DOTNET

**1)Write a program to divide the number and raised the build in exception**

using System; public class Program

{

static void Main()

{

// take first int input from user

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

int num1 = int.Parse(Console.ReadLine());

// take second int input from user

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

int num2 = int.Parse(Console.ReadLine()); try

{

// code that may raise raise an exception

int result = num1 / num2;

Console.WriteLine("Division of two numbers is: " + result);

}

// this catch block gets executed only when an exception is raised catch (DivideByZeroException)

{

Console.WriteLine("An exception occurred:");

} finally {

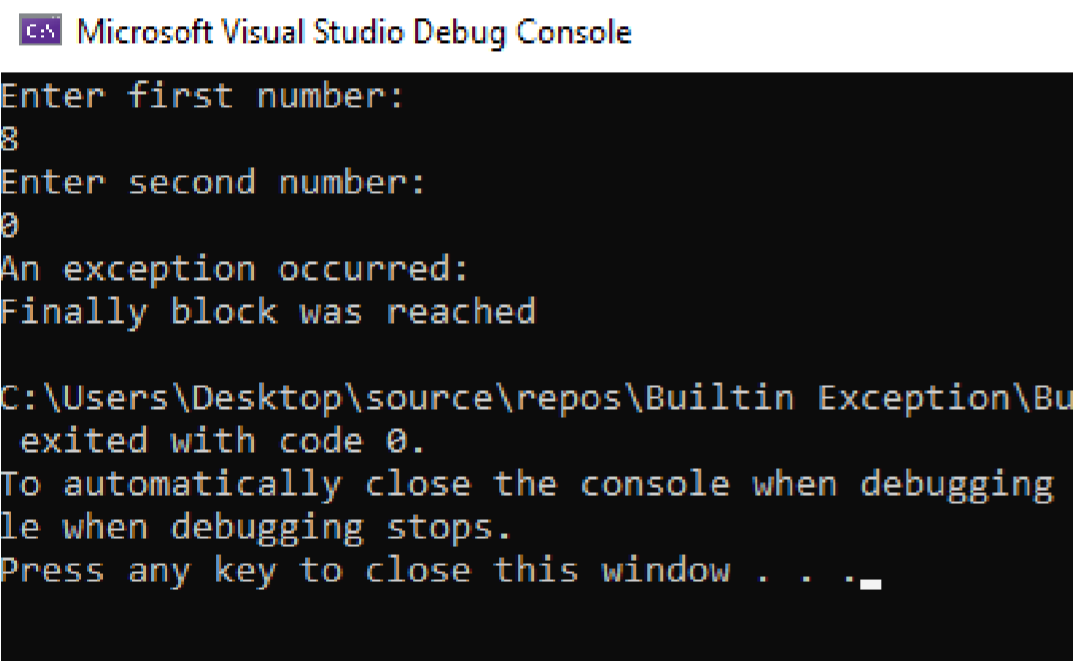
// this code is always executed whether of exception occurred or not

Console.WriteLine("Finally block was reached");

}

}

}



**2)Write a program to work with StringBuilder Create a string Assign it with large string value consisting of no of words.Access the string character by character and print**

using System.Text; namespace stringbuilder2

{

class Program

{

static void Main(string[] args)

{

StringBuilder s1 = new StringBuilder("My College", 90);

//access character by character

for (int i = 0; i < s1.Length; i++)

{

Console.WriteLine(s1[i]);

}

Console.WriteLine();

Console.WriteLine("Original String is:" + s1); s1.Replace("College", "Pizza");

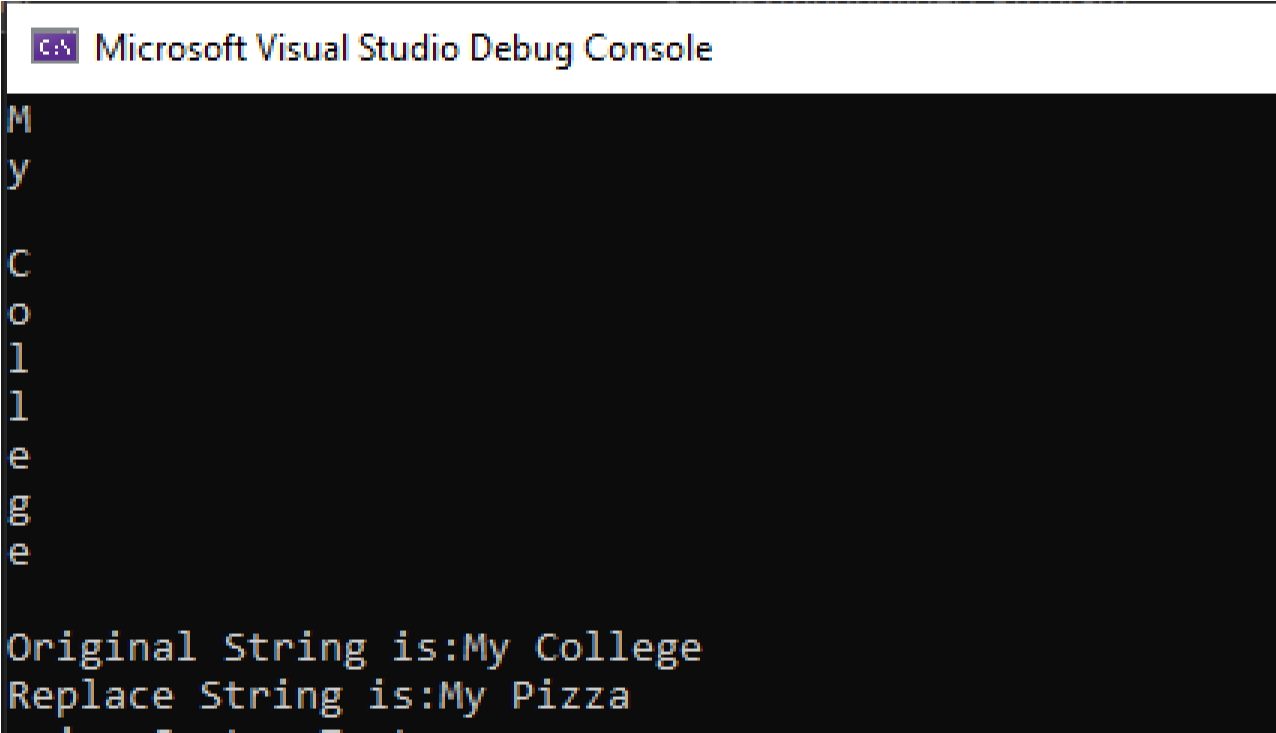
Console.WriteLine("Replace String is:" + s1);

