculate Product of each row of Jagged array.

using System;

class ProductOfJaggedArray{

static void Main(){

int[][] arr = {

new int[] {1,2,3,4},

new int[] {5,6,7},

new int[] {8,9},

};

foreach(int[] ar in arr){

int product = 1;

foreach(int num in ar)

product *= num;

Console.WriteLine("Product : {0}", product);

}

}

3

Output

Product : 24

Product : 210

Product : 72

8. Write a C# program to Create and Use namespace.

BCA.cs

```
namespace BCA
{
    using System;
    public class A
    {
        public void Display()
        {
            Console.WriteLine("Welcome to BCA.");
        }
    }
}
```

MainProgram.cs

```
using System;
using BCA;
class MainProgram
{
    static void Main()
    {
        A a = new A();
        a.Display();
    }
}
```

Output

Welcome to BCA

9. Write a C# program to find sum and Product of number using delegate.

```
using System;
class DelegatesProgram
{
    public delegate void Calculation(int x, int y);
    static void Sum(int a, int b)
    {
        Console.WriteLine("Sum : {0}", (a+b));
    }
}
```

```

static void Product(int a, int b) {
    Console.WriteLine("Product : {0}", (a * b));
}

static void Main() {
    Calculation calc = new Calculation(Sum);
    calc += Product;
    calc(10, 20);
}

```

3

Output

Sum = 30

Product = 200

10. Write a C# program to calculate factorial of a number using delegate.

```

using System;
class FactorialDelegate {
    delegate void Factorial(int n);
    static void FactorialCalc(int num) {
        int fact = 1;
        for (int i = 1; i <= num; i++)
            fact *= i;
        Console.WriteLine("Factorial of {0} is {1}.", num, fact);
    }
}

static void Main() {
    Factorial fact = new Factorial(FactorialCalc);
    fact(5);
}

```

3

Output

Factorial of 5 is 120.

11. Write a C# program to demonstrate multicasting of delegates.

```

using System;
class MulticastingDelegates {
    delegate void Multicast();
    static void Method1() {
        Console.WriteLine("Method 1");
    }
    static void Method2() {
        Console.WriteLine("Method 2");
    }
    static void Method3() {
        Console.WriteLine("Method 3");
    }
    static void Main() {
        Multicast obj = new Multicast(Method1);
        obj += Method2;
        obj += Method3;
        obj();
    }
}

```

OUTPUT

Method 1

Method 2

Method 3

12. Write a C# program to check whether a number is odd or even using function delegate.

```

using System;
class FunctionDelegates {
    static string CheckOddEven(int n) {
        if (n % 2 == 0) {
            return "Even";
        } else {
            return "Odd";
        }
    }
    static void Main() {
        Func<int, string> obj = new Func<int, string>(CheckOddEven);
    }
}

```

```
Console.WriteLine("5 is {0}.", obj(5));
Console.WriteLine("10 is {0}.", obj(10));
```

}

)

Output

5 is Odd.

10 is Even.

13. Write a C# program to check whether a number is prime or composite using action delegate.

using System;

class PrimeOrComposite

static void Check(int n)

int counter = 0;

for(int i = 2; i < n; i++)

if(n % i == 0)

counter++;

break;

}

if(counter == 0)

Console.WriteLine("{0} is Prime Number.", n);

else

Console.WriteLine("{0} is Composite Number.", n);

}

}

static void Main()

Action<int> obj = new Action<int>(check);

obj(5);

obj(10);

obj(13);

}

}

Output

5 is Prime Number.

10 is Composite Number.

13 is Prime Number.

14. Write a C# program to demonstrate Statement Lambda.

```
using System;
class StatementLambda{
```

```
    delegate void MyDel(int n);
```

```
    static void Main(){
```

```
        MyDel obj = (n) => {
```

```
            if(n%2==0){
```

```
                Console.WriteLine($"n is Even Number.");
```

```
            } else {
```

```
                Console.WriteLine($"n is Odd Number.");
```

```
}
```

```
    obj(5);
```

```
    obj(10);
```

```
}
```

```
3
```

Output

5 is Odd Number.

10 is Even Number.

15. Write a C# program to demonstrate expression lambda.

```
using System;
```

```
class ExpressionLambda{
```

```
    static int Sum(int x, int y) => x+y;
```

```
    static void Main(){
```

```
        Console.WriteLine($"Sum of 10 and 20 is {30.}", Sum(10, 20));
```

```
}
```

```
3
```

Output

Sum of 10 and 20 is 30.

