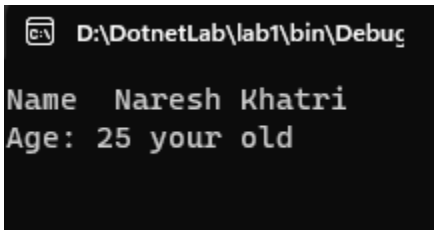
Practical 2 (Class concept, Inheritance, abstract class and interface)

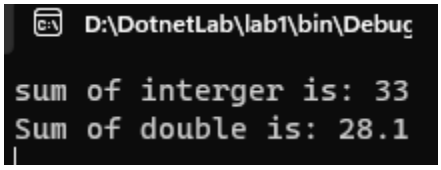
1) Write a program to demonstrate class, constructor, properties and method

```
using System;
namespace lab1
{
class Question1
{
//properties
public String FirstName{ get; set; }
public String LastName { get; set; }
public int Age { get; set; }
//constructor
public Question1( string firstname , string lastname, int age)
{
this.FirstName = firstname;
this.LastName = lastname;
this.Age = age;
}
public void Dispaly()
{
Console.WriteLine($"Name  {this.FirstName} {this.LastName}");
Console.WriteLine($"Age: {this.Age} your old");
}
}class Program
{
Public static void Main(string[] args){
Question1 obj = new Question1("Naresh", "Khatri", 25);
obj.Dispaly()
}
```



2) Write a program to demonstrate method overloading?

```
using System;
namespace lab1
{
internal class Question2
{
public int Add(int x, int y)
{
return x + y;
}
public double Add(double x, double y, double z)
{
return x + y + z;
}
}
class Program
{
static void Main(string[] args){
Question2 obj2 = new Question2();
int result1 = obj2.Add(12, 21);
double result2 = obj2.Add(12.2, 13.4, 2.5);
Console.WriteLine("sum of interger is: " +result1);
Console.WriteLine("Sum of double is: " +result2);
}
}
```

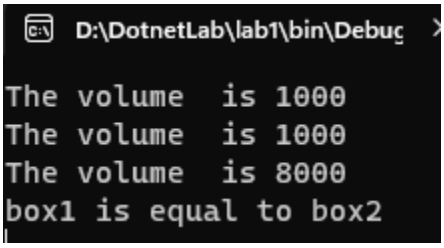


3) Write a program to demonstrate arithmetic operator overloading and relational operator overloading?

```
using System;
namespace lab1
{
internal class Question3
{
private int length;
private int breadth;
private int height;
public void SetBoxes(int lenght, int breadth, int height)
{
this.length = lenght;
this.breadth = breadth;
this.height = height;
}
// method that calcualte volume
public void CalcVol()
{
Console.WriteLine("The volume  is " + (length * breadth * height));
}
//overloading
public static Question3 operator +(Question3 b1, Question3 b2)
{
Question3 b3 = new Question3();
b3.length = b1.length + b2.length;
b3.breadth = b1.breadth + b2.breadth;
b3.height = b1.height + b2.height;
return b3; // return type is class, so object should be return
}
// overloading  "==" operator (comparison operator)
public static bool operator ==(Question3 b1, Question3 b2)
{
bool temp = false;
```

```
if (b1.length == b2.length && b1.breadth == b2.breadth && b1.height == b2.height)
{
temp = true;
return true;
}
else
{
return temp;
}
}
public static bool operator !=(Question3 b1, Question3 b2)
{
bool temp = false;
if (b1.length != b2.length && b1.breadth != b2.breadth && b1.height != b2.height)
{
temp = true;
return true;
}
else
{
return temp;
}
}
}
class Program
{
bb
static void Main(string[] args)  {
Question3 box1 = new Question3();
Question3 box2 = new Question3();
box1.SetBoxes(10, 10, 10);
box2.SetBoxes(10, 10, 10);
box1.CalcVol();
box2.CalcVol();
```

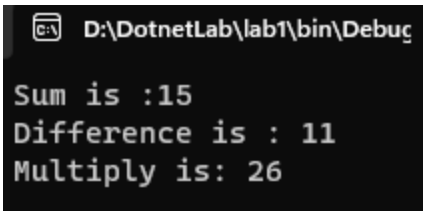
```
// calling operator overloading.....
Question3 box3 = box1 + box2;
box3.CalcVol();
if (box1 == box2)
{
Console.WriteLine("box1 is equal to box2");
}
else  {
Console.WriteLine("box1 is not eqaul to box2");
}
}
}
```



4)Create a class Calculate which contains data member num1 and num2 both in integer and methods setCalc() to set the data, calcSum() that calculate the sum of num1 and num2 and display the result, calcMulti() that calculate the multiplication of num1 and num2 and returns the result, calcDifference that calculate the difference between num1 and num2 and display the result. Now, create some instance of Calculate and invoke all the methods.

```
using System;
namespace lab1
{
class Question_4
{
private int num1;
private int num2;
public void setCalc(int num1, int num2)  {
this.num1 = num1;
this.num2 = num2;
}
public void calcSum() {
int sum =this.num1 +this.num2;
Console.WriteLine("Sum is :"+sum);
}
public int calcMulti()  {
return num1 * num2;
}
public void calcDifference(){        {
int diff = num1 - num2;
Console.WriteLine("Difference is : " +diff);
}
}
class Program{
static void Main(string[] args){
Question_4 obj4 = new Question_4();
obj4.setCalc(13, 2);
obj4.calcSum();
int mul = obj4.calcMulti();
obj4.calcDifference();
```