Console.Read();

}

}

}



**3)Write a program to implement Custom Exception. Create InvalidStudentNameException class in a school application, which does not allow any special character or numeric value in a name of any of the students. Use Regex("^[a-zA-Z]+$") to check Student Name.**

The get method returns the value of the variable name. The set method assigns a value to the name variable using System; using System.Text;

using System.Text.RegularExpressions; namespace customer

{

class Program

{

class Student

{

public string StudentName {

get; set;

}

}

class InvalidStudentNameException : Exception

{

public InvalidStudentNameException(string name): base(String.Format("Invalid Student Name: {0}", name))

{

}

}

static void Main(string[] args)

{

Student newStudent = null; try

{

newStudent = new Student();

Console.WriteLine("Enter your name"); string a = Console.ReadLine();

newStudent.StudentName = a;

ValidateStudent(newStudent);

}

catch (InvalidStudentNameException ex)

{

Console.WriteLine(ex.Message);

}

}

private static void ValidateStudent(Student std)

{

Regex regex = new Regex("^[a-zA-Z]+$");

if (!regex.IsMatch(std.StudentName))

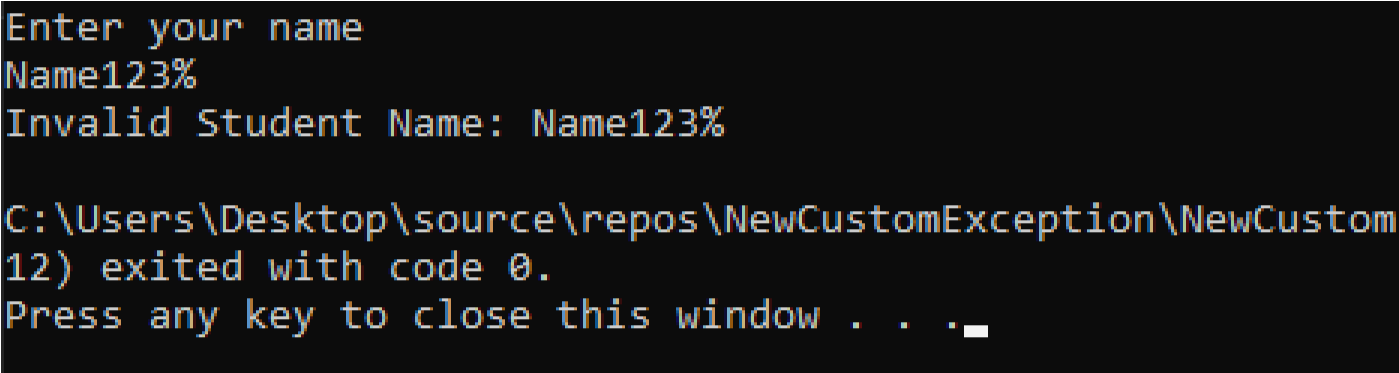
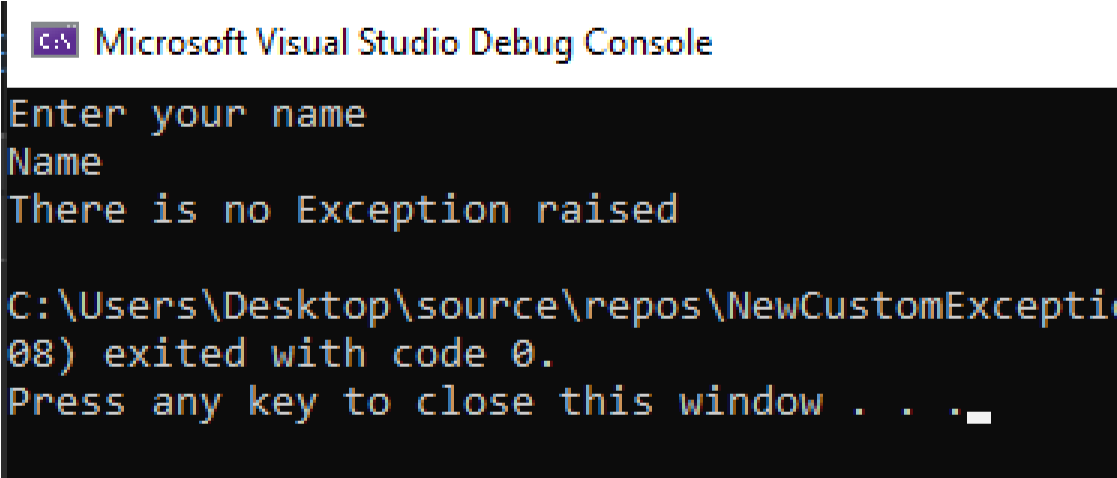
throw new InvalidStudentNameException(std.StudentName); else

Console.WriteLine("There is no Exception raised");