16. Write a C# program to demonstrate exception handling using try, catch and finally block.

```
using System;
class ExceptionHandling {
    static void Main() {
        int a=10, b=0;
        try {
            Console.WriteLine("Division of {0} and {1} is {2}.", a, b, b);
        } catch (Exception e) {
            Console.WriteLine(e.Message);
        } finally {
            Console.WriteLine("Finally Block is Executed.");
        }
    }
}
```

Output

Attempted to divide by zero.

Finally Block is Executed.

17. Write a C# program to handle index out of range exception.

```
using System;
class OutOfRangeException {
    static void Main() {
        int[] arr = new int[5];
        try {
            for (int i=0; i<10; i++)
                arr[i] = i;
        } catch (Exception e) {
            Console.WriteLine(e.Message);
        }
    }
}
```

Output

Index was outside the bounds of the array.

18. Write a C# Program to handle IO Exception.

```
using System;
using System.IO;
class IOExceptionEx
{
    static void Main()
    {
        try
        {
            StreamReader sr = new StreamReader("data.txt");
            Console.WriteLine(sr.ReadToEnd());
        }
        catch (IOException e)
        {
            Console.WriteLine(e.Message);
        }
    }
}
```

Output

Could not find file "/home/santosh/Desktop/practical/data.txt"

19. Write a C# program to handle NullReferenceException.

```
using System;
class NullReference
{
    static void Main()
    {
        try
        {
            string str=null;
            Console.WriteLine(str.Length);
        }
        catch (NullReferenceException e)
        {
            Console.WriteLine(e.Message);
        }
    }
}
```

Output

Object reference not set to an instance of an object.

20. Write a C# program to input two numbers. If both numbers are equal then throw your own custom exception. Otherwise calculate sum of two numbers.

```

using System;
class EqualException : Exception
    public EqualException(string msg) : base(msg) {}}

class CustomException {
    static void Main() {
        Console.WriteLine("Enter First Number : ");
        int a = int.Parse(Console.ReadLine());
        Console.WriteLine("Enter Second Number : ");
        int b = int.Parse(Console.ReadLine());
        try {
            if (a == b) {
                throw new EqualException("Equal Numbers.");
            }
            else {
                Console.WriteLine("Sum of " + a + " and " + b + " is " + (a + b));
            }
        }
        catch (EqualException e) {
            Console.WriteLine(e.Message);
        }
    }
}

```

Output

```

Enter First Number :
10
Enter Second Number :
10
Equal Numbers

```

21. Write a C# program for rethrowing of exception.

```

using System;
class Rethrowing {
    static void Main() {
        Console.WriteLine("Enter a Number : ");
    }
}

```

```

int a = int.Parse(Console.ReadLine());
try {
    if (a < 0)
        throw new Exception("Number is Negative.");
    else
        Console.WriteLine("Number is Positive.");
} catch (Exception e) {
    Console.WriteLine(e.Message);
    throw;
}

```

3

Output

Enter a Number:

-5

Number is Negative.

Unhandled Exception:

System.Exception : Number is Negative.

=====

22. Write a program in C# to create a list of numbers and display the numbers greater than 80 as output using LINQ. Also display all selected numbers in descending order.

```

using System;
using System.Collections.Generic;
using System.Linq;
class DisplayNumberLinq {
    static void Main() {
        List<int> lot = new List<int>() { 100, 30, 80, 180, 470, 48, 490,
                                             6958, 65, 66, 8, 69, 55, 105, 140, 65, 4, 165, 21, 1, 58, 4, 65 };
        var filteredData = from x in lot
                           where x > 80
                           order by x descending
                           select x;
    }
}

```

```

Console.WriteLine("Number Greater than 80 are:-");
foreach (var x in filteredData)
    Console.WriteLine(x + ",");
}

```

Output

Number Greater than 80 are:
6958, 480, 470, 180, 165, 140, 105, 100

23. Write a c# program to demonstrate aggregate function using LINQ.

```

using System;
using System.Linq;
class AggregateLINQ {
    static void Main() {
        int[] arr = { 4, 8, 10, 48, 17, 46, 79, 48, 25, 20 };
        var res = from x in arr
                  where x > (
                      from y in arr
                      select y
                  ).Average()
                  select x;
        foreach (int data in res) {
            Console.WriteLine(data);
        }
    }
}