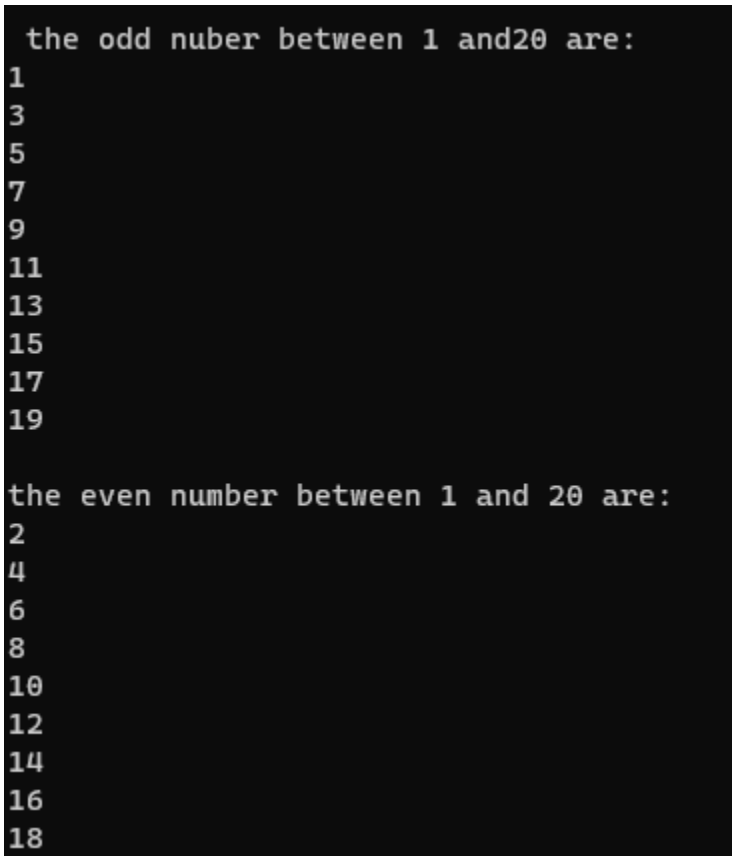
```
Console.WriteLine("Multiply is :" +mul);
}
}
}
```



5) Create a class Number having instance variable x and y both in integer, default constructor that set the value of x and y to 0, parameterized constructor that sets the value of x and y, method findOdd() that calculates the even no. occurring between x and y and display the result, findEven() that calculates the odd no. occurring between x and y and display the results. Now, create some instance of Number and invoke all the methods.

```
using System;
namespace lab1{
class Number    {
private int x;
private int y;
public Number(){
x = 0;
y = 0;
}
public Number( int x , int y){
this.x = x;
this.y = y;
}
public void Findodd(){
Console.WriteLine($" the odd nuber between {x} and{y} are:");
for(int i = x; i<y; i++){
if( i%2 !=0){
Console.WriteLine($"{i}");
}
}
Console.WriteLine();
}
public void FindEven() {
Console.WriteLine($"the even number between {x} and {y} are:");
for( int i =x; i<y; i++) {
if( i%2 == 0)  {
Console.WriteLine($"{i}");
}
}
```

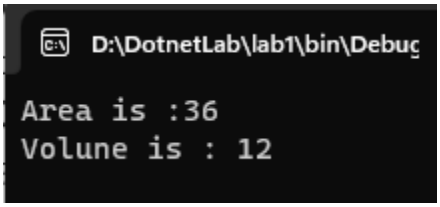
```
Console.WriteLine();
}
}
class Program{
static void Main(string[] args){
Number obj5 = new Number();
Number obj6 = new Number(1, 20);
obj6.Findodd();
obj6.FindEven();
}
}
}
```



6) Create a class Shape that contains instance variable length, breadth and height. Create a default constructor that sets the value of instance variable to zero, constructor with two parameter that will sets the value of length and breadth only and constructor with three parameter that will sets the value of length, breadth and height. After this create calcAreaRectangle() that calculates the area of rectangle, calcVolumeBox() that calculates volume of box and display the result. Now create first object of Shape wihich will have name rectangle and calls constructor with two parameter and calAreaRectangle() method, create second object of Shape that will have name Box which will call constructor with three parameter and calcVolumeBox() method.

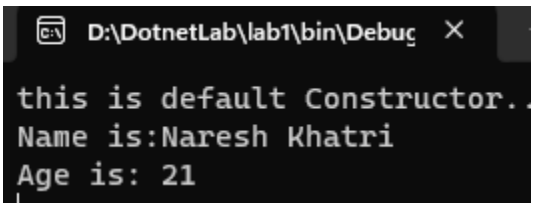
```
using System;

namespace lab1
{ class Shape
{
private int length;
private int breadth;
private int height;

public Shape()
{
length = 0;
breadth = 0;
height = 0;
}

public Shape( int length, int breadth)
{
this.length = length;
this.breadth = breadth;
}
public  Shape(int lenght, int breadth, int height)
{
this.length = lenght;
this.breadth = breadth;
this.height = height;
}
public void  calcAreaReactangel()
{
int area = length * breadth;
Console.WriteLine("Area is :" +area);
}
public void calcVolumeBox()
{
int volume = length * breadth * height;
Console.WriteLine("Volune is : " +volume );
}

}
class Program
{
static void Main(string[] args)
{

Shape shape = new Shape();
Shape rectangle = new Shape(12, 3);
Shape Box = new Shape(2, 1, 6);
rectangle.calcAreaReactangel();
```

```
Box.calcVolumeBox();
}

}

}
```



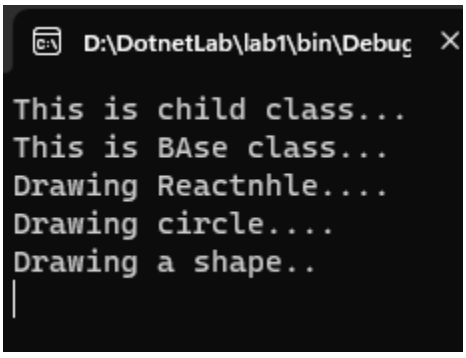
7)Write a program to demonstrate different types of constructor?

```
using System;

namespace lab1
{
class Question7
{
private string name;
private int age;

public Question7()
{
Console.WriteLine("this is default Constructor..");
}
public Question7( string name, int age)
{
this.name = name;
this.age = age;
}

public void display()
{
Console.WriteLine($"Name is:{name}");
Console.WriteLine($"Age is: {age} ");
}
}

class Program
{
static void Main(string[] args)
{

Question7 qn1 = new Question7();
Question7 obj7 = new Question7("Naresh Khatri", 21);
obj7.display();
}
}
}
```