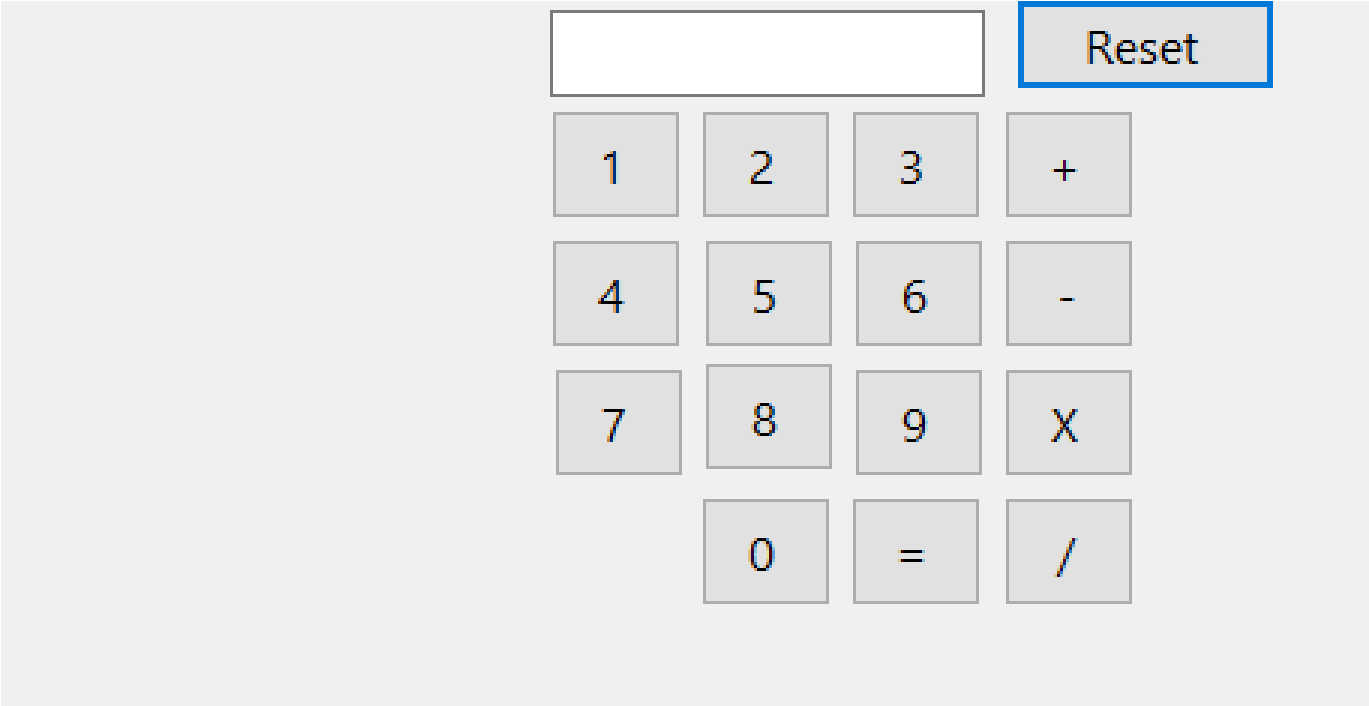
}

}

}



**4)Write a form based program offering binary calculator having following functionality Add, multiply, subtract, divide**



Code

Public Class Form1

Dim a As Double

Dim operation As Double

Private Sub TextBox1\_TextChanged(sender As Object, e As EventArgs) Handles

TextBox1.TextChanged

End Sub

Private Sub Reset\_Click(sender As Object, e As EventArgs) Handles Reset.Click

TextBox1.Text = ""

End Sub

Private Sub B1\_Click(sender As Object, e As EventArgs) Handles B1.Click

TextBox1.Text = TextBox1.Text + "1"

End Sub

Private Sub B2\_Click(sender As Object, e As EventArgs) Handles B2.Click

TextBox1.Text = TextBox1.Text + "2"

End Sub

Private Sub B3\_Click(sender As Object, e As EventArgs) Handles B3.Click

TextBox1.Text = TextBox1.Text + "3"

End Sub

Private Sub B4\_Click(sender As Object, e As EventArgs) Handles B4.Click

TextBox1.Text = TextBox1.Text + "4"

End Sub

Private Sub B5\_Click(sender As Object, e As EventArgs) Handles B5.Click

TextBox1.Text = TextBox1.Text + "5"

End Sub

Private Sub B6\_Click(sender As Object, e As EventArgs) Handles B6.Click TextBox1.Text = TextBox1.Text + "6"

End Sub

Private Sub B7\_Click(sender As Object, e As EventArgs) Handles B7.Click

TextBox1.Text = TextBox1.Text + "7"

End Sub

Private Sub B8\_Click(sender As Object, e As EventArgs) Handles B8.Click

TextBox1.Text = TextBox1.Text + "8"

End Sub

Private Sub B9\_Click(sender As Object, e As EventArgs) Handles B9.Click

TextBox1.Text = TextBox1.Text + "9"

End Sub

Private Sub B0\_Click(sender As Object, e As EventArgs) Handles B0.Click

TextBox1.Text = TextBox1.Text + "0"

End Sub

Private Sub Button17\_Click(sender As Object, e As EventArgs) Handles Multiply.Click a = Val(TextBox1.Text) operation = 3 TextBox1.Text = ""

End Sub

Private Sub Button15\_Click(sender As Object, e As EventArgs) Handles Plus.Click a = Val(TextBox1.Text) operation = 1 TextBox1.Text = ""

End Sub

Private Sub Button14\_Click(sender As Object, e As EventArgs) Handles Equal.Click

If operation = 1 Then

TextBox1.Text = a + Val(TextBox1.Text)

End If

If operation = 2 Then

TextBox1.Text = a - Val(TextBox1.Text)

End If

If operation = 3 Then

TextBox1.Text = a \* Val(TextBox1.Text)

End If

If operation = 4 Then

TextBox1.Text = a / Val(TextBox1.Text)

End If

End Sub

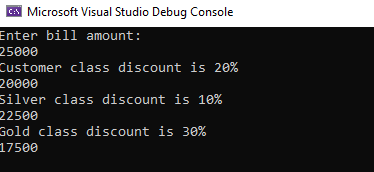
Private Sub Button16\_Click(sender As Object, e As EventArgs) Handles Minus.Click a = Val(TextBox1.Text) operation = 2 TextBox1.Text = ""

End Sub

Private Sub Button18\_Click(sender As Object, e As EventArgs) Handles Division.Click a = Val(TextBox1.Text) operation = 4 TextBox1.Text = ""

End Sub

End Class



5) Design a form which offer user registration form on click of OK, register user data should get saved in xml.

