**Answer the following**

**1. A farmer is asking you to tell him how many legs can be counted among all his animals. The**

**farmer breeds three species:**

**chickens = 2 legs**

**cows = 4 legs**

**pigs = 4 legs**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{ Console.WriteLine("Enter the Chickens count :");

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

Console.WriteLine("Enter the Cows count :");

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

Console.WriteLine("Enter the Pigs count :");

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

AnimalCalculation animalCalculation = new AnimalCalculation(chick, Cow, Pig);

Console.WriteLine(animalCalculation.Result());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class AnimalCalculation

{

private int chick;

private int cow;

private int pig;

public AnimalCalculation(int chick, int cow, int pig)

{

this.Chick = chick;

this.Cow = cow;

this.Pig = pig;

}

public int Chick { get => chick; set => chick = value; }

public int Cow { get => cow; set => cow = value; }

public int Pig { get => pig; set => pig = value; }

public int Result()

{

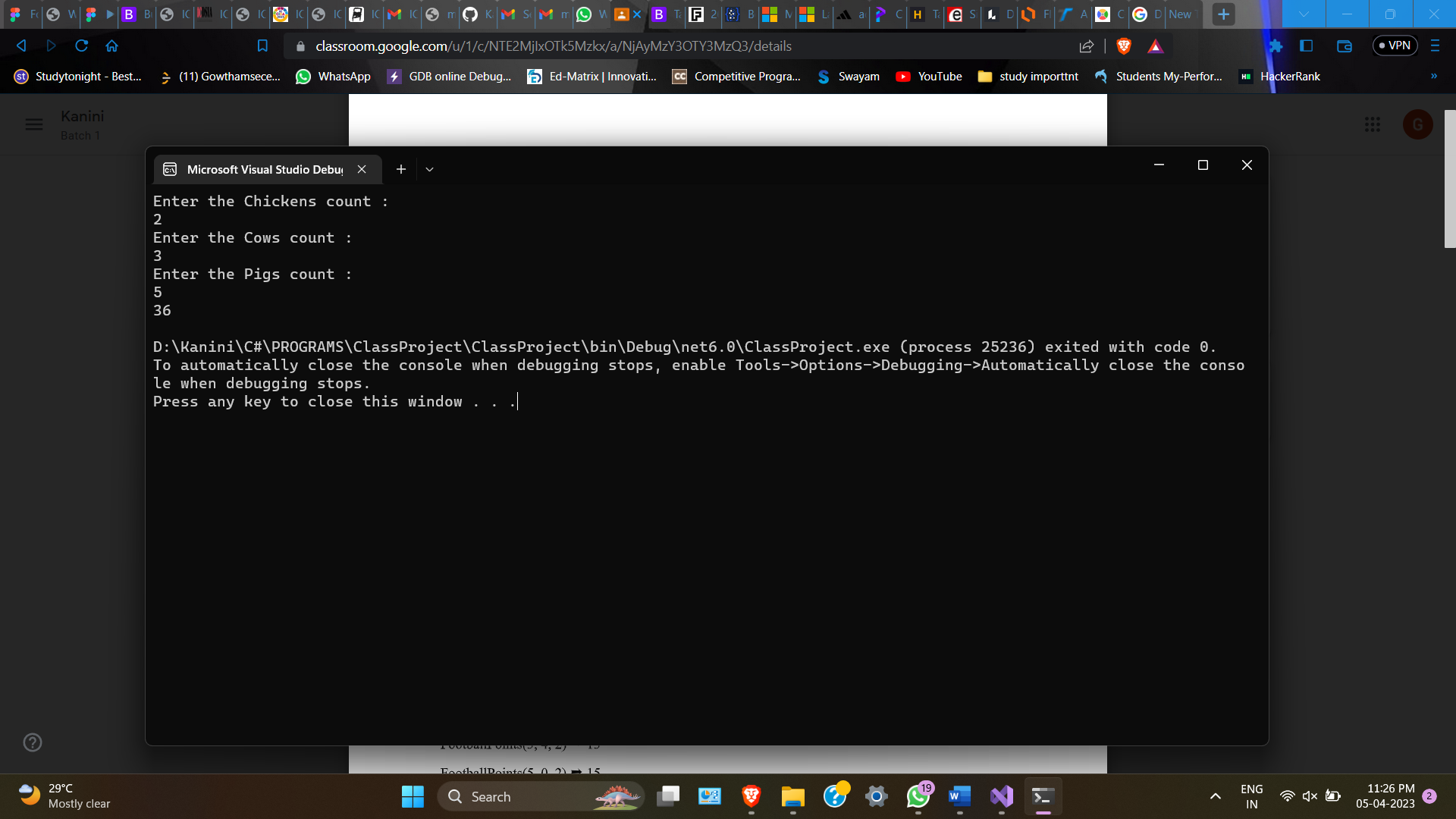
int count = this.Chick \* 2 + this.Cow \*4 + this.Pig \*4;