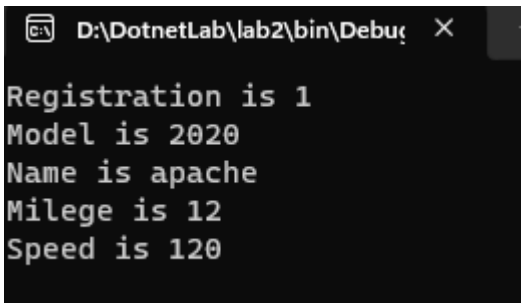
8) Write a program to demonstrate single level, multilevel inheritance?

```
using System;
namespace lab1
{
// single level
class Question8
{
public void show()
{
Console.WriteLine("This is BAse class...");
}
}
class Demo : Question8
{
public void Display()
{
Console.WriteLine("This is child class...");
}
}

// mulilevel
class Shape
{
public void drawshape()
{
Console.WriteLine("Drawing a shape..");
}

}
class circle : Shape
{
public void drawCircle()
{
Console.WriteLine("Drawing circle....");
}
}
class Rectange : circle
{
public void DrawRect()
{
Console.WriteLine("Drawing Reactnhle....");
}
}
class Program{
static void Main(string[] args) {
Demo demo = new Demo();
demo.Display();
demo.show();
Rectange rect = new Rectange();
rect.DrawRect();
rect.drawCircle();
rect.drawshape();

}

}

}
```



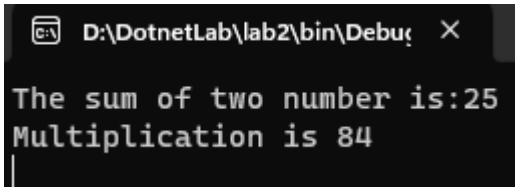
9) Write a program to demonstrate use of base keyword.

```
using System;
namespace lab2
{
class Question9
{
private int reg;
private String model;
private String name;
public  Question9(int reg, String model, String name)
{
this.reg = reg;
this.model = model;
this.name = name;

}
public void Display()
{
Console.WriteLine("Registration is " + this.reg);
Console.WriteLine("Model is " + this.model);
Console.WriteLine("Name is " + this.name);
}
}
class Bike : Question9
{
private string milege;
private string speed;
public Bike(string milege, string speed, int reg, string model, string name)   :
base(reg, model, name)

{
this.milege = milege;
this.speed = speed;
}

public void DisplayBike()
{
Console.WriteLine("Milege is " + this.milege);
Console.WriteLine("Speed is " + this.speed);

}
}
internal class Program
{
static void Main(string[] args)
{
Bike obj = new Bike("12", "120", 1, "2020", "apache");
obj.Display();
obj.DisplayBike();
}
}
}
```



10)Write a program to demonstrate method overriding (dynamic polymorphism)?

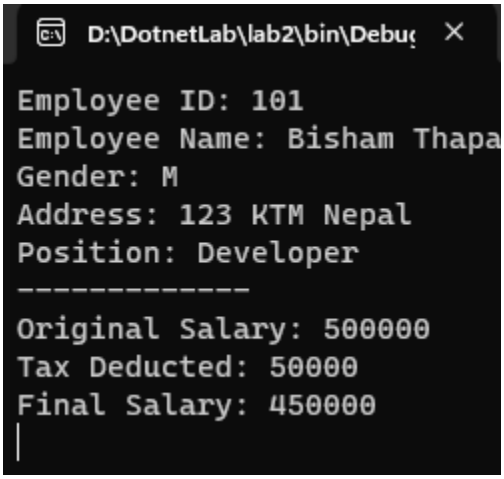
```
using System;
namespace lab2
{
class Question10
{
public virtual void calc1(int x, int y)
{
Console.WriteLine("The sum of two number is:" + (x + y));
}
public virtual void calc2( int x, int y , int z)
{
Console.WriteLine("Multiplication is " +(x*y*z));
}
}
class Solution : Question10
{
public override void calc1(int x, int y)
{
base.calc1(x, y);
Console.WriteLine("The difference is :" + (x - y));
}
public override void calc2(int x, int y, int z)
{
base.calc2(x, y, z);
Console.WriteLine("The result is :" + (x + y * z));
}
}

class Program
{
static void Main(string[] args)
{

Solution obj1 = new Solution();
obj1.calc1(12, 13);
obj1.calc2(3, 4, 7);
}
}
}
```



11) Create a class EmployeeDetails having data member empId, empName, empGender, empAddress, and empPosition, constructor to set the details and display method to display the details. Create a subclass SalaryInfo that will inherit EmployeeDetails having own data member salary which will record salary per year, constructor to set the value of salary and method calcTax() that deduct the tax on salary and display the final salary. Tax rate is given as (if salary <= 400000 tax is 1%, salary between 400001 to 800000 tax is 10% and salary > 800000 tax 20%). Now create the object of Salary info and demonstrate the scenario.

```
using System;
namespace lab2
{
class EmployeeDetails
{
protected int empId;
protected string empName;
protected char empGender;
protected string empAddress;
protected string empPosition;

public EmployeeDetails(int empId, string empName, char empGender, string empAddress,
string empPosition)
{
this.empId = empId;
this.empName = empName;
this.empGender = empGender;
this.empAddress = empAddress;
this.empPosition = empPosition;
}

public void DisplayDetails()
{
Console.WriteLine("Employee ID: " + empId);
Console.WriteLine("Employee Name: " + empName);
Console.WriteLine("Gender: " + empGender);
Console.WriteLine("Address: " + empAddress);
Console.WriteLine("Position: " + empPosition);
}
}
class SalaryInfo : EmployeeDetails
{
private double salary;

public SalaryInfo(int empId, string empName, char empGender, string empAddress, string
empPosition, double salary)
: base(empId, empName, empGender, empAddress, empPosition)
{
this.salary = salary;
}
public void CalcTax()
{
double taxRate;
if (salary <= 400000)
{
taxRate = 0.01;
}
else if (salary <= 800000)
{
taxRate = 0.1;
```