```

Output

48
46
79
48

24. Write a C# program to store id, name, age, address and salary of 5 employees in a list. Now write LINQ for the following;
- Select name and age of employees whose salary is greater than 20000 and age is less than 35.
 - Select all records of employees whose name starts with letter 'R' and age greater than 25 in descending order based on age.
 - Select all records of employee whose salary is maximum.
 - Select id and name of employee whose salary is between 20000 and 30000.
 - Select id and name of employees whose address is BTM and salary is greater than average salary of employees.
 - Select records of employee group by address in ascending order based on age.

using System;

using System.Collections.Generic;

using System.Linq;

class Employee{

 public int id, age;

 public double salary;

 public string name, address;

 public Employee(int id, string name, int age, string address, double salary) {

 this.id = id;

 this.name = name;

 this.age = age;

 this.address = address;

 this.salary = salary;

}

3

class LINQQuery{

 static void Main() {

 List<Employee> list = new List<Employee>();

 list.Add(new Employee(1, "Rohit", 25, "KTM", 20000));

 list.Add(new Employee(2, "Raj", 30, "KVT", 30000));

 list.Add(new Employee(3, "Ravi", 35, "BTM", 40000));

 list.Add(new Employee(4, "Rahul", 40, "BR7", 50000));

 list.Add(new Employee(5, "Krishna", 45, "BTM", 30000));

var res1 = from x in list

where x.salary > 20000 && x.age < 35

select new {x.name, x.age};

Console.WriteLine("Name and Age of Employee whose salary > 20000 and age < 35");

foreach (var item in res1)

Console.WriteLine("{{0}{1}{2}}", item.name, item.age);

var res2 = from x in list

where x.name.StartsWith("R") && x.age > 25
orderby x.age descending

select n;

Console.WriteLine("Employee Records whose name starts with letter 'R' and age > 25 in descending order based on age:");

foreach (var item in res2)

Console.WriteLine("{{0}{1}{2}{3}{4}}", item.id,
item.name, item.age, item.address, item.salary);

var res3 = from x in list

where x.salary == (from y in list select y.salary).Max()

select x;

Console.WriteLine("Records of Employee whose salary is max:");

foreach (var item in res3)

Console.WriteLine("{{0}{1}{2}{3}{4}}", item.id,
item.name, item.age, item.address, item.salary);

var res4 = from x in list

where x.salary >= 20000 && x.salary <= 30000

select new {n.id, n.name};

Console.WriteLine("Id and Name of Employee whose salary is between 20000 and 30000:");

foreach (var item in res4)

Console.WriteLine("{{0}{1}}", item.id, item.name);

var res5 = from x in list

where x.address == "BTM" && x.salary > (

from y in list

Select y.salary
).Averagel
 select new {x.id, x.name};
 Console.WriteLine("Id and Name of Employees whose address
 is Btm and Salary > average:");
 foreach(var item in res5)
 (Console.WriteLine("S031t{13", item.id, item.name);
 var res6 = from x in list
 orderby x.age ascending
 group x by x.address;
 Console.WriteLine("Records of Employee Group by address in
 ascending order based on age:");
 foreach(var items in res6){
 Console.WriteLine("Address : S05", items.Key);
 foreach(var item in items)
 Console.WriteLine("S05t{13t{23t{33t{43",
 item.id, item.name, item.age, item.address,
 item.salary);
 }
 }
 }
 }
Output

Name and Age of Employees Whose salary > 20000 and age < 35:

Raj 30

Records of Employee Whose name starts with letter 'R' and age > 25 in
 descending order based on age :

4	Rahul	40	BRT	50000
3	Ravi	35	BTM	40000
2	Raj	30	KVT	30000

Records of Employee Whose Salary is Maximum

2	Raj	30	KVT	50000
1	Rahul	40	BRT	50000

ID and Name of Employee Whose salary is between 20000 and 30000 :

- 1 Rohit
- 5 Krishna

ID and Name of Employees Whose address is BTM and salary > average :

- 3 Ravi

Records of Employee group by address in ascending order based on age :

Address : KTM

- 1 Rohit 25 KTM 20000

Address : KVT

- 2 Ravi 30 KVT 50000

Address : DTM

- | | | | | |
|---|---------|----|-----|-------|
| 3 | Ravi | 35 | DTM | 40000 |
| 5 | Krishna | 45 | DTM | 50000 |

