1. **Write a simple console application to perform arithmetic operations**

**Code:**

using System;

namespace q1

{

class Program

{

static void Main(string[] args)

{

int num1, num2;

//Input

Console.WriteLine("Enter two numbers: ");

int.TryParse(Console.ReadLine(), out num1);

int.TryParse(Console.ReadLine(), out num2);

Console.WriteLine("\nChoose one of the following arithmetic operation:");

Console.WriteLine("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division.\n5. All\nChoice: ");

//Input of choice

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

//Switch case for the choice of arithmetic operation

switch (option)

{

case 1: Console.WriteLine("\n{0} + {1} = {2}", num1, num2, num1 + num2);

break;

case 2: Console.WriteLine("\n{0} - {1} = {2}", num1, num2, num1 - num2);

break;

case 3: Console.WriteLine("\n{0} \* {1} = {2}", num1, num2, num1 \* num2);

break;

case 4: Console.WriteLine("\n{0} / {1} = {2}", num1, num2, (double)num1 / num2);

break;

case 5: Console.WriteLine("\n{0} + {1} = {2}\n{0} - {1} = {3}\n{0} \* {1} = {4}\n{0} / {1} = {5}", num1, num2, num1 + num2, num1 - num2, num1 \* num2, (double)num1 / num2);

break;

default: Console.WriteLine("\nChoose a valid option");

break;

}

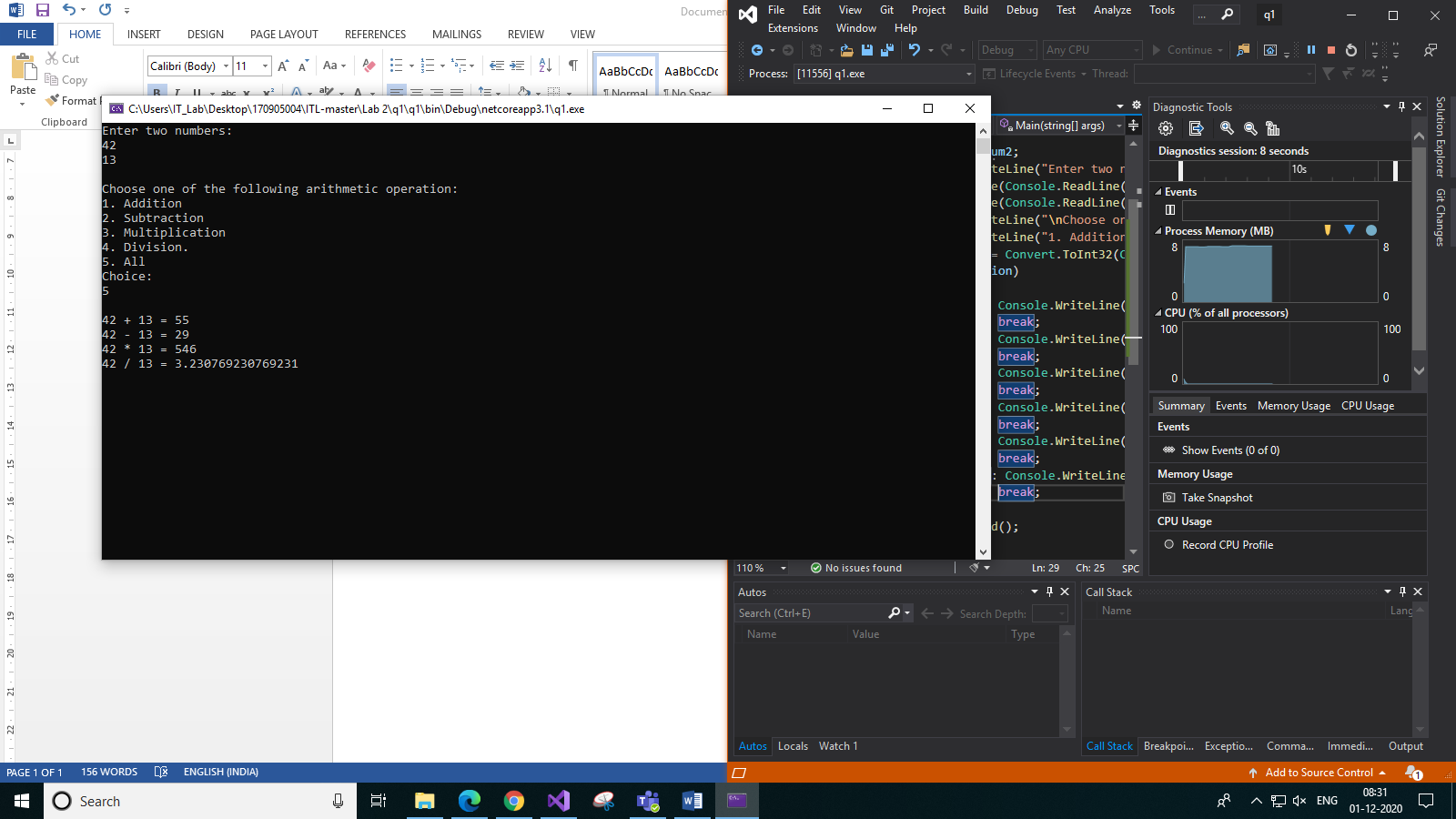
Console.Read();