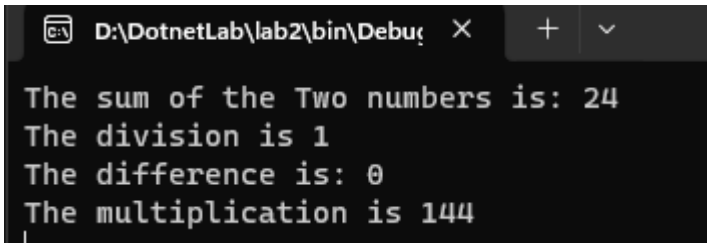
```
}
else
{
taxRate = 0.2;
}
double taxAmount = salary * taxRate;
double finalSalary = salary - taxAmount;
Console.WriteLine("Original Salary: " + salary);
Console.WriteLine("Tax Deducted: " + taxAmount);
Console.WriteLine("Final Salary: " + finalSalary);
}
}
class Program
{
  static void Main(string[] args)
 {
  SalaryInfo employeeSalary = new SalaryInfo(101, "Bisham Thapa", 'M', "123 KTM
Nepal", "Developer", 500000);
  employeeSalary.DisplayDetails();
  Console.WriteLine("-------------");
  employeeSalary.CalcTax();
}
}
```



12)Create an abstract class Calculator with instance variable int x, int y. Create constructor to initialize instance variable and create normal method findSum() that prints the sum of x and y and two abstract method findDIff() that prints difference and findMulti() that returns the multiplication. After this create a class Solution that inherits abstract class Calculator. Solution class contain two instance variable a and b both in int and one method calcDiv() that prints division. Use constructor to initialize the instance variable. Now create AbstractDemo class and show implementation of abstract class.

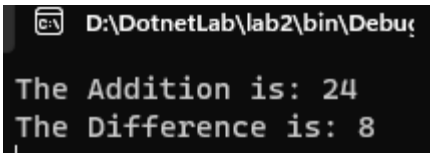
```
using System;
namespace lab2
{
abstract class Calculator
{
private int x;
private int y;
public Calculator(int x, int y)
{
this.x = x;
this.y = y;
}
public void FindSum()
{
int sum = x + y;
Console.WriteLine("The sum of the Two numbers is: " + sum);
}
public abstract void FindDiff(int x, int y);
public abstract int FindMulti(int x, int y);
}
class Solution : Calculator
{
private int a;
private int b;

public Solution(int x, int y, int a, int b) : base(x, y)
{
this.a = a;
this.b = b;
}
public override void FindDiff(int x, int y)
{
int diff = x - y;
Console.WriteLine("The difference is: " + diff);
}
public override int FindMulti(int x, int y)
{
int multi = x * y;
Console.WriteLine("The multiplication is " + multi);
return multi;
}
public void CalcDiv()
{
if (b != 0)
{
int div = this.a /this.b;
Console.WriteLine("The division is " + div);
}
else
{
```

```
Console.WriteLine("Cannot divide by zero");
}
}
}
class Program
{
static void Main(string[] args)
{
Solution s1 = new Solution(12,12,12,12);
s1.FindSum();
s1.CalcDiv();
s1.FindDiff(12, 12);
s1.FindMulti(12, 12);
}
}
}
```



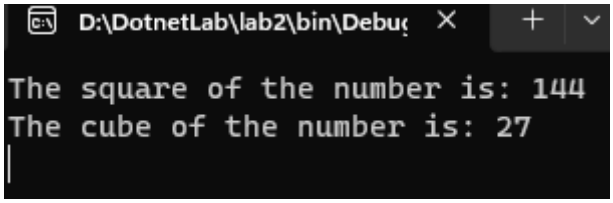
13)Create an interface named Num with two functions int add (int x, int y) and int diff (int x, int y) then make a class Solve that implements that interface Num.

```
using System;
namespace lab2
{
interface Num
{
int  add(int x, int y);
int diff(int x, int y);
}
class solve: Num
{
public int add( int x, int y)
{
return x + y;

}
public int diff(int x, int y)
{
return x - y;
}
}
class Program
{
static void Main(string[] args)
{
solve solve = new solve();
int add = solve.add(12, 12);
int diff = solve.diff(12, 4);
Console.WriteLine("The Addition is: "+add);
Console.WriteLine("The Difference is: " +diff);
}
}
}
```



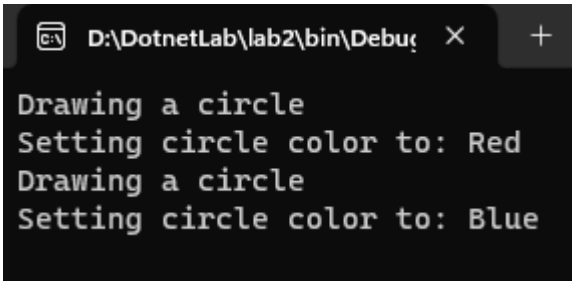
14)Create an interface called Number with two abstract methods int square (int x) and intcube (int x). Then create a class NumberDemo which implements Number interface and overrides the methods. Now create NumberDemo class and show interface implementation.

```
using System;
namespace lab2
{

interface Number
{
int Square(int x);
int Cube(int x);
}

class NumberDemo : Number
{
public int Square(int x)
{
return x * x;
}

public int Cube(int x)
{
return x * x * x;
}
}
class Program
{
static void Main(string[] args)
{
NumberDemo obj = new NumberDemo();
int sqrt = obj.Square(12);
int cube =obj.Cube(3);
Console.WriteLine("The square of the number is: "+sqrt);
Console.WriteLine("The cube of the number is: " + cube);
Console.ReadKey();
}
}
}
```