return count;

}

}

}



**2. Create a function that takes the number of wins, draws and losses and calculates the number of**

**points a football team has obtained so far.**

**wins get 3 points draws get 1 point losses get 0 points**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{

Console.WriteLine("Enter the Number of Wins :");

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

Console.WriteLine("Enter the Number of Draws :");

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

Console.WriteLine("Enter the NUmber of Losses :");

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

if(win>=0 && draw>=0 && loss>= 0)

{

FootballCalculation footballCalculation = new FootballCalculation(win, draw, loss);

Console.WriteLine(footballCalculation.Result());

}

else

{

Console.WriteLine("Enter the correct numbers by all");

}

}

}

}

FootballCalculation

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class FootballCalculation

{

private int win;

private int draw;

private int loss;

public FootballCalculation(int win, int draw, int loss)

{

this.Win = win;

this.Draw = draw;

this.Loss = loss;

}

public int Win { get => win; set => win = value; }

public int Draw { get => draw; set => draw = value; }

public int Loss { get => loss; set => loss = value; }

public int Result()

{

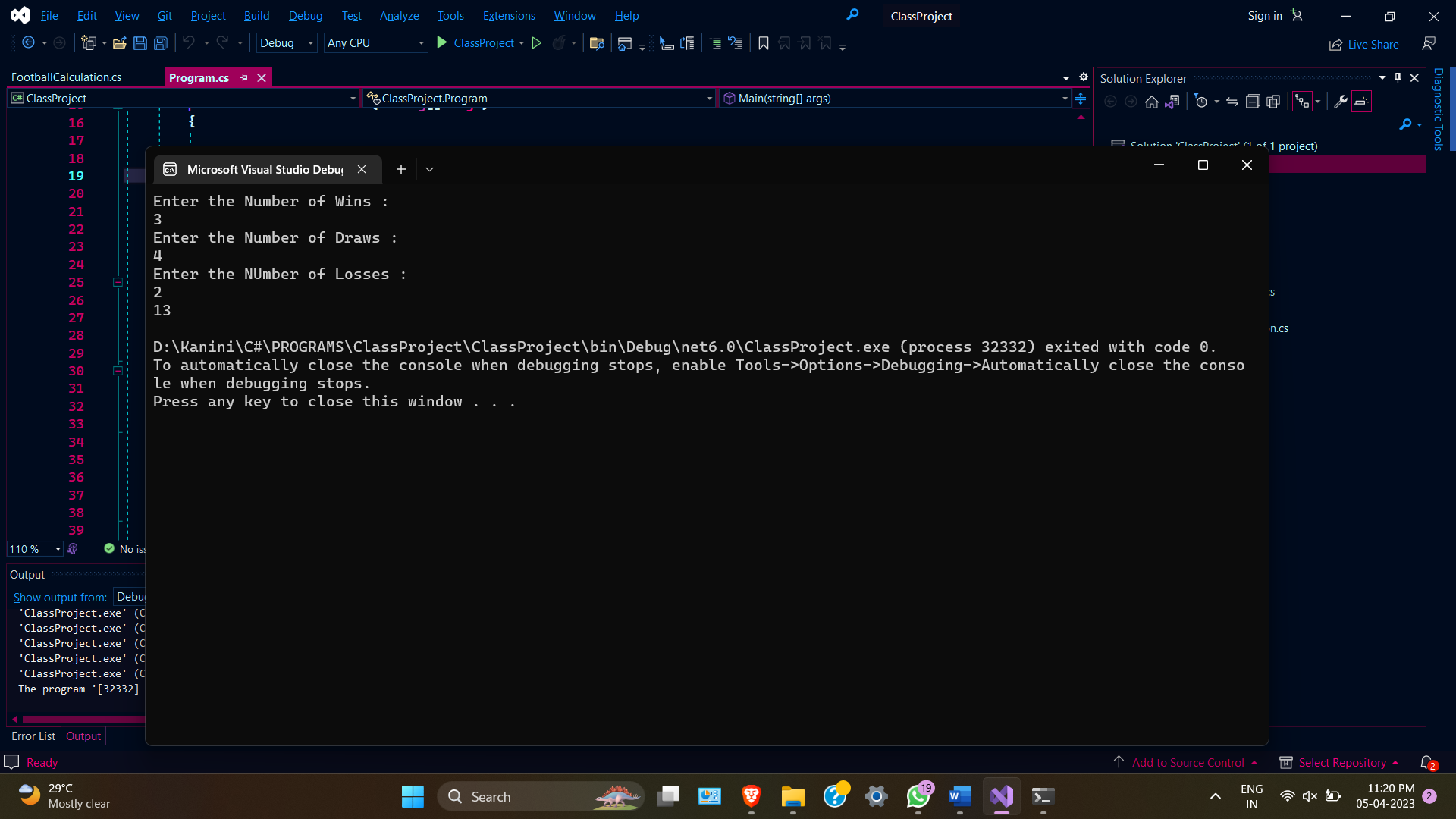
int count = this.Win\*3 + this.Draw\*1 + this.Loss\*0;