Address : BRT

- 4 Rahul 40 BRT 50000

25. Write a c# program to demonstrate Join Using LINQ.

using System;

using System.Linq;

using System.Collections.Generic;

class Student {

public int sid, cid;

public String name, address;

public Student(int sid, string name, string address, int cid) {

this.sid = sid;

this.name = name;

this.address = address;

this.cid = cid;

}

3

class Course {

public int cid;

public String cname;

```

public Course(int cid, string cname) {
    this.cid = cid;
    this.cname = cname;
}

class Joining {
    static void Main() {
        List<Student> std = new List<Student>();
        std.Add(new Student(1, "Rahul", "KTM", 1));
        std.Add(new Student(2, "Raj", "KVT", 2));
        std.Add(new Student(3, "Ravi", "BTM", 1));
        std.Add(new Student(4, "Rashi", "BRT", 3));
        std.Add(new Student(5, "Krishna", "BTM", 2));
        List<Course> cs = new List<Course>();
        cs.Add(new Course(1, "C#"));
        cs.Add(new Course(2, "Java"));
        cs.Add(new Course(3, "Python"));
        var res = from x in std
                  join y in cs
                  on x.cid equals y.cid
                  where y.cname == "C#"
                  select new {x.name, x.address, y.cname};

        Console.WriteLine("Name\tAddress\tCourse");
        foreach(var data in res)
            Console.WriteLine("{0}\t{1}\t{2}", data.sname, data.address,
                             data.cname);
    }
}

```

Output

Name	Address	Course
Rahul	KTM	C#
Ravi	BTM	C#

LABSHEET - 2

- Q. Create a simple class name Person that contains basic information like name, age, gender etc. Your class should also contain functions/methods for storing and displaying data.

```
using System;
class Person {
    public string name, gender,
    public int age;
    public void setData(string name, string gender, int age) {
        this.name = name;
        this.age = age;
        this.gender = gender;
    }
    public void displayData() {
        Console.WriteLine($"Name : {this.name}\nGender : {this.gender},
                        \nAge : {this.age}");
    }
}
```

class Program {

```
static void Main() {
    Person p = new Person();
    p.setData("Santosh Bhandari", "male", 23);
    p.displayData();
}
```

}

Output

Name : Santosh Bhandari

Gender : Male

Age : 23

- Q. 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.

```
using System;
class Box {
    private int length, breadth, height;
    public void ReadData(int l, int b, int h) {
```

```

length = l;
breadth = b;
height = h;
}

public void Volume() {
    Console.WriteLine($"Volume : {length * breadth * height}");
}

class Program {
    static void Main() {
        Box b = new Box();
        b.ReadData(10, 20, 30);
        b.Volume();
    }
}

Output
Volume : 6000

```

3. Write a program to define a class Circle with its data members π and r and member functions getdata() for initializing data members and calculate() for finding area of a circle. Return result from calculate() and display result in main() function. Use π as constant.

```

using System;
class Circle {
    private double pi = 3.14, r;
    public void getdata(double r) {
        this.r = r;
    }
    public double calculate() {
        return pi * r * r;
    }
}

class Program {
    static void Main() {
        Circle c = new Circle();
        c.getdata(10);
        Console.WriteLine("Area : {0}", c.calculate());
    }
}

```

Output

Area : 314

4.

Define a class Rectangle with data members : length and breadth. The program should read length and breadth of a rectangle in main() function and supply them in parameterized constructor to initialize its data member.

using System;

class Rectangle{

private int length, breadth,

public Rectangle(int l, int b){

length = l;

breadth = b;

}

public Area1{

Console.WriteLine(\$"Area : {this.length * this.breadth}");

}

}

class Program{

static void Main(){

Rectangle r = new Rectangle(50, 100),

r.Area1;

}

Output

Area : 5000

5.