15)Demonstrate multiple inheritance using interface

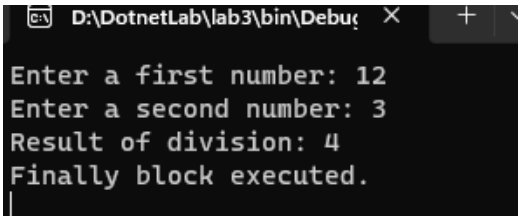
```
using System;
namespace lab2
{
interface IShape
{
void Draw();
}
interface IColor
{
void SetColor(string color);
}
class Circle : IShape, IColor
{
private string color;
public void Draw()
{
Console.WriteLine("Drawing a circle");
}
public void SetColor(string color)
{
this.color = color;
Console.WriteLine("Setting circle color to: " + color);
}
}
class Program
{
static void Main(string[] args)
{            Circle circle = new Circle();
circle.Draw();
circle.SetColor("Red");
IShape shape = circle;
shape.Draw();
IColor color = circle;
color.SetColor("Blue");

}
}
}
```

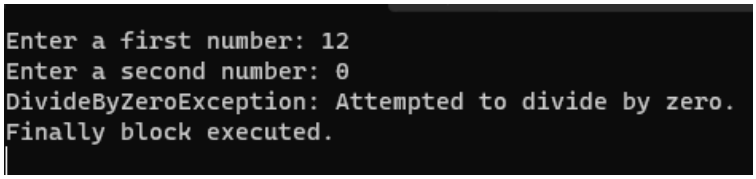


Practical 3 (Delegate and Event, Database, Exception Handling)

1. Demonstrate the use of try… catch?

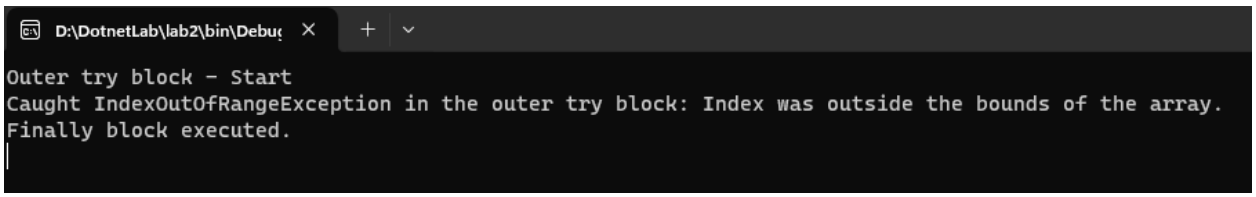
```
using System;
namespace lab3
{
internal class Program
{
static void Main(string[] args)
{
try
{

Console.Write("Enter a first number: ");
string number = Console.ReadLine();
Console.Write("Enter a second number: ");
string num1 = Console.ReadLine();
int number1 = int.Parse(number);
int number2 = int.Parse(num1);
int result = number1 / number2;
Console.WriteLine("Result of division: " + result);
}
catch (FormatException ex)
{

Console.WriteLine("FormatException: " + ex.Message);
}
catch (DivideByZeroException ex)
{
Console.WriteLine("DivideByZeroException: " + ex.Message);
}
catch (Exception ex)
{
Console.WriteLine("Exception: " + ex.Message);
}
finally
{
Console.WriteLine("Finally block executed.");
}

Console.ReadKey();
}
}
}
```





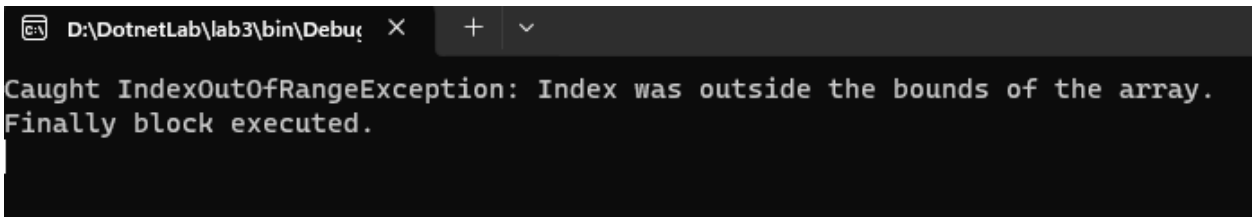
2)Demonstrate the use of nested try catch.

```
using System;
namespace lab2
{
class Program
{
static void Main(string[] args)
{
try
{
Console.WriteLine("Outer try block - Start");
int[] numbers = { 1, 2, 3 };
Console.WriteLine("Attempting to access an element outside the array bounds: " +
numbers[5]);
try
{
Console.WriteLine("Inner try block - Start");
int result = DivideNumbers(10, 0);
Console.WriteLine("Result of division: " + result);
Console.WriteLine("Inner try block - End");
}
catch (DivideByZeroException innerEx)
{
Console.WriteLine("Caught DivideByZeroException in the inner try block: " +
innerEx.Message);
}
Console.WriteLine("Outer try block - End");
}
catch (IndexOutOfRangeException outerEx)
{
Console.WriteLine("Caught IndexOutOfRangeException in the outer try block: " +
outerEx.Message);
}
catch (Exception ex)
{
Console.WriteLine("Caught general Exception: " + ex.Message);
}
finally
{
Console.WriteLine("Finally block executed.");
}
Console.ReadKey();
}
}
}
```