return count;

}

}

}



**3. Create a function that takes three arguments prob, prize, pay and returns true if prob \* prize &gt; pay;**

**otherwise return false.**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{

Console.WriteLine("Enter the prob");

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

Console.WriteLine("Enter the prize");

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

Console.WriteLine("Enter the pay");

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

Gamble gamble = new Gamble();

Console.WriteLine(gamble.ProfitableGamble(prob, prize, pay));

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class Gamble

{

private double prob;

private int prize;

private int pay;

public double Prob { get => prob; set => prob = value; }

public int Prize { get => prize; set => prize = value; }

public int Pay { get => pay; set => pay = value; }

public bool ProfitableGamble(double prob, int prize, int pay)

{

if (prob \* prize > pay)

{

return true;

}

else

{

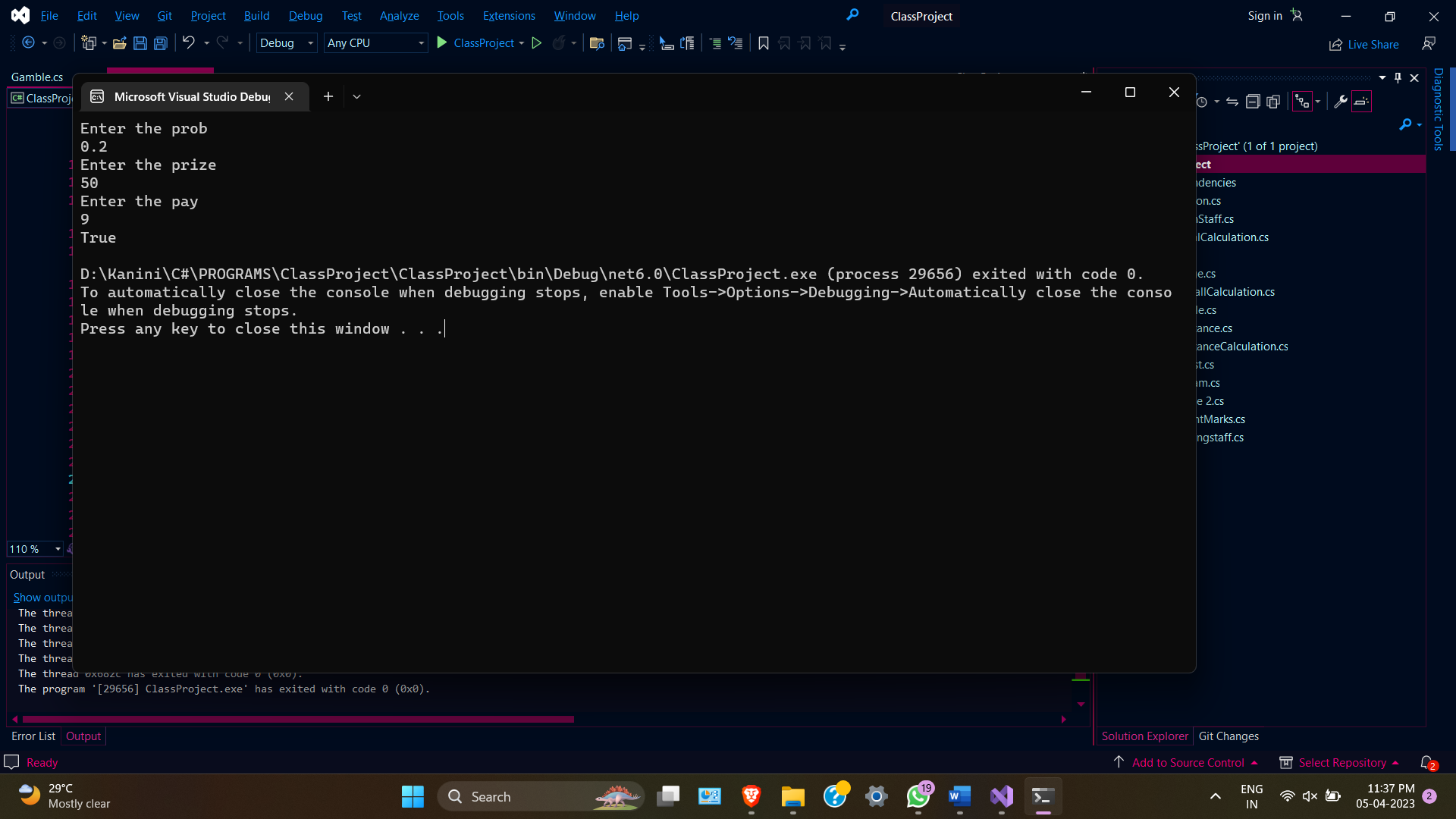
return false;

}

}

}

}



**4. Here&#39;s an image of four models. Some of the cubes are hidden behind other cubes. Model one**

**consists of one cube. Model two consists of four cubes, and so on...**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{

{

Console.WriteLine("Enter the number");

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

Box box = new Box();

Console.WriteLine("Number os stacked boxes : " + box.StackBoxes(number));

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class Box

{

private int number;

public int Number { get => number; set => number = value; }

public int StackBoxes(int number)

