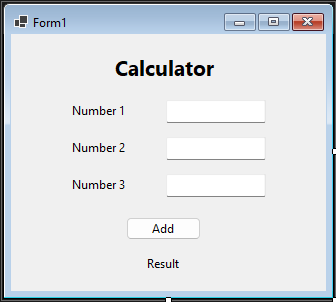
# Practical Guide 06 – Part I

## Question 01

**I.**

****

**II.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_06\_01\_01

{

public class Calculator

{

public int Add(int a,int b)

{

return a + b;

}

public int Add(int a,int b, int c)

{

return a + b + c;

}

public double Add(double a,double b)

{

return a + b;

}

}

}

**III.**

namespace Lab\_06\_01\_01

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

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

{

Calculator calc = new Calculator();

int num1, num2, num3;

bool isNum3Empty = string.IsNullOrEmpty(txtNum3.Text);

if(int.TryParse(txtNum1.Text, out num1) && int.TryParse(txtNum2.Text, out num2)) {

if (isNum3Empty)

{

int result = calc.Add(num1, num2);

lblResult.Text = $"Sum of two numbers: {result}";

}

else if (int.TryParse(txtNum3.Text, out num3))

{

int result = calc.Add(num1, num2, num3);

lblResult.Text = $"Sum of three numbers: {result}";

}

else

{

lblResult.Text = "invalid input in third number";

}

}

else

{

lblResult.Text = "Invalid input in first two numbers";

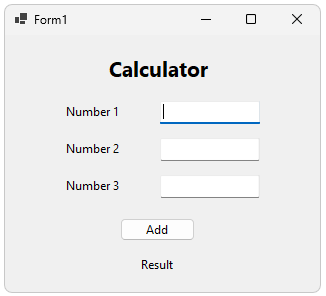
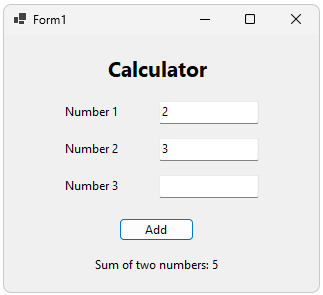
}

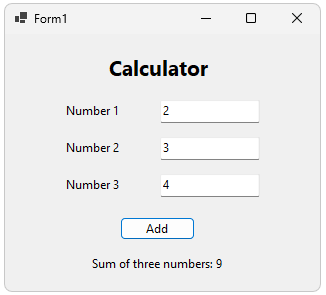
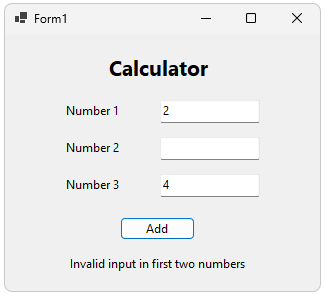
}

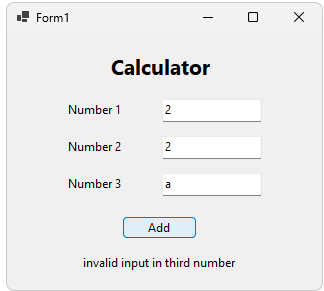
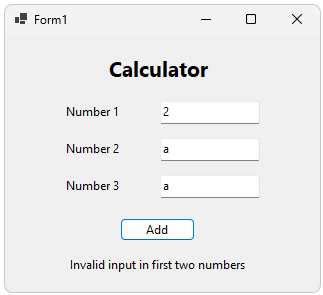
}

}

**IV.**

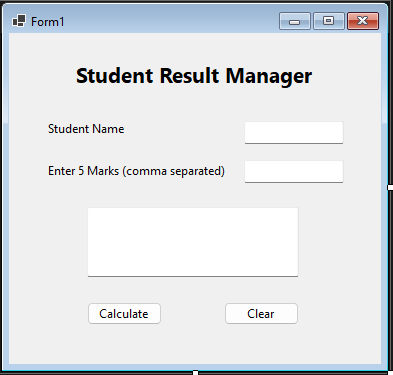
**** ****

**** ****

**** ****

## Question 02

**I.**

****

**II.**

using System;

using System.Collections.Generic;

using System.Diagnostics.Eventing.Reader;

using System.Linq;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_06\_01\_02

{

internal class Student

{

private string name;

private int[] marks;

public Student(string name, int[] marks)

{

this.name = name;

this.marks = marks;

}

public double CalculateAverage()

{

int total = 0;

foreach (int mark in marks)

{

total += mark;

}

return (double)total / marks.Length;

}

public string GetGrade()

{

double avg = CalculateAverage();

if (avg >= 75) return "Distinction";

else if (avg >= 60) return "credit";

else if (avg >= 50) return "Pass";

else return "Fail";

}

public string GetGrade(double avg)

{

if (avg >= 75) return "Distinction (External)";

else if (avg >= 60) return "credit (External)";

else if (avg >= 50) return "Pass (External)";

else return "Fail (External)";

}

public int FindMaxRecursive(int[] arr, int n)

{

if (n == 1) return arr[0];

return Math.Max(arr[n - 1], FindMaxRecursive(arr, n - 1));

}

public string GetSummery()

{

double avg = CalculateAverage();

int maxMark = FindMaxRecursive(marks, marks.Length);

string grade = GetGrade(avg);

return $"Name: {name}\r\nAverage: {avg:F2}\r\nMax Mark: {maxMark}\r\nGrade: {grade}";

}

}

}

**III.**

namespace Lab\_06\_01\_02

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

txtName.Clear();

txtMarks.Clear();

txtResult.Clear();

}

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

{

try

{

string studentName = txtName.Text;

string[] parts = txtMarks.Text.Split(',');

int[] marks = Array.ConvertAll(parts, int.Parse);

if (marks.Length != 5)

{

MessageBox.Show("Please enter exactly 5 marks separated by commas.");

return;

}

Student s = new Student(studentName, marks);

txtResult.Text = s.GetSummery();

}

catch

{

MessageBox.Show("Error: + ex.Message");

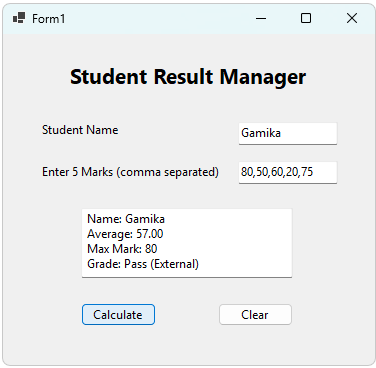
}

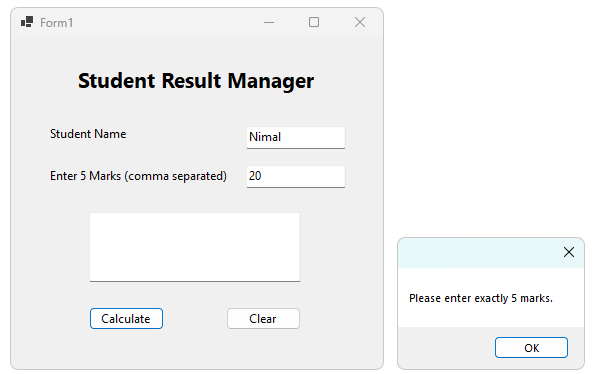
}

}

}

**V.**

****

****