3)Demonstrate the use of multiple catch statement.

```
using System;
namespace lab3
{
internal class Program
{
static void Main(string[] args)
{
try
{
// Simulate a situation where an exception may occur
int[] numbers = { 1, 2, 3 };
Console.WriteLine("Attempting to access an element outside the array bounds: " +
numbers[5]);
}
catch (IndexOutOfRangeException ex)
{
// Catch and handle IndexOutOfRangeException
Console.WriteLine("Caught IndexOutOfRangeException: " + ex.Message);
}
catch (DivideByZeroException ex)
{
// Catch and handle DivideByZeroException
Console.WriteLine("Caught DivideByZeroException: " + ex.Message);
}
catch (Exception ex)
{
// Catch and handle any other exceptions
Console.WriteLine("Caught general Exception: " + ex.Message);
}
finally
{
// The code in the finally block will be executed regardless of whether an exception
occurred or not
Console.WriteLine("Finally block executed.");
}

Console.ReadKey();
}
}
}
```



4)Demonstrate how custom exception can be made using Exception class.

```
using System;
namespace lab3
{
class AgeException : Exception
{
// if age is less than 16 throw your error..
public AgeException(string s) : base(s)
{
Console.WriteLine("age cannot be less the 16..");
}
}
class OwnException
{
public void sendYourage(int age)
{
if (age < 16)
{
//throw own exception
throw new AgeException("age  can't be less than 16");
}
else
{
Console.WriteLine("your are eligible to participate");
}
}
}
class Program
{
static void Main(string[] args)
{
OwnException obj = new OwnException();
try
{
obj.sendYourage(14);
}
catch (AgeException a)
{
Console.WriteLine(a);
}

}
```

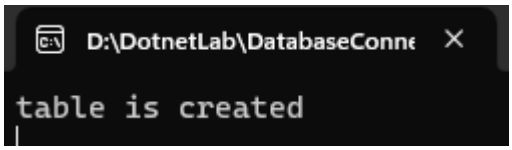
```
age cannot be less the 16..
lab3.AgeException: age  can't be less tha 16
   at lab3.OwnException.sendYourage(Int32 age) in D:\DotnetLab\lab3\OwnException.cs:line 20
   at lab3.Program.Main(String[] args) in D:\DotnetLab\lab3\Program.cs:line 12
```

1. Write a program to create a table “tbl_reg” having fields id primary key, username, password, repassword, gender, course, country. After this perform following:

```
using System;
using System.Data.SqlClient;
using System.Data;
namespace DatabaseConnection
{
internal class CreateTable
{
public void TableCreate()
{
string cs = "Data Source =LAPTOP-8KI500RH\\SQLEXPRESS;Initial Catalog = db_swsc;
Integrated Security = true;";

try
{
SqlConnection sc = new SqlConnection(cs);

if (sc.State == ConnectionState.Open)
{
Console.WriteLine("server connected");
}
sc.Open();
string tblquery = "create table tbl_reg(id int primary key, " + " username
varchar(50)," + " password varchar(50)," + " repassword varchar(50)," + " gender
varchar(50)," + "  course varchar(50), " + " country varchar(50))";
SqlCommand cmd = new SqlCommand(tblquery, sc);

int res = cmd.ExecuteNonQuery();
if (res > 0)
{
Console.WriteLine("table created");
}
}
catch (SqlException e)
{
Console.WriteLine(e);
}
}
}
class program
{
public static void Main(String[] args)
{

CreateTable obj = new CreateTable();
obj.TableCreate()
}
}
```

id	username	password	repassword	gender	course	country

}



b)Insert any 5 record by taking input from user

```
using System;
using System.Data.SqlClient;
using System.Data;
namespace DatabaseConnection
{
internal class InsertOperation
{
public void insert()
{
string cs = "Data Source =LAPTOP-8KI500RH\\SQLEXPRESS;Initial Catalog = db_swsc;
Integrated Security = true;";
try
{
SqlConnection sc = new SqlConnection(cs);

if (sc.State == ConnectionState.Open)
{
Console.WriteLine("server connected");
}
sc.Open();
Console.WriteLine("Enter your id:");
string id = Console.ReadLine();
Console.WriteLine("Enter your name:");
string name = Console.ReadLine();
Console.WriteLine("Enter your password:");
string password = Console.ReadLine();
Console.WriteLine("Enter your repasword:");
string repassword = Console.ReadLine();
Console.WriteLine("Enter your gender:");
string gender = Console.ReadLine();
Console.WriteLine("Enter your course:");
string course = Console.ReadLine();
Console.WriteLine("Enter your country:");
string country = Console.ReadLine();

//query
StringinsQuery= "insert into tbl_reg
values(@id,@username,@password,@repassword,@gender,@course,@country)";
SqlCommand cmd = new SqlCommand(insQuery, sc);
cmd.Parameters.AddWithValue("@id", id);
cmd.Parameters.AddWithValue("@username", name);
cmd.Parameters.AddWithValue("@password", password);
cmd.Parameters.AddWithValue("@repassword", repassword);
cmd.Parameters.AddWithValue("@gender", gender);
cmd.Parameters.AddWithValue("@course", course);
cmd.Parameters.AddWithValue("@country", country);

int res = cmd.ExecuteNonQuery();
```

```
if (res > 0)
{
Console.WriteLine("Data inserted");
}
sc.Close();
}
catch (SqlException s)
{
Console.WriteLine(s);
}
}
}
class program
{
public static void Main(String[] args)
{

InsertOperation obj = new InsertOperation();
obj.insert();
Console.ReadKey();
}
}
}
```

```
Enter your id:
5
Enter your name:
Rita kc
Enter your password:
rita12
Enter your repasword:
rita12
Enter your gender:
female
Enter your course:
BBM
Enter your country:
India
Data inserted
```