Write a program to demonstrate static constructor.

using System;

class Person{

public static string Name;

static Person(){

Name = "John";

}

```

class Program{
    static void Main(){
        Console.WriteLine("Name of Person : " + person.Name),
    }
}

```

Output

Name of Person : John

6. Write a program to illustrate the usage of this keyword.

```

using System;
class Person{
    public string Name;
    public Person(string Name){
        this.Name = Name;
    }
}

class Program{
    static void Main(){
        Person p = new Person("John Smith");
        Console.WriteLine("Name of Person : " + p.Name);
    }
}

```

Output

Name of Person : John Smith

7. Write a program to calculate Sum and Product of two numbers using automatic property.

```

using System;
class Numbers{
    public int Number1 {get; set;}
    public int Number2 {get; set;}
    public int Sum {

```

```

    get {
        return Number1 + Number2;
    }
}

public int Product {
    get {
        return Number1 * Number2;
    }
}

class Program {
    static void Main() {
        Number n = new Number();
        n.Number1 = 10;
        n.Number2 = 20;
        Console.WriteLine("Sum : " + n.Sum);
        Console.WriteLine("Product : " + n.Product);
    }
}

```

Output

Sum : 30
 Product : 200

8. Write a C# program to show the usage of abstract class and methods.

```

using System;
abstract class Shape {
    public abstract void Draw();
}

class Rectangle : Shape {
    public void Draw() {
        Console.WriteLine("Drawing Rectangle");
    }
}

class Program {
    static void Main() {
    }
}

```

Rectangle r = new Rectangle();
r.Draw()

3

3

OutputDrawing Rectangle

9. Write a C# program to show the usage of Sealed class.

Using System;

Sealed class A{

```
    public void Show(){
        Console.WriteLine("class A");
```

3

3
class B : A{

3

class Program{

```
    static void Main(){
        A a = new A();
        a.Show();
```

3

3

Output

Throw an error

10. Write a C# program to demonstrate method overloading.

Using System;

class A{

public void Show(int a){

```
    Console.WriteLine("One Argument : "+a);
```

3

```

public void Show(int a, int b){
    Console.WriteLine("Two Argument : {0}, {1}", a, b);
}
3
class Program{
    static void Main(){
        A a = new A();
        a.Show(10);
        a.Show(10, 20);
    }
}
3

```

Output

One Argument : 10
 Two Argument : 10, 20

11. Write a C# program to demonstrate method overriding using virtual method.

```

using System;
class A{
    public virtual void Show(){
        Console.WriteLine("Class A");
    }
}
3
class B:A{
    public override void Show(){
        Console.WriteLine("Class B");
    }
}
3
class Program{
    static void Main(){
        B a = new B();
        a.Show();
    }
}
3

```

Output

class B

12. Write programs to demonstrate all types of Inheritance.

Single Inheritance

using System;

class A{

public void Display(){}

Console.WriteLine("class A");

}

class B : A{

public void Show(){}

Console.WriteLine("class B");

}

}

class Program{

static void Main(){}

B b = new B();

b.Display();

b.Show();

}

}

Output

class A

class B

Multilevel Inheritance

using System;

class A{

public void Display(){}

Console.WriteLine("class A");

}

class B : A{

```

public void Show() {
    Console.WriteLine("Class B");
}
3
class C : B {
    public void Print() {
        Console.WriteLine("Class C");
    }
}
3
class Program {
    static void Main() {
        C c = new C();
        c.Display();
        c.Show();
        c.Print();
    }
}
3

```

Output

```

class A
class B
class C

```

Hierarchical Inheritance

```

using System;
class A {
    public void Display() {
        Console.WriteLine("Class A");
    }
}
3
class B : A {
    public void Show() {
        Console.WriteLine("Class B");
    }
}
3
class C : B {
    public void Print() {
        Console.WriteLine("Class C");
    }
}
3
class Program {

```

```
static void Main()
{
    B b = new B();
    C c = new C();
    b.Display();
    b.Show();
    c.Display();
    c.Print();
}
```

3

3

Output

```
Class A
Class B
Class A
Class C
```

Multiple Inheritance

```
using System;
interface A
{
    void Display();
}

class B
{
    public void Show()
    {
        Console.WriteLine("Class B");
    }
}

class C : B, A
{
    public void Print()
    {
        Console.WriteLine("Class C");
    }

    public void Display()
    {
        Console.WriteLine("Class A");
    }
}

class Program
{
    static void Main()
    {
        C c = new C();
        c.Display();
        c.Show();
    }
}
```

(cont'd);

3

Output

Class A

Class B

Class C

13. Write a program to calculate area and perimeter of rectangle using interface.

Using System,

interface IRectangle

```
int Area(int length,int breadth)
```

```
int Perimeter(int length, int breadth);
```