6) Design a form which offer user student name, class, and age. Form on click of OK, register student data should get saved in xml.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WindowsFormsApp1

{

public class Studentclass

{

public string Name { get; set; }

public string Password{ get; set; }

}

}

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;

using System.Xml;

using System.Xml.Serialization;

using System.IO;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

XmlSerializer xs;

List<Studentclass> ls;

public Form1()

{

InitializeComponent();

ls = new List<Studentclass>();

xs = new

XmlSerializer(typeof(List<Studentclass>));

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

FileStream fs = new FileStream("users.xml", FileMode.Create);

Studentclass sc = new Studentclass();

sc.Name = textBox1.Text;

sc.Password = textBox2.Text;

ls.Add(sc);

xs.Serialize(fs,ls);

fs.Close();

}

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

{

FileStream fs = new FileStream("users.xml", FileMode.Open);

ls=(List<Studentclass>)xs.Deserialize(fs);

dataGridView1.DataSource = ls;

fs.Close();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

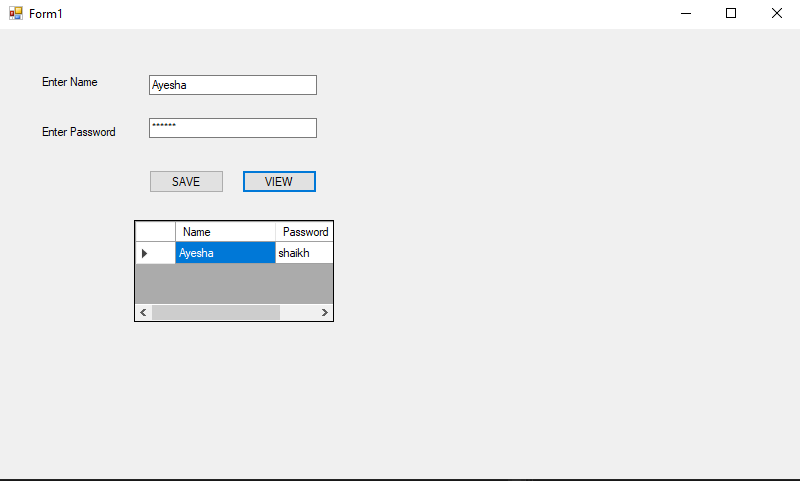
{

}

}

}

**Output:**

****

**7) Create base class Customer and subclasses SilverCustomer and GoldCustomer. Define discount() method in Customer class which returns 20% discount Overload discount method in the subclasses and return different discount value Define base class variable as "Customer cust" Assign different objects of Customer, SilverCustomer and GoldCustomer to variable cust one after other and invoke discount method each time. What is the discount % returned each time?**

using System;

class customer {

public double discount(float b)

{

return b-(b\*0.2);

}

}

class silver: customer

{

public double discount(float b)

{

return b-(b\*0.1);

} }

class gold : customer

{

public double discount(float b)

{

return b-(b\*0.3);

} }

class MainClass { public static void Main (string[] args) {

Console.WriteLine("Enter bill amount:"); float bill = Single.Parse(Console.ReadLine()); customer c1=new customer();

Console.WriteLine("Customer class discount is 20%");

Console.WriteLine(c1.discount(bill));

silver s1=new silver();

Console.WriteLine("Silver class discount is 10%");

Console.WriteLine(s1.discount(bill));

gold g1=new gold();

Console.WriteLine("Gold class discount is 30%");

Console.WriteLine(g1.discount(bill));

}

}

8)magicsquare

Public Class Form1

Dim intmagic(3, 3) As Integer

Dim G As Graphics

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

TextBox1.Text = intmagic(0, 0) + intmagic(0, 1) + intmagic(0, 2) + intmagic(0, 3)

G.Clear((Color.FromName("CControl")))

For x = 0 To 3

For y = 0 To 3

G.DrawRectangle(Pens.Blue, New Rectangle(x \* 64, y \* 64, 64, 64))

G.DrawString(intmagic(x, y), Me.Font, Brushes.Yellow, New Point(x \* 64, y \* 64))

Next

Next

End Sub

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

G = Me.CreateGraphics

intmagic(0, 0) = 16

intmagic(0, 1) = 5

intmagic(0, 2) = 9

intmagic(0, 3) = 4

intmagic(1, 0) = 3

intmagic(1, 1) = 10

intmagic(1, 2) = 6

intmagic(1, 3) = 15

intmagic(2, 0) = 2

intmagic(2, 1) = 11

intmagic(2, 2) = 7

intmagic(2, 3) = 14

intmagic(3, 0) = 13

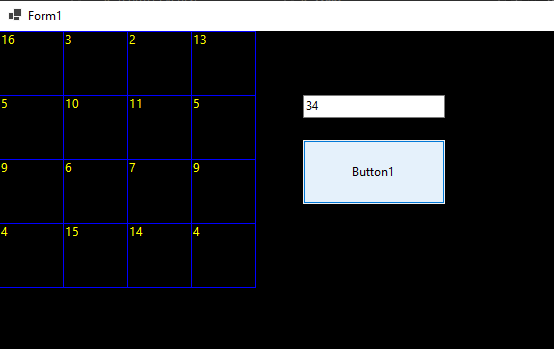
intmagic(3, 1) = 5

intmagic(3, 2) = 9

intmagic(3, 3) = 4

End Sub

End Class



9)