	id	username	password	repassword	gender	course	country
1	1	Naresh Khatri	Naresh@12	Naresh@12	Male	BCA	Nepal
2	2	Bisham Thapa	Bisha@12	Bisham@12	Male	BCA	Nepla
3	3	Ram kc	Ram@12	Ram@12	Male	BCA	Nepal
4	4	Sita shahi	sita12	sita12	female	BIT	Nepal
5	5	Rita kc	rita12	rita12	female	BBM	India

c) Display all the record from tbl_reg.

```
using System;
using System.Data.SqlClient;
using System.Data;
namespace DatabaseConnection
{
internal class DisplayOperation
{
public void display()
{
try
{
string cs = "Data Source =LAPTOP-8KI500RH\\SQLEXPRESS;Initial Catalog = db_swsc;
Integrated Security = true;";
SqlConnection sc = new SqlConnection(cs);

if (sc.State == ConnectionState.Open)
{
Console.WriteLine("server connected");
}
sc.Open();
string disQuery = "select *from tbl_reg";
SqlCommand cmd = new SqlCommand(disQuery, sc);
// for display executeReader should be used to fetch data
// it return SqlDataReader.
SqlDataReader row = cmd.ExecuteReader();
while (row.Read())
{
Console.WriteLine("id is " + row["id"]);
Console.WriteLine("username is " + row["username"]);
Console.WriteLine("password is " + row["password"]);
Console.WriteLine("repassword is " + row["repassword"]);
Console.WriteLine("gender is " + row["gender"]);
Console.WriteLine("course is " + row["course"]);
Console.WriteLine("country is " + row["country"]);
Console.WriteLine("-----------------------------------");
}
}
catch (SqlException ex)
{
Console.WriteLine(ex);
}
}
}
class program
{
public static void Main(String[] args)
{
DisplayOperation obj = new DisplayOperation();
```

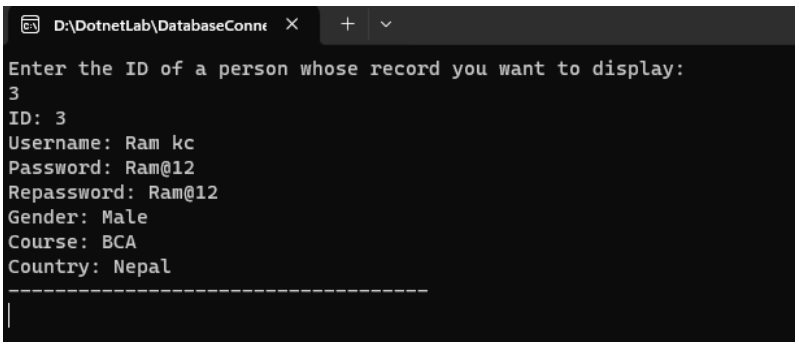
```
D:\DotnetLab\DatabaseConne
id is 1
username is Naresh Khatri
password is Naresh@12
repassword is Naresh@12
gender is Male
course is BCA
country is Nepal
-----------------------------------
id is 2
username is Bisham Thapa
password is Bisha@12
repassword is Bisham@12
gender is Male
course is BCA
country is Nepla
-----------------------------------
id is 3
username is Ram kc
password is Ram@12
repassword is Ram@12
gender is Male
course is BCA
country is Nepal
-----------------------------------
id is 4
username is Sita shahi
password is sita12
repassword is sita12
gender is female
course is BIT
country is Nepal
-----------------------------------
id is 5
username is Rita kc
password is rita12
repassword is rita12
gender is female
course is BBM
country is India
-----------------------------------
```

```
obj.display();
Console.ReadKey();
}
```

d) Display only the record of person whose id is given by user

```
using System;
using System.Data.SqlClient;

namespace DatabaseConnection
{
internal class DisplayById
{
public void ShowDetailsById()
{
string       cs       =       "Data       Source=LAPTOP-8KI500RH\\SQLEXPRESS;Initial
Catalog=db_swsc;Integrated Security=true;";
try
{
using (SqlConnection sc = new SqlConnection(cs))
{
sc.Open();
Console.WriteLine("Enter the ID of a person whose record you want to display:");
string id = Console.ReadLine();

// Select Query...
string displayQuery = "SELECT * FROM tbl_reg WHERE id = @id";
SqlCommand cmd = new SqlCommand(displayQuery, sc);
cmd.Parameters.AddWithValue("@id", id);

SqlDataReader row = cmd.ExecuteReader();
while (row.Read())
{
Console.WriteLine("ID: " + row["id"]);
Console.WriteLine("Username: " + row["username"]);
Console.WriteLine("Password: " + row["password"]);
Console.WriteLine("Repassword: " + row["repassword"]);
Console.WriteLine("Gender: " + row["gender"]);
Console.WriteLine("Course: " + row["course"]);
Console.WriteLine("Country: " + row["country"]);
Console.WriteLine("-----------------------------------");
}
}
```

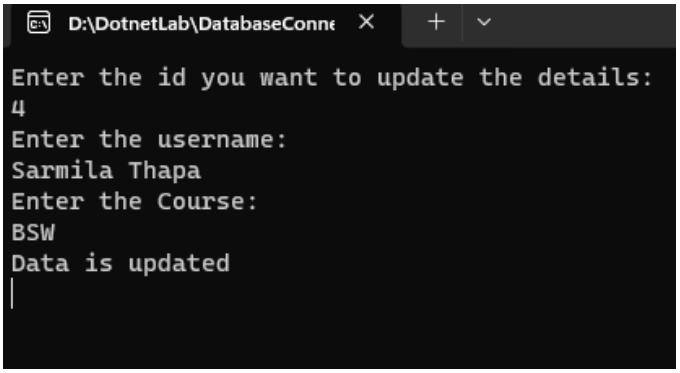
```
}
catch (SqlException e)
{
Console.WriteLine(e);
}
}
}
class program
{
public static void Main(String[] args)
{
DisplayById obj = new DisplayById();
obj.ShowDetailsById();
}
```



a. Update name and course of person to choice made by user according to id provide by user