3

```
class Rectangle : IRectangle {
```

```
public int Area(int length, int breadth){
```

return length * breadth;

3 public int Perimeter(int length, int breadth){

return 2 * (length + breadth);

3

3

class Programs

```
static void Main()
```

```
Rectangle r = new Rectangle();
```

Console.WriteLine("Area : "+r.Area(10,20));

```
Console.WriteLine("Perimeter : "+r.Perimeter(10,20));
```

3

Output

Area : 200

Perimeter : 60

14. Write a C# program to demonstrate the use of "base" keyword in inheritance.

Using System;

class A{

```
public void Show()
```

Console.WriteLine("Class A");

3

```
class B: A{
    public void Show(){
        base.Show();
        Console.WriteLine("Class B");
    }
}
```

3

```
class Program{
    static void Main(){
        B b = new B();
        b.Show();
    }
}
```

Output

class A
class B

15. Write a program to demonstrate indexes.

using System;

class A{

```
private string[] names = new String[5];
public String this[int index]{
    get{
        return names[index];
    }
    set{
        names[index] = value;
    }
}
```

3

```
class Program{
    static void Main(){
        A obj = new A();
        obj[0] = "Sandesh";
        obj[1] = "Ramesh";
        obj[2] = "Mahesh";
        obj[3] = "Ganesh";
    }
}
```

```

obj[4] = "Rajesh";
for(int i=0; i<5; i++) {
    Console.WriteLine(obj[i]);
}

```

3

3

Output

Santosh
Ramesh
Mahesh
Ganesh
Rajesh

16. Write a Program to demonstrate Struct.

```

using System;
struct Student {
    public int rollno, age;
    public string name;
    public void Display() {
        Console.WriteLine("Roll No : " + rollno);
        Console.WriteLine("Name : " + name);
        Console.WriteLine("Age : " + age);
    }
}

```

3

3
class Program

```

{
    static void Main() {
        Student s;
        s.rollno = 100;
        s.name = "John";
        s.age = 20;
        s.Display();
    }
}

```

3

3

Output

Roll No : 100

Name : John

Age : 20

17. Write a program to demonstrate enum.

using System;

class Programs

```
enum Days {Sun, Mon, Tue, Wed, Thu, Fri, Sat};
```

```
static void Main()
```

```
int n = (int) Days.Sun;
```

```
int y = (int) Days.Fri;
```

```
Console.WriteLine("Sun = " + n);
```

```
Console.WriteLine("Fri = " + y);
```

3

3

Output

Sun = 0

Fri = 5

18. 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.

using System;

class ~~Programs~~ Polygon

```
public int length, breadth;
```

```
public Polygon(int l, int b){
```

```
length = l;
```

```
breadth = b;
```

3

3

```

class Rectangle : Polygon {
    public Rectangle (int l, int b) : base (l, b) {
        Console.WriteLine ("Area of Rectangle : " + (length * breadth));
    }
}

class Triangle : Polygon {
    public Triangle (int l, int b) : base (l, b) {
        Console.WriteLine ("Area of Triangle : " + (length * breadth * 0.5));
    }
}

class Program {
    static void Main () {
        Rectangle r = new Rectangle (5, 10);
        Triangle t = new Triangle (5, 10);
    }
}

Output
Area of Rectangle : 50
Area of Triangle : 25
  
```

19. 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 rectangle. Let class 'Square' inherit the 'Rectangle' and 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.

using System;

```

class Rectangle {
    public int length, breadth;
    public Rectangle (int l, int b) {
        length = l;
        breadth = b;
    }
    public void Area () {
        Console.WriteLine ("Area of Rectangle : " + (length * breadth));
    }
}
  
```

```

public void Perimeter(){
    Console.WriteLine("Perimeter of Rectangle : " + (2 * (length+breadth)));
}

class Square : Rectangle{
    public Square(int s) : base(s,s){}
}

class Program{
    static void Main(){
        Rectangle r = new Rectangle(5,10);
        r.Area();
        r.Perimeter();

        Square s = new Square(5);
        s.Area();
        s.Perimeter();
    }
}

```

Output

Area : 50

Perimeter : 30

Area : 25

Perimeter : 20

20. 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 member. 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.