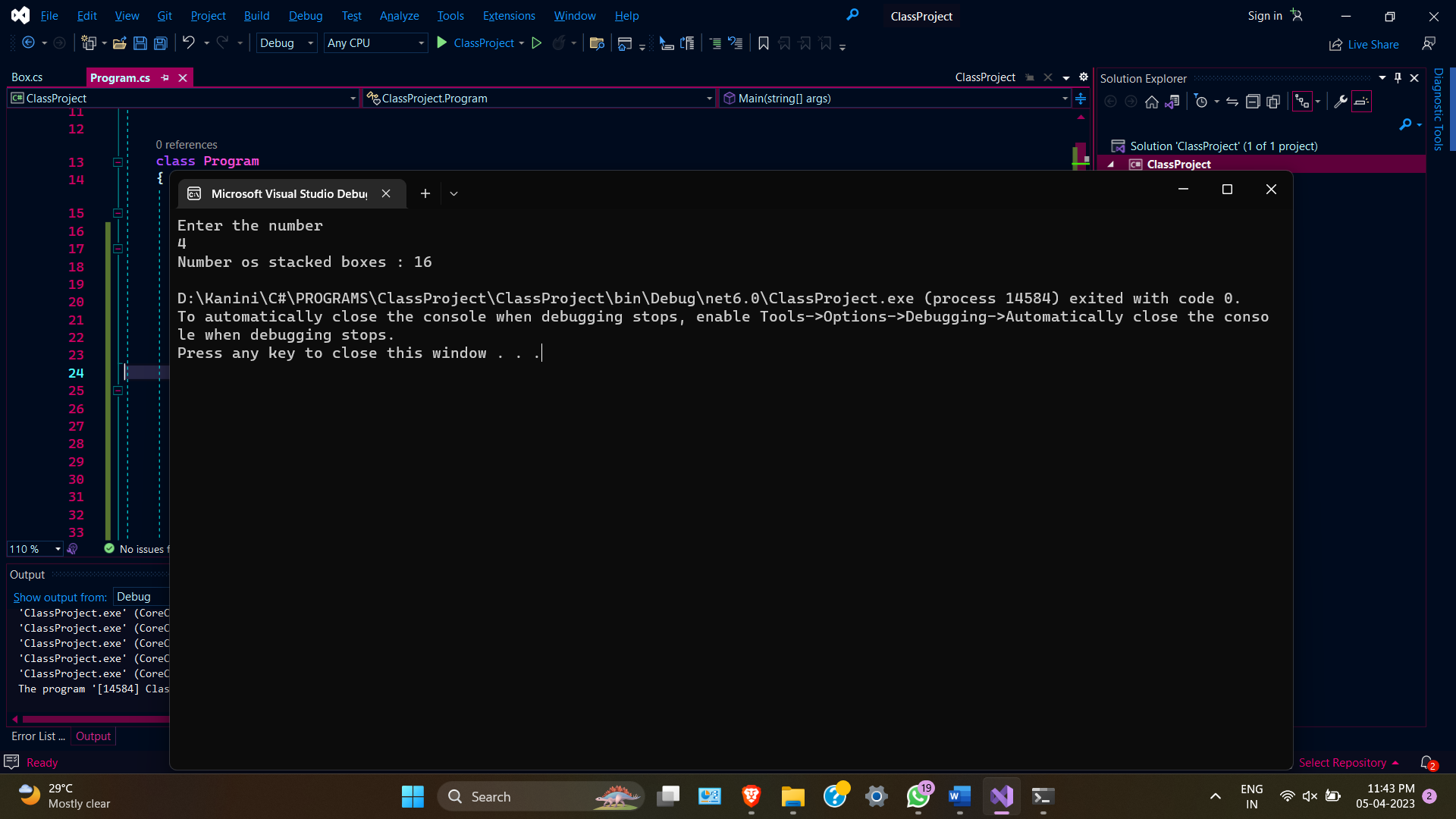
{

return number \* number;

}

}

}



**5. A bartender is writing a simple program to determine whether he should serve drinks to someone.**

**He only serves drinks to people 18 and older and when he&#39;s not on break.**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{

Console.WriteLine("Enter the age");

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

Console.WriteLine("Enter the breaktime");

bool breaktime = Convert.ToBoolean(Console.ReadLine());

Bartender bartender = new Bartender();

Console.WriteLine(bartender.ShouldServeDrinks(age, breaktime));

}

}  
}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class Bartender

{

private int age;

private bool breaktime;

public int Age { get => age; set => age = value; }

public bool Breaktime { get => breaktime; set => breaktime = value; }

public bool ShouldServeDrinks(int age, bool breaktime)

{

if ((age >= 18) && (breaktime == false))

{

return true;