Reflection Example

using System;

using System.Reflection;

namespace Demo

{

    public class MainClass

    {

    private static void Main()

    {

        //Get the type using GetType() static method

        Type T = Type.GetType("Demo.Customer");

        // print the type details

        Console.WriteLine("Full Name ={0}", T.FullName);

        Console.WriteLine("Just The Class Name={0}", T.Name);

        Console.WriteLine("Just the Namespace={0}", T.Namespace);

        Console.WriteLine();

            // PRINT THE LIST OF THE METHOD

            Console.WriteLine("Method in Customer Class");

            MethodInfo[] methods = T.GetMethods();

            foreach (MethodInfo method in methods)

            {

                Console.WriteLine(method.ReturnType.Name + "" + method.Name);

            }

        Console.WriteLine();

        Console.Read();

    }

    }

public class Customer

{

    public int Id { get; set; }

    public string Name { get; set; }

    public Customer(int id , string Name)

    {

        this.Id = Id;

        this.Name = Name;

    }

    public Customer()

    {

        this.Id = -1;

        this.Name= String.Empty;

    }

    public void PrintId()

    {

        Console.WriteLine("ID ={0}", this.Id);

    }

    public void PrintName()

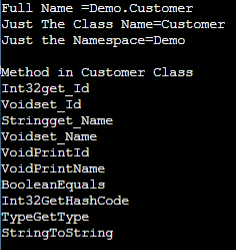
    {

        Console.WriteLine("Name={0}", this.Name);

    }

}

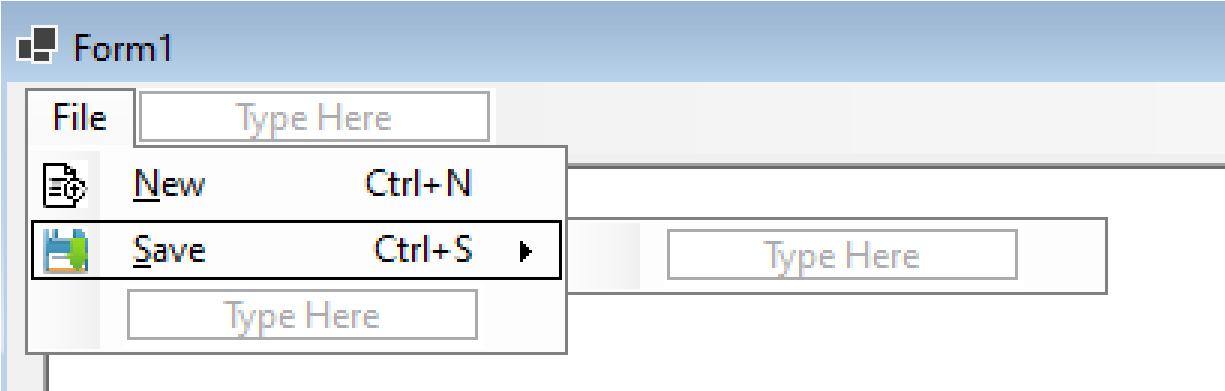
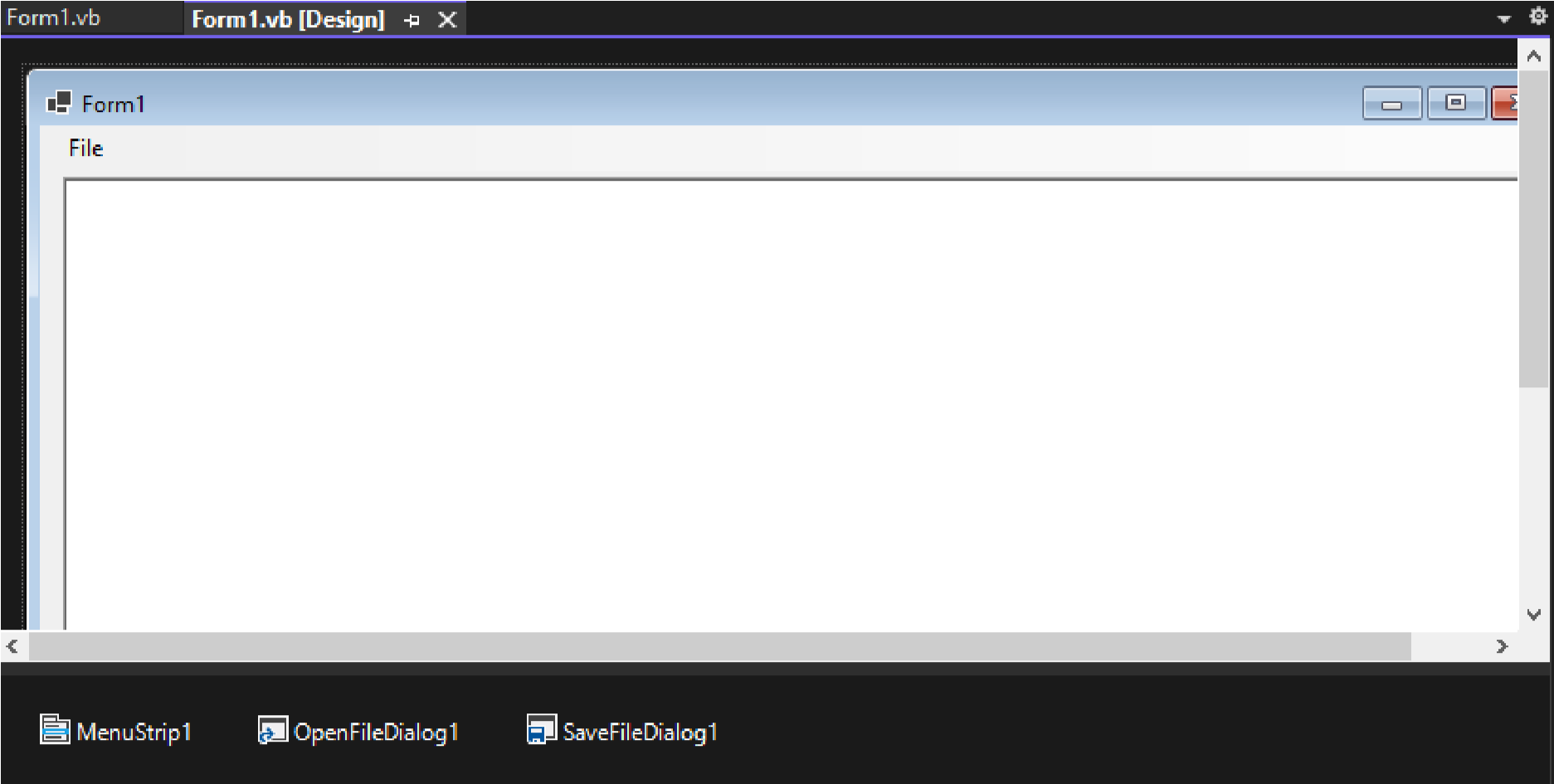
}