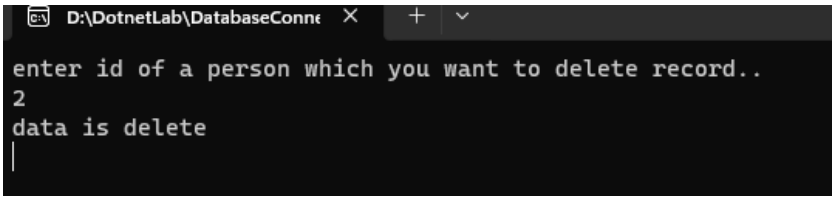
```
using System;
using System.Data.SqlClient;
using System.Data;

namespace DatabaseConnection
{
internal class UpdateOperation
{
public void update()
{
try
{
string cs = "Data Source=LAPTOP-8KI500RH\\SQLEXPRESS;Initial
Catalog=db_swsc;Integrated Security=true;";
SqlConnection sc = new SqlConnection(cs);

if (sc.State == ConnectionState.Open)
{
Console.WriteLine("Server connected");
}
sc.Open();
Console.WriteLine("Enter the id you want to update the details: ");
string uplid = Console.ReadLine();
Console.WriteLine("Enter the username: ");
string upusername = Console.ReadLine();
Console.WriteLine("Enter the Course: ");
string upcourse = Console.ReadLine();

string upQuery = "UPDATE tbl_reg SET username = @upusername, course = @upcourse WHERE
id = @uplid";
SqlCommand cmd = new SqlCommand(upQuery, sc);

cmd.Parameters.AddWithValue("@upusername", upusername);
cmd.Parameters.AddWithValue("@upcourse", upcourse);
cmd.Parameters.AddWithValue("@uplid", uplid);
```

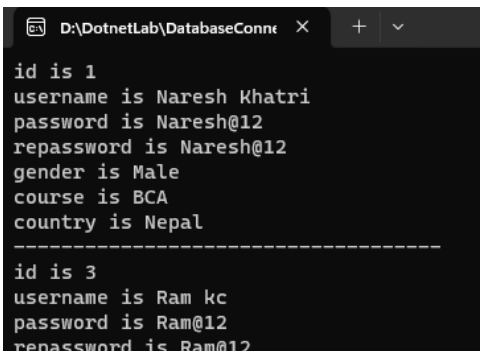
```
int res = cmd.ExecuteNonQuery();
if (res > 0)
{
Console.WriteLine("Data is updated");
}
}
catch (SqlException e)
{
Console.WriteLine(e);
}
}
}
class program
{
public static void Main(String[] args)
{ UpdateOperation obj = new UpdateOperation();
obj.update();
}
```



a. Delete the record of person according to id which is given by user

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DatabaseConnection
{
internal class DeleteById
{
public void Delete()
{
try
{
string cs = "Data Source =LAPTOP-8KI500RH\\SQLEXPRESS;Initial Catalog = db_swsc;
Integrated Security = true;";
SqlConnection sc = new SqlConnection(cs);

if (sc.State == ConnectionState.Open)
{
Console.WriteLine("server connected");
}
sc.Open();
//deleting the user information based on user input..
//query..
Console.WriteLine("enter id of a person which you want to delete record..");
string delid = Console.ReadLine();

string deleteQuery = "delete from tbl_reg where id =@id";
SqlCommand cmd = new SqlCommand(deleteQuery, sc);
cmd.Parameters.AddWithValue("@id", delid);
int res = cmd.ExecuteNonQuery();
```





```
if (res > 0)
{
Console.WriteLine("data is delete");
}

}
catch (SqlException e)
{
Console.WriteLine(e);
}
}
}
class program
{
public static void Main(String[] args)
{
DeleteById obj = new DeleteById();
obj.Delete();
}
}
}
```

3) Now create a login portal that ask username and password of user and if username and password matched with database record display the record of that person otherwise display “username or password is incorrect”. Use database and table name same as in question number 2.

```
using System;
using System.Data.SqlClient;
using System.Data;

namespace DatabaseConnection
{
internal class LoginPortal
{
public void LoginProtocol()
{
try
{
string       cs       =       "Data       Source=LAPTOP-8KI500RH\\SQLEXPRESS;Initial
Catalog=db_swsc;Integrated Security=true;";

using (SqlConnection sc = new SqlConnection(cs))
{
if (sc.State == ConnectionState.Open)
{
Console.WriteLine("Server connected");
}
sc.Open();
Console.Write("Enter username: ");
string username = Console.ReadLine();
Console.Write("Enter password: ");
string password = Console.ReadLine();
string loginQuery = "SELECT * FROM tbl_reg WHERE username = @username AND password =
@password";
using (SqlCommand cmd = new SqlCommand(loginQuery, sc))
{
cmd.Parameters.AddWithValue("@username", username);
```

```
cmd.Parameters.AddWithValue("@password", password);
using (SqlDataReader row = cmd.ExecuteReader())
{
while (row.Read())
{
Console.WriteLine("Login Successfully....");
Console.WriteLine("--------------Your Details--------");
Console.WriteLine("ID: " + row["id"]);
Console.WriteLine("Username: " + row["username"]);
Console.WriteLine("Password: " + row["password"]);
Console.WriteLine("Gender: " + row["gender"]);
Console.WriteLine("Course: " + row["course"]);
Console.WriteLine("Country: " + row["country"]);
}
}
}
}
}
catch (SqlException e)
{
Console.WriteLine(e);
}
}
}
class program
{
public static void Main(String[] args)
{
LoginPortal obj = new LoginPortal();
obj.LoginProtocol();
}
```