}

else

{

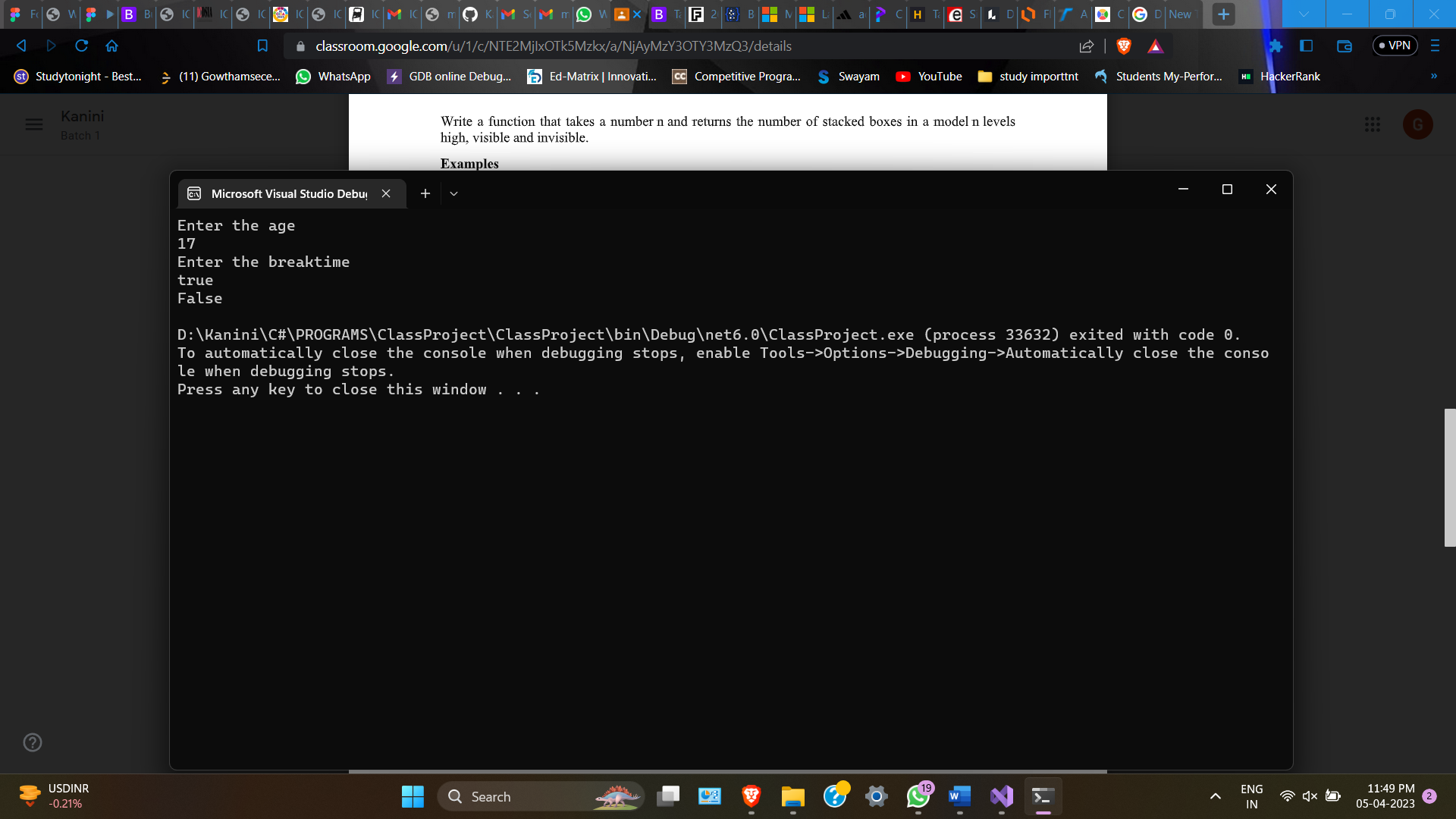
return false;

}

}

}

}



**6. For each of the 6 coffee cups I buy, I get a 7th cup free. In total, I get 7 cups. Create a function**

**that takes n cups bought and return the total number of cups I would get.**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{

Console.WriteLine("Enter the cup count");

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

Coffee coffee = new Coffee();

Console.WriteLine("Total Number of cups : " + coffee.TotalCups(cup));

**}**

**}**

**}**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class Coffee

{

private int cups;

public int Cups { get => cups; set => cups = value; }

public int TotalCups(int cups)

