Introduction to Selenium IDE

Selenium is an open-source and a portable automated software testing tool for testing web applications. It has capabilities to operate across different browsers and operating systems. Selenium is not just a single tool but a set of tools that helps testers to automate web-based application smoreefficiently.

Selenium supports the languages like HTML, Java, PHP, Perl, Python, and Ruby and C #. It supports the browsers like IE, Mozilla Firefox, Safari, Google Chrome and Opera. It supports the operating systems like Windows , Linuxand Mac.

Selenium IDE

The Selenium-IDE (Integrated Development Environment) is an easy-to-use Firefox plug-in to develop Selenium test cases. It provides a Graphical User Interface for recording user actions using Firefox which is used to learn and use Selenium, but it can only be used with Firefox browsers other browsers are not supported.

Selenium Commands–"Selenese"

Seleniumcommands,oftencalledselenese,arethesetofcommandsthatrunyourtests. A sequence of these commands is a test script. Some of the commands used are:

a) open

b) open window

c) click

d) click and wait

e) clickat

f) store

g) echo

White Box Testing Techniques:

 Statement Coverage -This technique is aimed at exercising all programming statements with minimal tests.

 Branch Coverage - This technique is running a series of tests to ensure that all branches are tested at least once.

 Path Coverage -This technique corresponds to testing all possible paths which means that each statement and branch is covered.

 Function Coverage: Testing to identify how many program functions are covered by test cases.

Unit testing

It is a testing methodology where individual units are tested in isolation from other units. This is usually done by developers. A unit can be considered as a class or methods inside a class which needs to be tested individually.

A unit test is a piece of code written by a developer that executes a specific functionality in the code to be tested and asserts a certain behaviour or state.

Type of unit testing

There are two way stopper form unit testing:

1) Manual testing

2) Automated testing.

1) Manual Testing If you execute the test cases manually without any tool support, it is known as manual testing. It is time consuming and less reliable.

2) Automated Testing If you execute the test cases by tool support, it is known as automated testing. It is fast and more reliable.

TestNG

TestNG is a testing framework that overcomes the limitations of popular testing framework called JUnit. The "NG" means "Next Generation". TestNG is similar to JUnit (especiallyJUnit4),but it is not a JUnit extension.

1.Take ATM system and study its system specifications and report the various bugs.

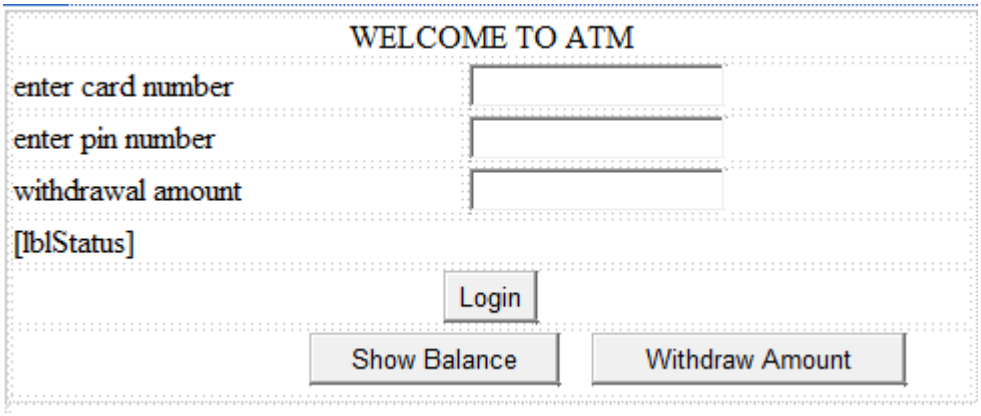
a. Machine is accepting ATM card.

b.Machine is rejecting expired card.

c.Successfull entry of pin number.

d.Unsuccessfull operation due to enter wrong PIN number 3 times.