}

}

}

**Output:**



1. **Write a Console C# program to realise DateTime.Add member function without using DateTime/TimeStamp instances. The inputs to the program are valid date (in the format "DD: MM: YY: hh:mm:ss") and # of ticks (range from 10000000 - 999999999999) which have to be accepted from the user. The output will be a number which is generated by adding the ticks to the given date. (Note: 1 tick = 100 nano secs)**

**Code:**

using System;

namespace q2

{

class Program

{

static bool isLeapYear(int year)

{

// If a year is multiple of 400, then it is a leap year

if (year % 400 == 0)

return true;

// Else If a year is multiple of 100, then it is not a leap year

if (year % 100 == 0)

return false;

// Else If a year is multiple of 4, then it is a leap year

if (year % 4 == 0)

return true;

return false;

}

static int NumOfDays(int month, int year)

{

switch (month)

{

case 2: if (isLeapYear(year))

{

return 29;

}

else

return 28;

case 1: case 3: case 5: case 7: case 8: case 10: case 12: return 31;

default: return 30;

}

}

static void Main(string[] args)

{

//Input

Console.WriteLine("Enter the date in the format \"DD: MM: YY: hh: mm: ss\": ");

string dateTime = Console.ReadLine();

Console.WriteLine("Enter the number of ticks: ");

long ticks;

long.TryParse(Console.ReadLine(), out ticks);

//Check if "ticks" is in range

if (ticks<10000000 || ticks > 999999999999)

{

Console.WriteLine("Not valid. Enter ticks within the range 10000000 and 999999999999");

return;

}

//Convert ticks to seconds

int seconds = Convert.ToInt32(ticks \* Math.Pow(10.0, -7.0));

//Convert the date taken as input to an integer array

string[] arr = dateTime.Split(": ");

int[] dateArray = Array.ConvertAll(arr, int.Parse);

//Calcuate new date

int minutes = 0, hours = 0, days = 0;

//If new ss is greater than 60, add it to minutes. The input will always be 1 second or more due to the contraint check.

if (seconds + dateArray[5] >= 60)

{

minutes = (seconds + dateArray[5]) / 60;

dateArray[5] = (seconds + dateArray[5]) % 60;

}

else

dateArray[5] += seconds;

//If new mm is greater than 60, add it to hours.

if (minutes>0 && minutes + dateArray[4] >= 60)

{

hours = (minutes + dateArray[4]) / 60;

dateArray[4] = (minutes + dateArray[4]) % 60;

}

else

dateArray[4] += minutes;

//If new hh is greater than 24, add it to days.

if (hours > 0 && hours + dateArray[3] >= 24)

{

days = (hours + dateArray[3]) / 24;

dateArray[3] = (hours + dateArray[3]) % 24;

}

else

dateArray[3] += hours;

//If new days is greater than 1, add it to DD. The ticks will never be greater than a day due to the constraint check.