{

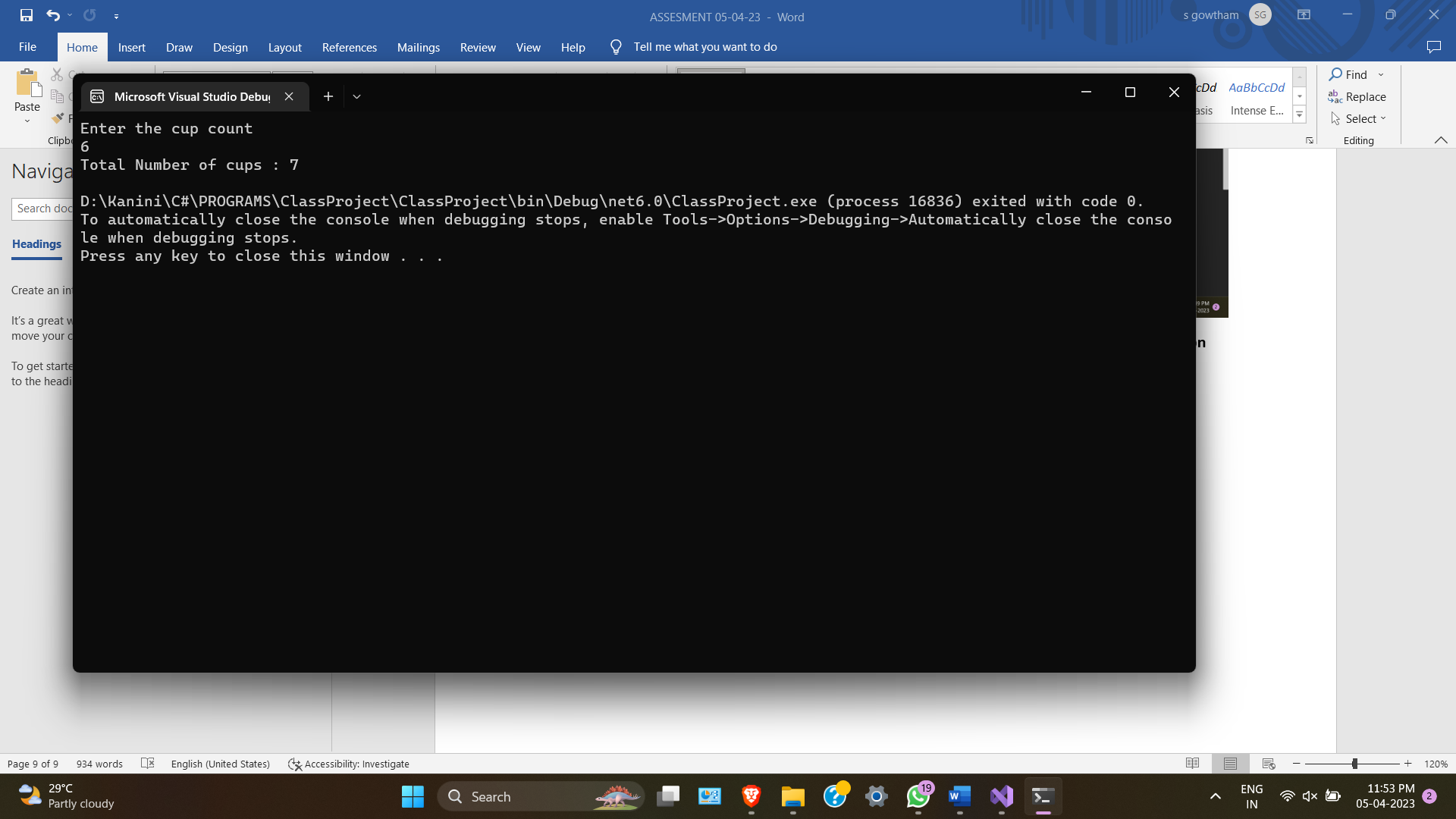
int total\_cups = (cups / 6) + cups;

return total\_cups;

}

}

}



**7. Create a function that adds a string ending to each member in an array.**

**CODE**

using ClassProject;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

class Program

{

public static void Main(String[] args)

{

Ending ending = new Ending();

string[] temp = { "clever", "meek", "hurried", "nice" };

ending.AddEnding(temp, "ly");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassProject

{

internal class Ending

{

public void AddEnding(string[] words, string addon)

{

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

{

words[i] = words[i] + addon;

}

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

{

Console.Write(words[i] + " ");

}

}

}

}

8. Create a function that returns how many possible arrangements can come from a certain number

of switches (on / off). In other words, for a given number of switches, how many different

patterns of on and off can we have?

CODE  
using Basic\_Program;

using System.Net.Http.Headers;

class Demo1

{

public static void Main(string[] args)

{

Console.WriteLine("Enter the number of switches");

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

Switch switches = new Switch();

Console.WriteLine(switches.PosCom(num));

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Basic\_Program

{

internal class Switch

{

public int PosCom(int num)

{

return (int)Math.Pow(2, num);

}

}

}

