Ans 1: The Polymorphism Feature (Method Overriding) of OOP will resemblance the situation.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Labreport

{

    internal class Program

    {

        public class RituPhoneNumber

        {

            public virtual void numberRitu()

            {

                Console.WriteLine("Ritu's NORMAL No.: 01675468787");

            }

        }

        public class RituOfficeNumber:RituPhoneNumber

        {

            public override void numberRitu()

            {

                Console.WriteLine("Ritu's OFFICE No.: 01454674566");

            }

        }

        static void Main(string[] args)

        {

            RituPhoneNumber phoneNumber=new RituPhoneNumber();

            phoneNumber.numberRitu();

            RituOfficeNumber officePhoneNumber=new RituOfficeNumber();

            officePhoneNumber.numberRitu();

        }

    }

}

Ans.2

using System;

namespace Exam\_files

{

class Program

{

static void Main(string[] args)

{

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

double number = Convert.ToDouble(Console.ReadLine());

double num = Math.Abs(number);

if (num < 0)

{

Console.WriteLine("Negative");

}

else if (number > 0 && number <=1000000)

{

Console.WriteLine("Postive");

}

else if (num> 1000000)

{

Console.WriteLine("Large");

}

else if (number > 0 && number < 1)

{

Console.WriteLine("Small");

}

else if (number == 0)

{

Console.WriteLine("Zero");

}

}

}

}

Ans3:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_Report

{

internal class Program

    {

        public class Address

        {

            public String Street;

            public String City;

            public String PostalCode;

            public String Country;

        }

        public class Person:Address

        {

            public String Name;

            public String phoneNumber;

            public String Email;

        }

        public class Student:Person

        {

            public Double number;

            public Double averageMark;

        }

        public class Professor : Person

        {

            public double salary;

        }

       public static void Main(String[] args)

        {

            Student Student1 = new Student();

            Student1.Name = "Imam";

            Student1.phoneNumber = "01114564564";

            Student1.Email = "imam69@iubat.edu";

            Student1.Street = "7/A, Uttara";

            Student1.City = "Dhaka";

            Student1.PostalCode = "1230";

            Student1.Country = "Bangladesh";

            Student1.number = 881;

            Student1.averageMark = 80.8;

            Console.WriteLine("Name: {0} \nPhone Number: {1}\nEmail: {2} \nStreet: {3}\nCity: {4}\nPostalCode: {5}\nCountry: {6}\nMarks: {7}\nAverage Mark: {8}", Student1.Name,Student1.phoneNumber, Student1.Email, Student1.Street, Student1.City, Student1.PostalCode, Student1.Country, Student1.number, Student1.averageMark);

        }

    }}

Ans 4: Class and Object features and Operator Overloading can resemblance this situation:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Labreport

{

    internal class Program

    {

        public class DhakaBank

        {

            public double depositAmount;

            public double withdrawnAmount;

            public double balance;

            public string Name;

            public string email;

            public DhakaBank( String Name, String email)

            {

                this.Name = Name;

                this.email = email;

                Console.WriteLine("\t\tDhaka Bank Ltd.");

                Console.WriteLine("\t\t Imam's Account");

            }

            public static DhakaBank operator -(DhakaBank d, DhakaBank w)

            {

                return d - w;

            }

        }

        static void Main(string[] args)

        {

            DhakaBank p1 = new DhakaBank( "Imam", "imam69@iubat.edu");

            Console.WriteLine("Deposit an Amount..... ");

            p1.depositAmount = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("Withdraw an Amount..... ");

            p1.withdrawnAmount = Convert.ToDouble(Console.ReadLine());

            p1.balance = p1.depositAmount - p1.withdrawnAmount;

            Console.WriteLine("\tBalance Info Sent to: {0}", p1.email);

            Console.WriteLine("Name: "+p1.Name);

            Console.WriteLine("Email: "+p1.email);

            Console.WriteLine("Last Deposit: {0}tk\nLast Withdrawn: {1}tk\nTotal Balance: {2}tk", p1.depositAmount, p1.withdrawnAmount, p1.balance);

        }

    }}

Ans. 5

using System;

using System.Text;

namespace Exam\_files

{

class Calculation

{

public void Total (float i, float j)

{

float k = (float)((i \* 0.25) + (j \* 0.75));

if (k <= 100 && k>=80)

{

Console.WriteLine("Grade A "+k);

}

else if (k <= 79 && k >= 70)

{

Console.WriteLine("Grade B " + k);

}

else if (k <= 69 && k >= 60)

{

Console.WriteLine("Grade C " + k);

}

else if (k <= 59 && k >= 50)

{

Console.WriteLine("Grade D " + k);

}

else

{

Console.WriteLine("Your Future is Dark lol "+k );

}

}

}

class Program

{

static void Main(string[] args)

{

float JSC\_Eng = Convert.ToInt32(Console.ReadLine());

float SSC\_Eng = Convert.ToInt32(Console.ReadLine());

Calculation obj = new Calculation();

obj.Total(JSC\_Eng, SSC\_Eng);

}

}

}

Ans 6.