//Check if adding day changes month

if (days>=1)

{

int numOfDays = NumOfDays(dateArray[1], dateArray[2]);

if (days + dateArray[0] > numOfDays)

{

dateArray[0] = 01;

if (dateArray[1] + 1 >= 12)

{

dateArray[1] = 01;

dateArray[2] = (dateArray[2] + 1) % 100;

}

}

else

dateArray[0] += days;

}

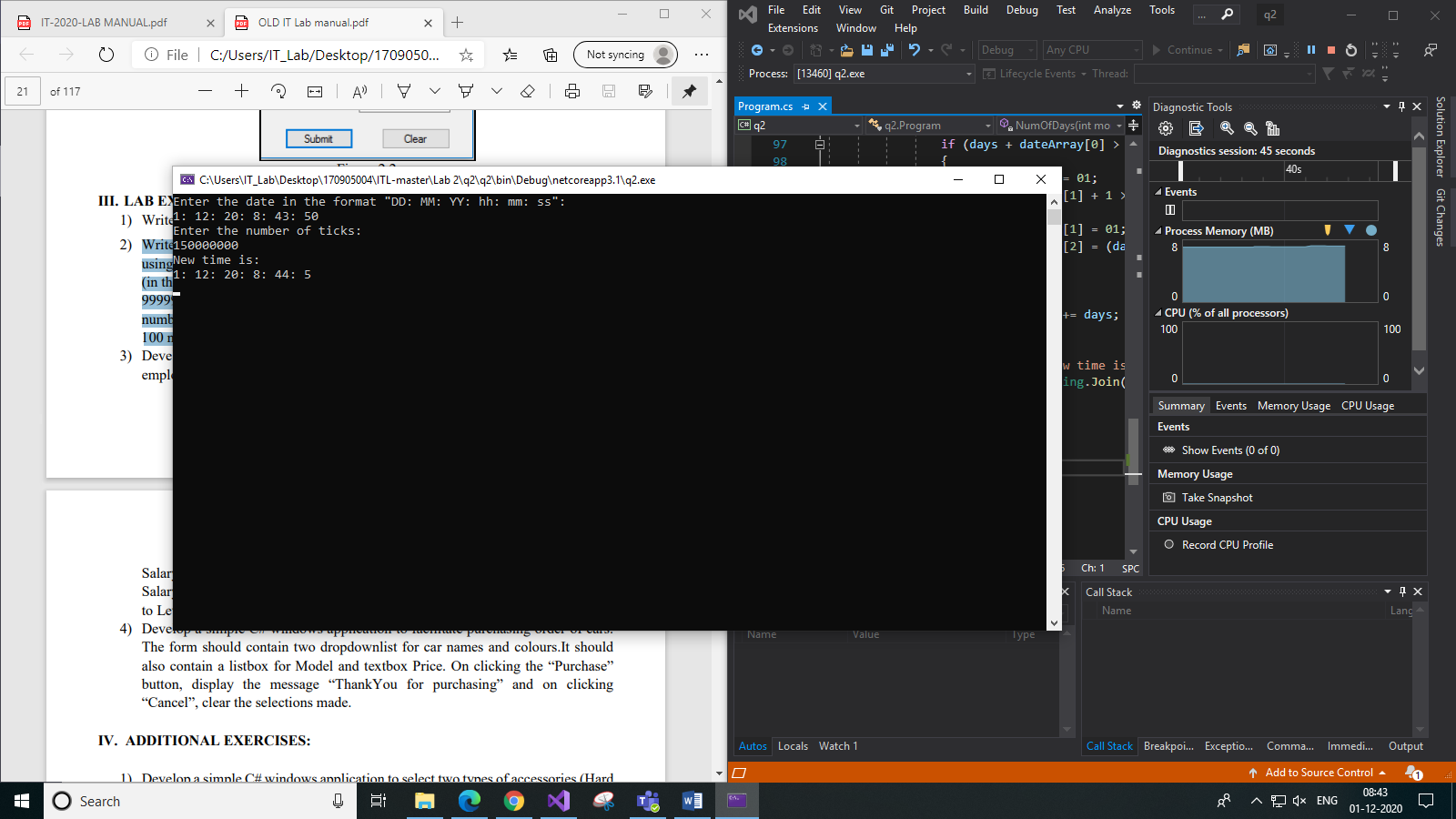
Console.WriteLine(String.Join(": ", dateArray));