.

10) **Implement a Simple Editor which has following featur Menu : File, New, Save**

·

Public Class Form1



Private Sub FileToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles FileToolStripMenuItem.Click

End Sub

Private Sub NewToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles NewToolStripMenuItem.Click

RichTextBox1.Text = ""

End Sub

Private Sub SaveToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

SaveToolStripMenuItem.Click

If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, RichTextBox1.Text, False)

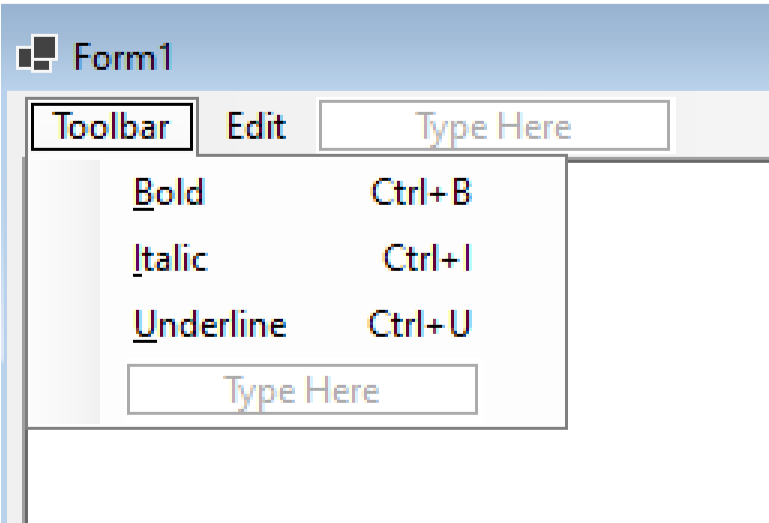
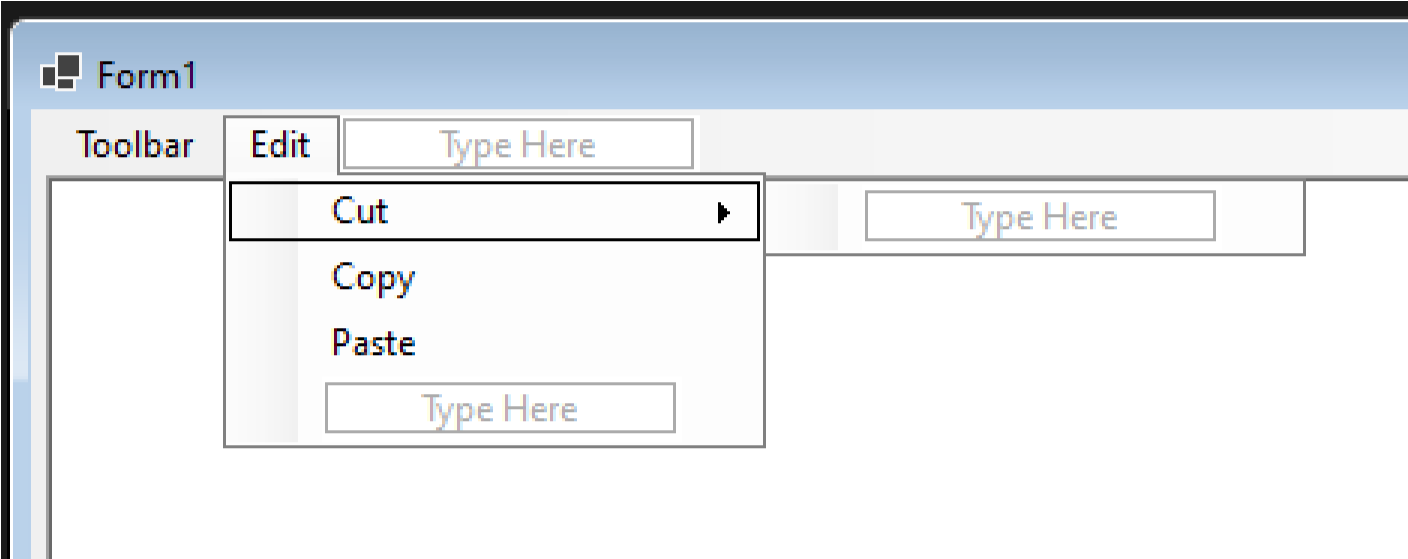
End If

MsgBox("File Saved")

End Sub

End Class

**8)Bold Italic Underline**



Public Class Form1

Private Sub BoldToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles BoldToolStripMenuItem.Click

RichTextBox1.SelectionFont = New Font("Times New Roman", 16, FontStyle.Bold)