Using System;

class Member{

 public string Name, Phone Number, Address;

 public int Age, Salary;

 public void PrintSalary(){

 Console.WriteLine("Salary : "+Salary);

}

}

class Employee : Member{

 public string Specialization;

}

class Manager : Member{

 public string Department;

}

class Program{

 static void Main(){

 Employee e = new Employee();

 e.Name = "John";

 e.Age = 30;

 e.PhoneNumber = "1234567890";

 e.Address = "Bishamade";

 e.Salary = 50000;

 e.Specialization = "Software Developer";

 Console.WriteLine(\$"Employee Details : \nName : {e.name}\nAge : {e.Age}\nPhone Number : {e.PhoneNumber}\nAddress : {e.Address}");

 e.PrintSalary();

 Console.WriteLine("Specialization : "+e.Specialization);

 Manager m = new Manager();

 m.Name = "Jane";

 m.Age = 35;

 m.PhoneNumber = "0987654321";

 m.Address = "Kathmandu";

 m.Salary = 100000;

 m.Department = "IT";

 Console.WriteLine(\$"Manager Details : \nName : {m.Name}\nAge : {m.Age}\nPhone Number : {m.PhoneNumber}\nAddress : {m.Address}");

```
m.PrintSalary());
Console.WriteLine("Department : " + m.Department);
```

3

3

Output

Employee Details:

Name : John

Age : 30

Phone Number : 1234567890

Address : Bramapuri

Salary : 50000

Specialization : Software Developer

Manager Details:

Name : Jane

Age : 35

Phone Number : 0987654321

Address : Kathmandu

Salary : 100000

Department : IT

Labsheet - 3

1. Write a C# program to create table named employee with columns eid as int, name as varchar, address as varchar and salary as double using ADO.NET.

```
using System;
using MySql.Data.MySqlClient;
class Program {
    static void Main() {
        string url = "SERVER=localhost;DATABASE=BCA;UID=root;PASSWORD=" + " ";
        MySqlConnection con = new MySqlConnection(url);
        con.Open();
        MySqlCommand cmd = new MySqlCommand("CREATE TABLE employee(eid int, name varchar(30), address varchar(50), salary double)", con);
        cmd.ExecuteNonQuery();
        Console.WriteLine("Table Created Successfully.");
        con.Close();
    }
}
```

Output

Table Created Successfully.

2. Write a C# program to insert records in above employee table.

```
using System;
using MySql.Data.MySqlClient;
class Program {
    static void Main() {
        string url = "SERVER=localhost;DATABASE=BCA;UID=root;PASSWORD=" + " ";
        MySqlConnection con = new MySqlCommand("INSERT INTO employee(eid, name, address, salary) VALUES(100, 'Santosh', 'Baramode', 35000), (101, 'krishna', 'Bhadrapur', 25000), (102, 'Manoj', 'Damak', 45000)", con);
        cmd.ExecuteNonQuery();
    }
}
```

```
Console.WriteLine("Data Inserted Successfully.");
con.Close();
```

3

3

Output

Data Inserted Successfully.

3. Write a C# program to update name as Ram and salary as 55000 of employee whose eid is 102.

```
using System;
using MySql.Data.MySqlClient;
class Program
{
    static void Main()
    {
        string vr1 = "SERVER=localhost;DATABASE=BCA;UID=root;PASSWORD=";
        MySqlConnection con = new MySqlConnection(vr1),
        con.Open();
        MySqlCommand cmd = new MySqlCommand("UPDATE employee SET
        name='RAM', salary=5500 WHERE eid=102;", con);
        cmd.ExecuteNonQuery();
        Console.WriteLine("Data Updated Successfully.");
        con.Close();
    }
}
```

3

Output

Data Updated Successfully.

4. Write a C# program to delete record of employee whose eid is 104.

```
using System;
using MySql.Data.MySqlClient;
class Program
{
    static void Main()
    {
    }
```

```
string vr1 = "SERVER=localhost;DATABASE=BCA;UID=root, PASSWORD=";
```

```

MySql/Connection con = new MySql/Connection(url),
con.Open(),
MySql/Command cmd = new MySql/Command ("DELETE FROM
employee WHERE eid=104; ", con),
cmd.ExecuteNonQuery(),
Console.WriteLine ("Data Deleted Successfully."),
con.Close(),
}

```

3

Output

Data Deleted Successfully.