}

}

}

**Output:**



1. **Develop a simple C# windows application to compute the bonus to be paid to an employee based on his performance level using a function. Use TextBox to input 20 Salary and ComboBox to select performance level. (Performance Level1 = 0.1 \* Salary, Level2 to Level4 = 0.09\*Salary, Level5 to Level7 = 0.07 \* Salary, Level8 to Level10 = 0.05 \* Salary)**

**Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace q3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

double salary;

double.TryParse(textBox1.Text, out salary);

string level = comboBox1.SelectedItem.ToString();

if (level == "Level1")

textBox2.Text = (0.1 \* salary).ToString();

else if (level == "Level2" || level == "Level3" || level == "Level4")

textBox2.Text = (0.09 \* salary).ToString();

else if (level == "Level5" || level == "Level6" || level == "Level7")

textBox2.Text = (0.07 \* salary).ToString();

else

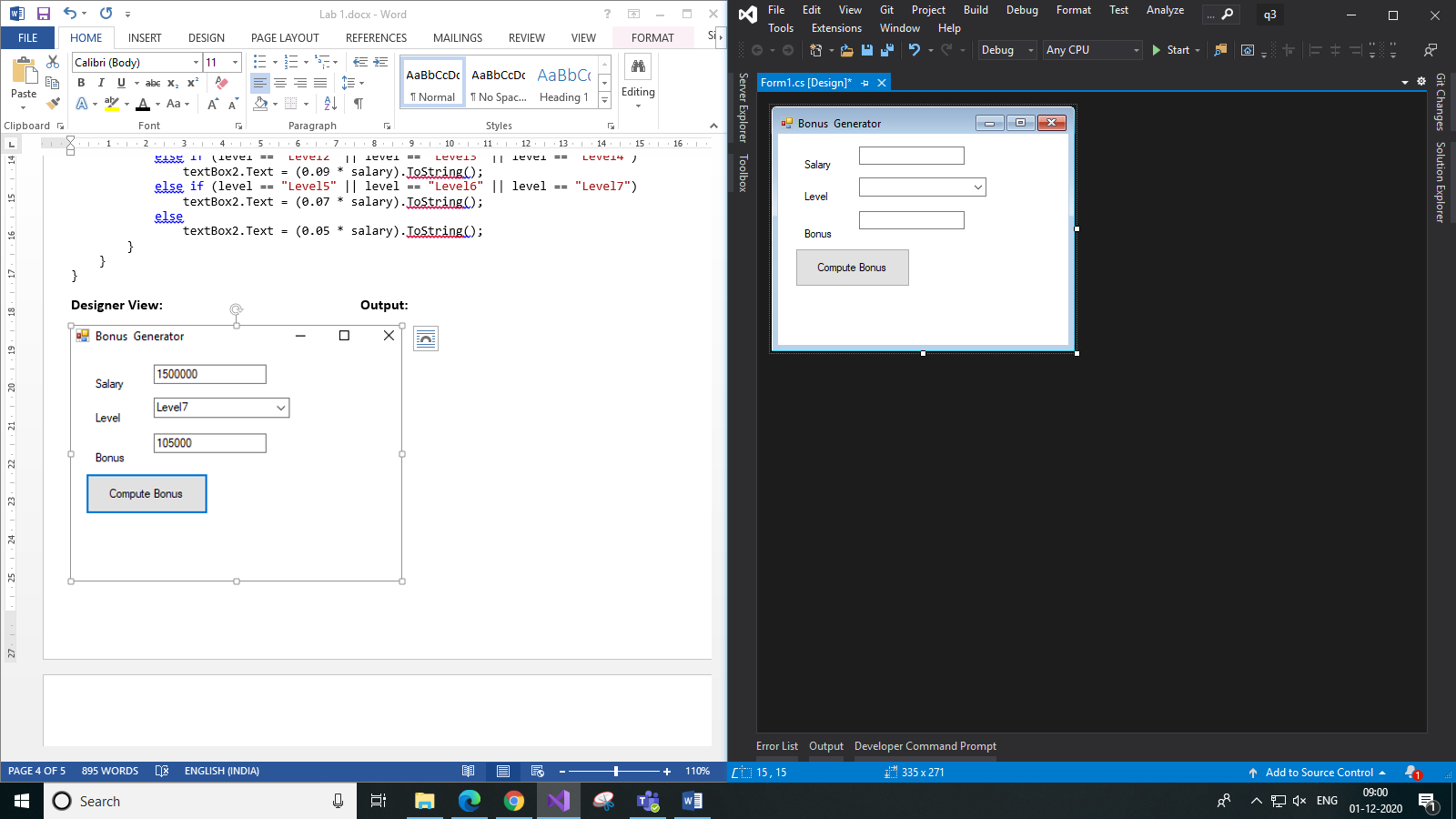
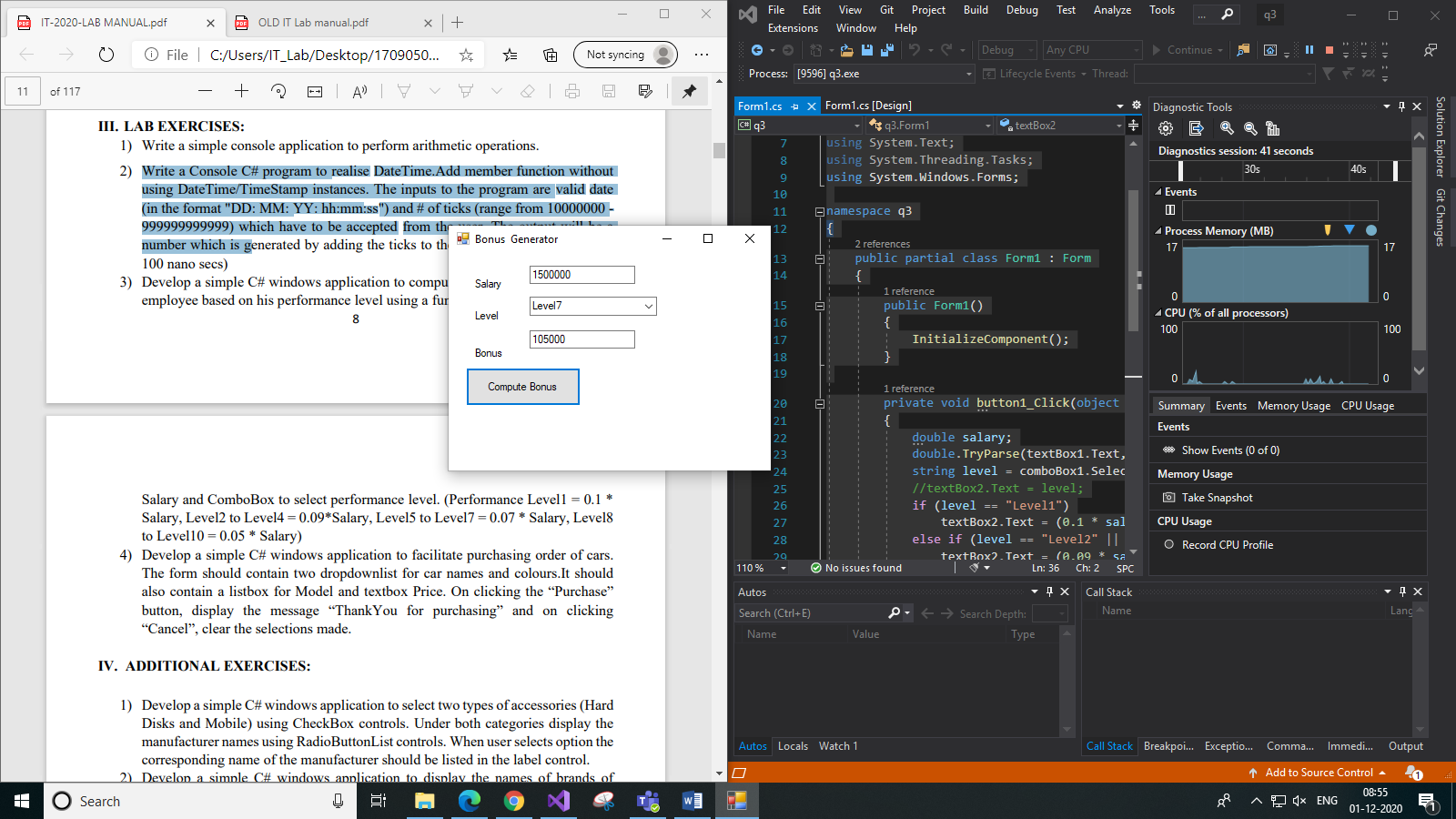
textBox2.Text = (0.05 \* salary).ToString();