using System;

using System.Collections.Generic;

using System.Text;

namespace LabFinalProblemPractice

{

class Shape

{

public double x { get; set; }

public double y { get; set; }

}

class Rectangle : Shape

{

public double width { get; set; }

public double hight { get; set; }

public Rectangle(double width, double hight)

{

this.width = width;

this.hight = hight;

}

public void area()

{

double result = width \* hight;

Console.WriteLine("The area of the rectangle: " + result);

}

}

class Ellipse : Shape

{

public double major\_axis { get; set; }

public double minor\_axis { get; set; }

public Ellipse(double major\_axis, double minor\_axis)

{

this.major\_axis = major\_axis;

this.minor\_axis = minor\_axis;

}

public void area()

{

double result = Math.PI \* major\_axis \* minor\_axis;

Console.WriteLine("The area of the ellipse: " + result);

}

}

class Triangle : Shape

{

public double sideOne { get; set; }

public double sideTwo { get; set; }

public double angle { get; set; }

public Triangle(double sideOne, double sideTwo, double angle)

{

this.sideOne = sideOne;

this.sideTwo = sideTwo;

this.angle = angle;

}

public void area()

{

double result = 0.5 \* sideOne \* sideTwo \* Math.Sin(angle);

Console.WriteLine("The area of the Triangle: " + result);

}

}

class Qus6

{

public static void Main(String[] args)

{

Rectangle rec = new Rectangle(1.2, 5.2);

rec.area();

Ellipse ell = new Ellipse(1.2, 5.2);

ell.area();

Triangle tri = new Triangle(6, 6, 40);

tri.area();

}

}

}

Ans.7

using System;

namespace Program7

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter Your Array Size");

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

int res;

int[] answer = new int[l];

Console.WriteLine("Enter Your Array");

for (int i = 0; i < answer.Length; i++)

{

answer[i] = Convert.ToInt32(Console.ReadLine());

}

Console.WriteLine("Enter Specified Number");

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

int count = 0;

for (int i = 0; i < answer.Length; i++)

{

for (int j = i; j < answer.Length - 1; j++)

{

res = answer[i] + answer[j + 1];

if (res == n)

{

count++;

if (count == 1)

{

Console.WriteLine("Pairs Of the elements and their sum: ");

}

Console.WriteLine(answer[i] + "+" + answer[j + 1] + " = " +

res);

break;

}

}

}

}

}

}

Ans 8:

The Polymorphism Feature (Method Overriding) of OOP will resemblance the situation.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Labreport

{

    internal class Program

    {

        public class RituPhoneNumber

        {

            public virtual void numberRitu()

            {

                Console.WriteLine("Ritu's NORMAL No.: 01675468787");

            }

        }

        public class RituOfficeNumber:RituPhoneNumber

        {

            public override void numberRitu()

            {

                Console.WriteLine("Ritu's OFFICE No.: 01454674566");

            }

        }

        static void Main(string[] args)

        {

            RituPhoneNumber phoneNumber=new RituPhoneNumber();

            phoneNumber.numberRitu();

            RituOfficeNumber officePhoneNumber=new RituOfficeNumber();

            officePhoneNumber.numberRitu();

        }

    }

}

ans. 9

using System;

namespace lab\_repoet\_1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("write 3 integer number =");

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

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

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

if(a > b && b > c && a>c)

{

Console.WriteLine("decresing");

}

else if(a < b && b < c && a < c)

{

Console.WriteLine("incresing");

}

else

{

Console.WriteLine("Give incresing or decresing number");

}

}

}

}

ans. 10

using System;

using System.Text;

namespace Exam\_files

{

class parent

{

public int i = 4;

public int j = 5;

public int k = 6;

}

//inheriting

class child : parent

{

public int l = 7;

}

class Program

{

static void Main(string[] args)

{

child obj = new child();

Console.WriteLine(obj.l);

Console.WriteLine(obj.i);

}

}

}

Explanation: Suppose you are the object and the two shops are two class. First one parent and the second shop child. Since the object can access special ice cream from 2nd shop but not the 1st shop means that you are an object of the child class. Which proves inheritance, because the 2nd shop has all the elements of 1st shop with an extra, but 1st shop doesn’t have the extra. You being the child of 2nd shop can access everything from 2nd shop and 1st shop.

Ans.11

using System;

using System.Text;

namespace Exam\_files

{

class Class

{

public void Batch()

{

Console.WriteLine("Batch");

}

}

class ClassOne

{

public void SectionA()

{

Console.WriteLine("SectionA");

}

}

class ClassTwo : ClassOne

{

public void SectionC()

{

Console.WriteLine("SectionC");

}

public void SectionD()

{

Console.WriteLine("SectionD");

}

}

class School : ClassTwo

{

}

class Program

{

static void Main(string[] args)

{

ClassOne obj1 = new ClassOne();

ClassTwo obj2 = new ClassTwo();

School obj3 = new School();

obj3.SectionD();

obj3.SectionC();

obj3.SectionA();

obj3.Batch();

}

}

}

We couldn’t access Batch from class as there is no other for School to inherit multiple classes. ***Remove obj3.Batch*** to run code.