End Sub

Private Sub ItalicToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ItalicToolStripMenuItem.Click

RichTextBox1.SelectionFont = New Font("Times New Roman", 16, FontStyle.Italic)

End Sub

Private Sub UnderlineToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles UnderlineToolStripMenuItem.Click

RichTextBox1.SelectionFont = New Font("Times New Roman", 16, FontStyle.Underline)

End Sub

Private Sub CutToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

CutToolStripMenuItem.Click

RichTextBox1.Cut()

End Sub

Private Sub CopyToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

CopyToolStripMenuItem.Click

RichTextBox1.Copy()

End Sub

Private Sub PasteToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

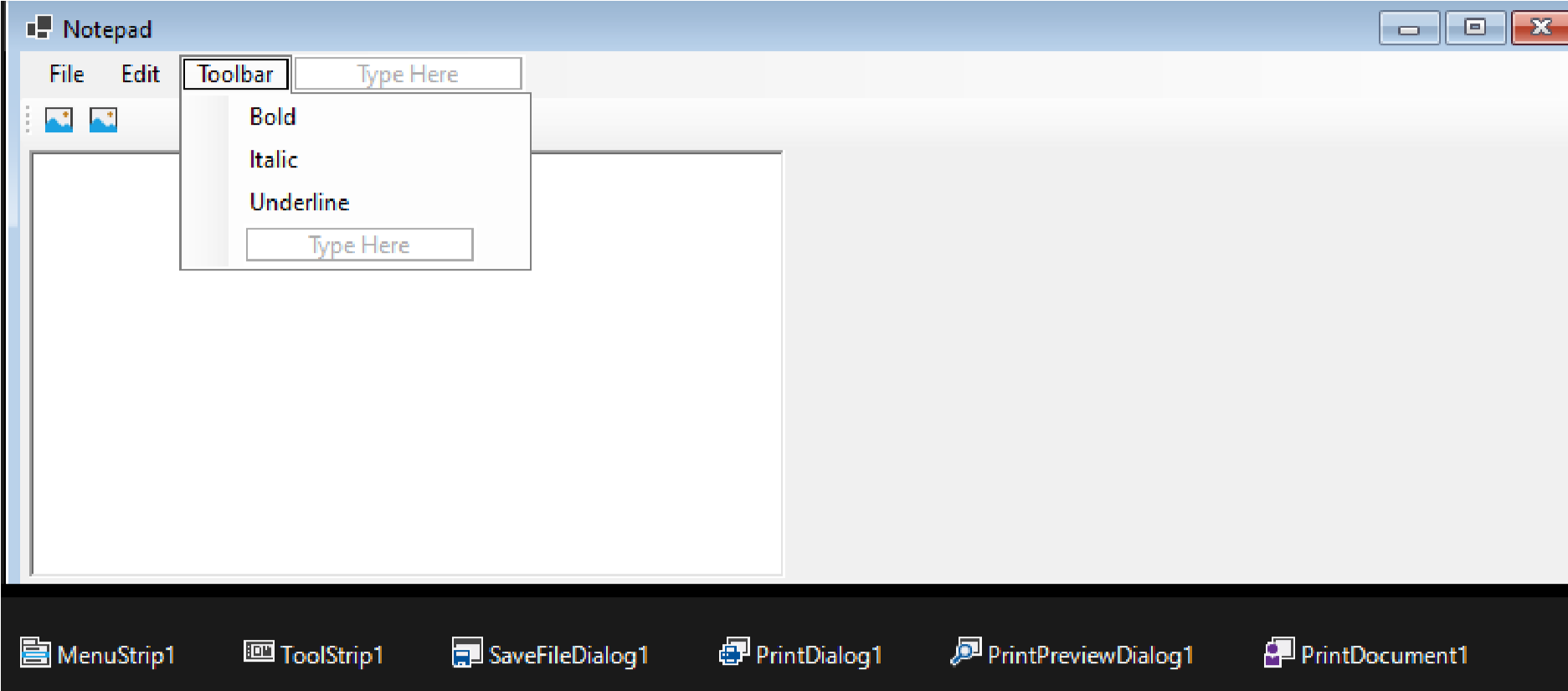
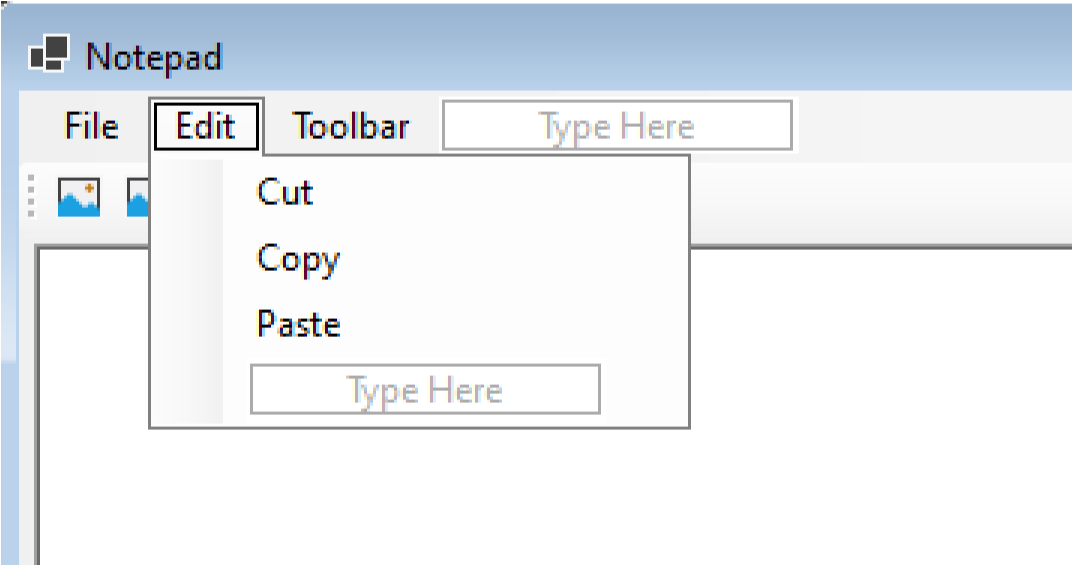
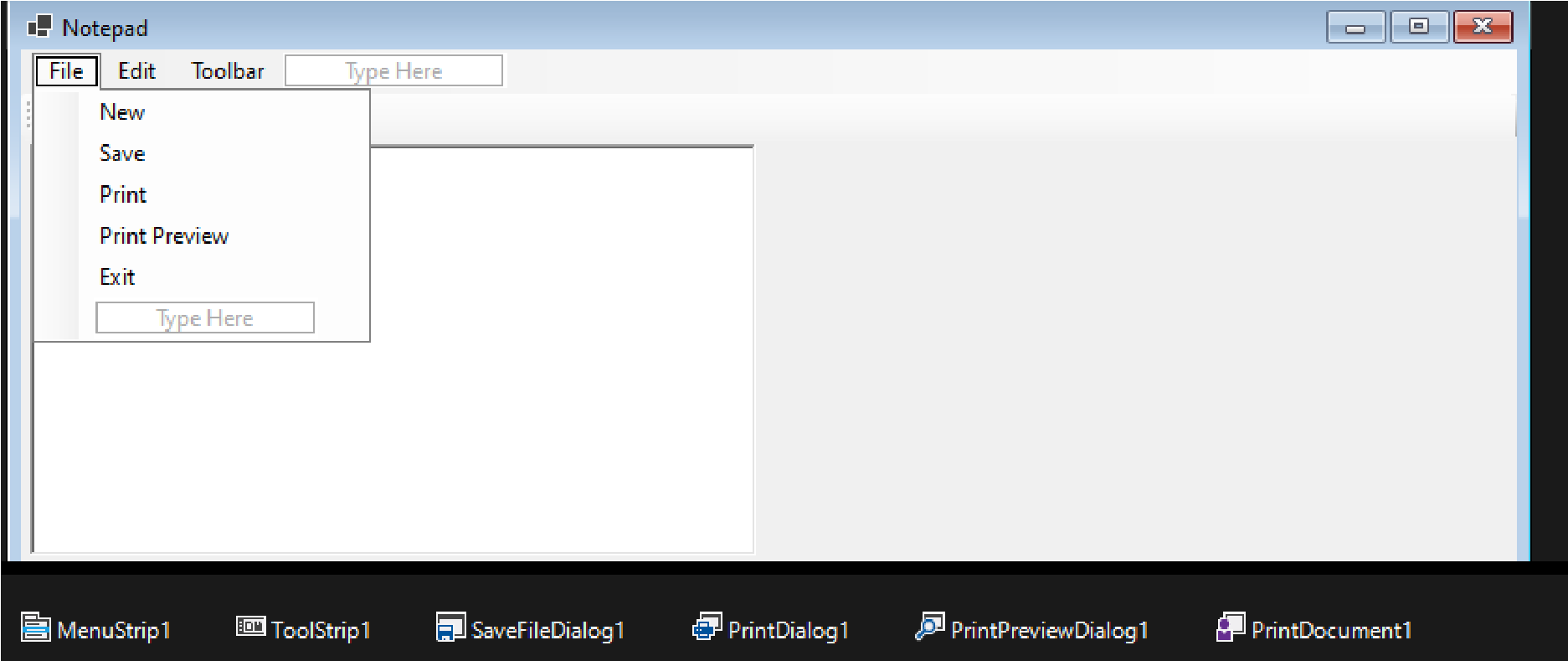
PasteToolStripMenuItem.Click