5. Write a C# program to Select records of employee whose address is Btm
and Salary is between 30000 to 40000 using Connected architecture (DataReader).

```

using System;
using MySql.Data.MySqlClient;
class Program
{
    string url = "SERVER=localhost;DATABASE=BCA;UID=root;PASSWORD=";
    MySql/Connection con = new MySql/Connection(url),
    con.Open(),
    MySql/Command cmd = new MySql/Command ("SELECT * FROM employee
    WHERE address='Baramode' AND salary BETWEEN 30000 AND
    40000; ", con),
    MySql/DataReader reader = cmd.ExecuteReader(),
    while (reader.read())
    {
        Console.WriteLine (reader[0] + " " + reader[1] + " " + reader[2]
        + " " + reader[3]);
    }
    con.Close();
}

```

3

Output

101 Santosh Baramode 35000

6. Write a C# Program to Select name and address of employee whose salary is maximum using disconnected architecture. (Data Adapter).

```

using System;
using MySql.Data.MySqlClient;
class Program{
    static void Main(){
        string url = "SERVER=localhost;DATABASE=BDA;UID=root;PASSWORD=;";
        MySqlConnection con = new MySqlConnection(url);
        con.Open();
        MySqlCommand cmd = new MySqlCommand("SELECT * FROM employee WHERE Salary = (SELECT MAX(Salary) from employee);", con);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
        System.Data.Dataset ds = new System.Data.Dataset();
        adapter.Fill(ds);
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++){
            Console.WriteLine(ds.Tables[0].Rows[i][1] + " " +
            ds.Tables[0].Rows[i][2]);
        }
        con.Close();
    }
}

```

Output

Ram Damak

7. Create following form using Asp. Net.

Personal Details	
Salutation	<input type="text" value="Mr"/>
First Name :	<input type="text"/>
Last Name :	<input type="text"/>
Gender :	<input type="radio"/> Male <input type="radio"/> Female
Email :	<input type="text"/>
Date of Birth :	<input type="text" value="dd/mm/yyyy"/>

Address:

```

<%@ Page language = "C#" %>
<!DOCTYPE html>
<html lang = "eng" xmlns = "http://www.w3.org/1999/xhtml">
<head runat = "Server">
<meta charset = "utf-8"/>
<title> Web Form </title>
</head>
<body>
<form id = "form1" runat = "Server">
<fieldset>
<legend> Personal Details </legend>
<label for = "salutation"> Salutation </label>
<select name = "salutation" id = "salutation">
<option value = "none"> None </option>
<option value = "mr"> Mr. </option>
<option value = "mrs"> Mrs. </option>
</select> <br/>
<label for = "firstname"> First Name : </label>
<input type = "text" id = "firstname" name = "firstname" /> <br/>
<label for = "lastname"> Last Name : </label>
<input type = "text" id = "lastname" name = "lastname" /> <br/>
<label for = "gender"> Gender: </label>
<input type = "radio" value = "male" name = "gender" /> Male
<input type = "radio" value = "female" name = "gender" /> Female <br/>
<label for = "email"> Email </label>
<input type = "email" id = "email" name = "email" /> <br/>
<label for = "dob"> Date of Birth : </label>
<input type = "date" id = "dob" name = "dob" /> <br/>
<label for = "address"> Address </label>
<textarea id = "address" name = "address" />

```

```

<br/>
<button type="submit">Submit </button>
</fieldset>
</form>
</body>
</html>

```

8. Create ASP.NET Web application to calculate sum of two numbers as follows:

First Number

Second Number

Result : 30

Get Result

```

<!DOCTYPE html>
<html>
<head runat="server">
    <title> Sum </title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:Label runat="server" Text="First Number"></asp:Label>
        <asp:TextBox ID="FirstNum" runat="server"></asp:TextBox><br/>
        <asp:Label runat="server" Text="Second Number"></asp:Label>
        <asp:TextBox ID="SecondNum" runat="server"></asp:TextBox><br/>
        <asp:Label ID="result" runat="server" Text="Result : ">
        </asp:Label> <br/> <br/>
        <asp:Button ID="Submit" runat="server" Text="Get Result"
            onClick="CalculateSum"/>
    </form>
</body>
</html>

```

```
</form>  
</body>  
</html>
```

.cs

```
protected void CalculateSum(object Sender, EventArgs e){  
    int n1 = Convert.ToInt32(firstNum.Text);  
    int n2 = Convert.ToInt32(secondNum.Text);  
    result.Text = "Sum : " + (n1 + n2);  
}
```

3