}

}

}

**Designer View: Output:**

*(Designer view included with code as it is a windows application and does not have code for the design)*

1. **The form should contain two dropdownlist for car names and colours.It should also contain a listbox for Model and textbox Price. On clicking the “Purchase” button, display the message “ThankYou for purchasing” and on clicking “Cancel”, clear the selections made.**

**Code:**

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace q4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

label5.Text = "Thank you for purchasing!";

}

private void button2\_Click(object sender, EventArgs e)

{

comboBox1.SelectedIndex = -1;

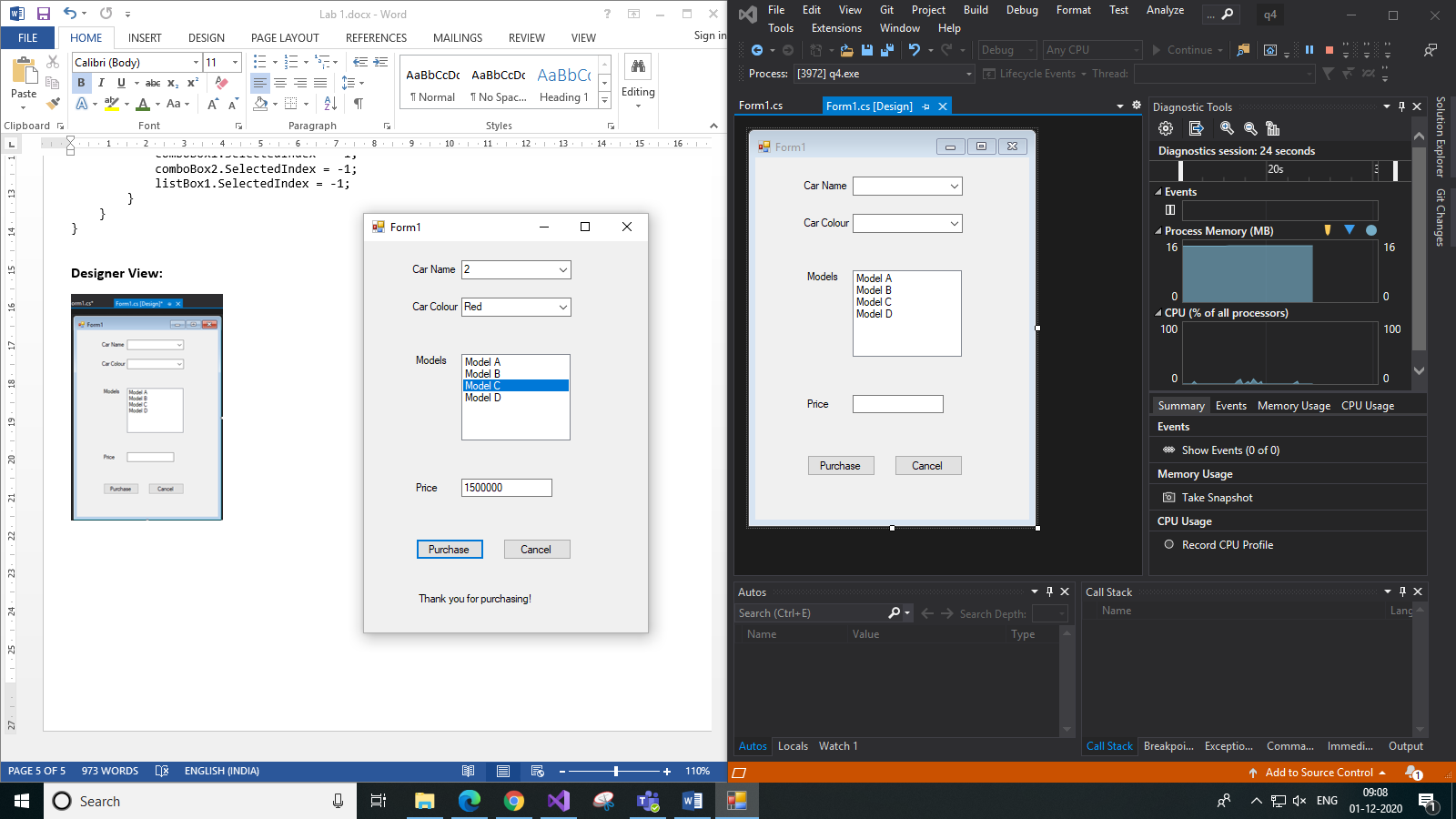
comboBox2.SelectedIndex = -1;