RichTextBox1.Paste()

End Sub

End Class

Public Class Form1



Private Sub NewToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

NewToolStripMenuItem.Click

RichTextBox1.Text = ""

End Sub

Private Sub SaveToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles SaveToolStripMenuItem.Click

If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, RichTextBox1.Text, False)

End If

MsgBox("File Saved")

End Sub

Private Sub PrintToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles PrintToolStripMenuItem.Click

PrintDocument1.Print()

End Sub

Private Sub PrintPreviewToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

PrintPreviewToolStripMenuItem.Click

PrintDialog1.ShowDialog()

End Sub

Private Sub ExitToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

ExitToolStripMenuItem.Click

Me.Close()

End Sub

Private Sub CutToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

CutToolStripMenuItem.Click

RichTextBox1.Cut()

End Sub

Private Sub CopyToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

CopyToolStripMenuItem.Click

RichTextBox1.Copy()

End Sub

Private Sub PasteToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles

PasteToolStripMenuItem.Click

RichTextBox1.Paste()

End Sub

Private Sub BoldToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles BoldToolStripMenuItem.Click

RichTextBox1.SelectionFont = New Font("Times New Roman", 16, FontStyle.Bold)

End Sub

Private Sub ItalicToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ItalicToolStripMenuItem.Click

RichTextBox1.SelectionFont = New Font("Times New Roman", 16, FontStyle.Italic)

End Sub

Private Sub UnderlineToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles UnderlineToolStripMenuItem.Click

RichTextBox1.SelectionFont = New Font("Times New Roman", 16, FontStyle.Underline)

End Sub

Private Sub ToolStripButton1\_Click(sender As Object, e As EventArgs) Handles ToolStripButton1.Click

If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, RichTextBox1.Text, False)

End If

MsgBox("File Saved")

End Sub

Private Sub PrintDocument1\_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim font1 As New Font("arial", 16, FontStyle.Regular)

e.Graphics.DrawString(RichTextBox1.Text, font1, Brushes.Black, 100, 100) End Sub

End Class