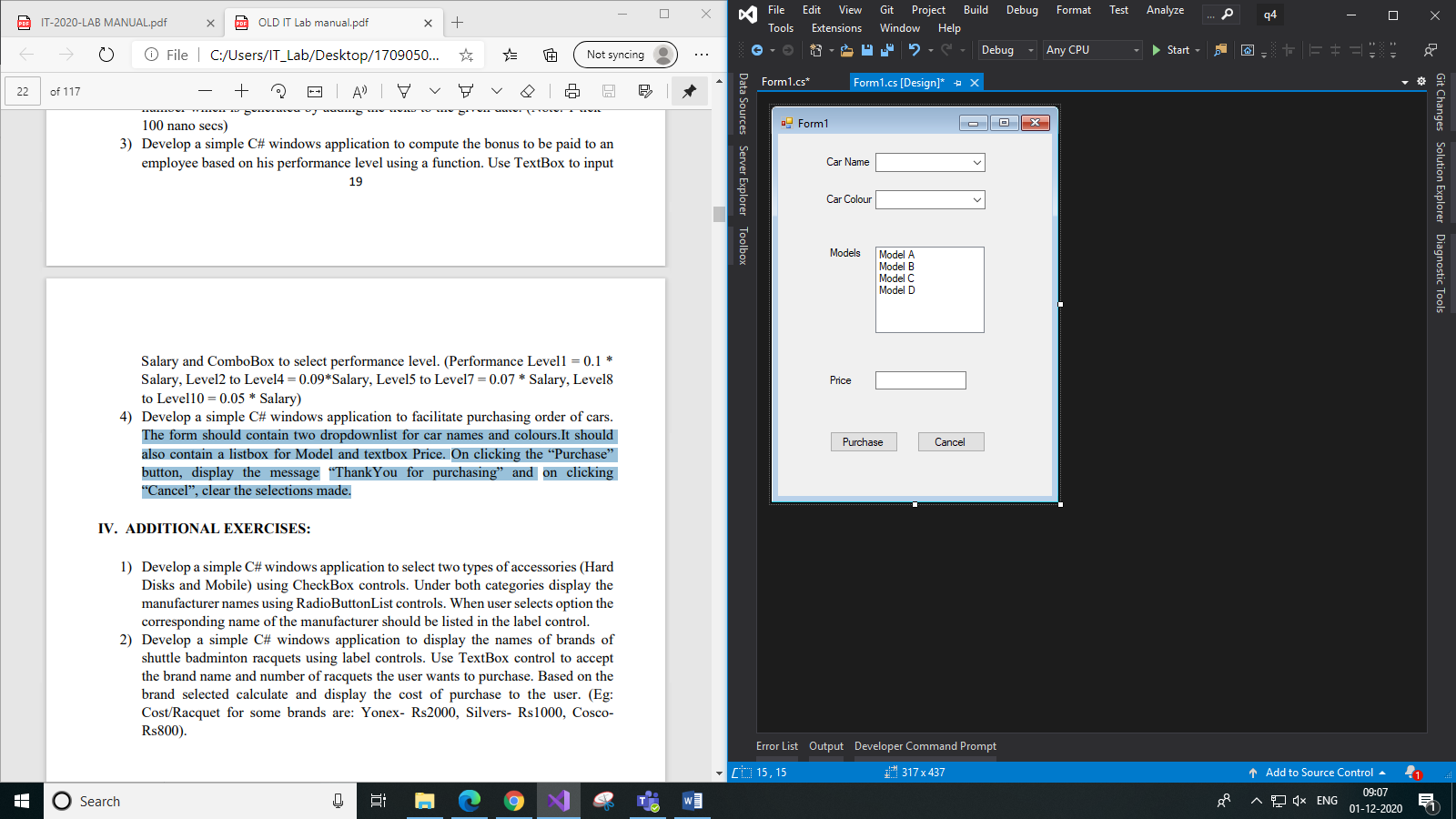
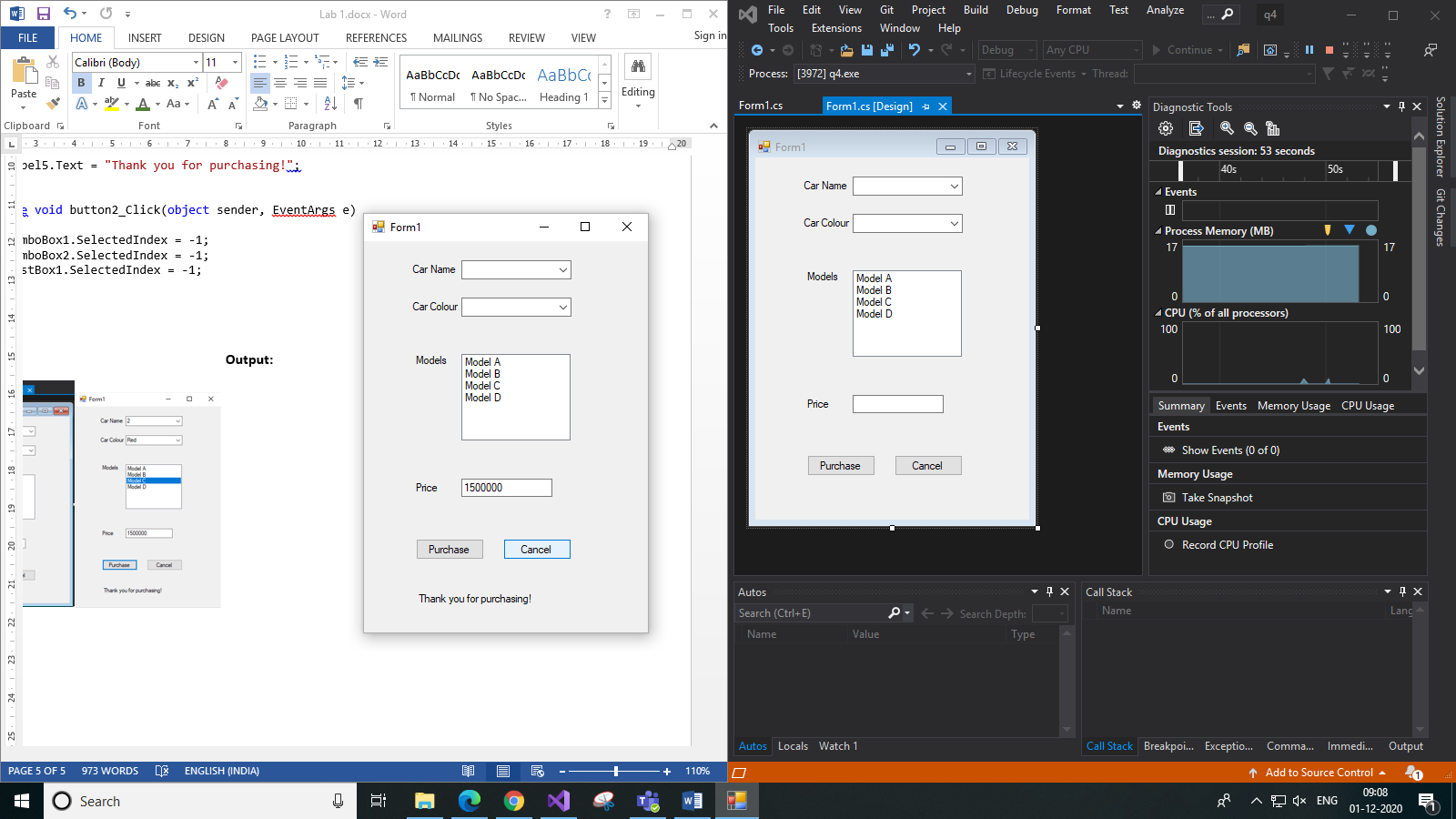
listBox1.SelectedIndex = -1;

}

}

}

**Designer View: Output:**



1. *On Clicking “Purchase” button b) On clicking “Cancel” button*

*(Designer view included with code as it is a windows application and does not have code for the design)*