Design:



Database connection:

DESKTOP-AC4AK63.... dbo.atm_entries

	atm_num	atm_pin	balance
▶	ca161001	1234	1500
	ca161002	5678	5200
*	NULL	NULL	NULL

WebForm.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
namespace WebApplication37
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      String cardNo;
      String pin;
      String withAmount;
      String expireDate;
      SqlConnection connection;
      Boolean isLogin = false;

      protected void Page_Load(object sender, EventArgs e)
      {

      }
protected void Page_Init(object sender, EventArgs e)
 connection = new SqlConnection("Data Source=localhost; Initial Catalog=test; user id = sa;
 password = Password@123;");
          connection.Open();
          }
 protected void btnLogin_Click(object sender, EventArgs e)
      {
          cardNo = txtCardNum.Text;
          pin = txtPinNum.Text;
          withAmount = txtWithdrawAmount.Text;
```

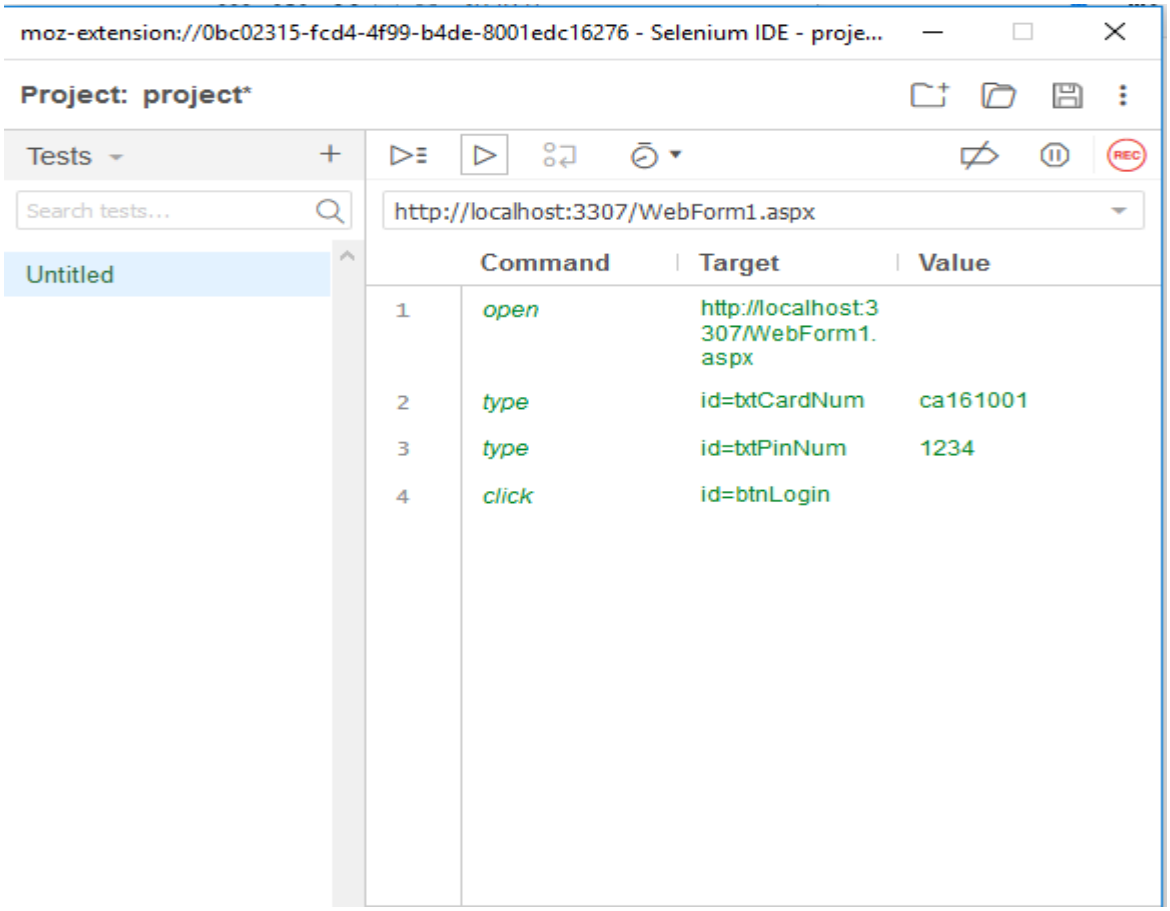
```
SqlCommand command = new SqlCommand("select * from atm_entries where atm_num = @param1 AND atm_pin = @param2", connection);
        command.Parameters.AddWithValue("@param1", cardNo);
        command.Parameters.AddWithValue("@param2", pin);
        SqlDataReader reader = command.ExecuteReader();
         if (reader.Read())
        {
        if (reader[1].ToString().Equals(pin))
           {
              isLogin = true;
              lblStatus.Text = "Login success";
              btnShowBalance.Visible = true;
              btnWithdraw.Visible = true;
              txtWithdrawAmount.Visible = true;
              withdrawLabel.Visible = true;
              btnLogin.Visible = false;
           }
           else
           {
              isLogin = false;
              lblStatus.Text = "Login failed";
           }
        }
        else
        {
           lblStatus.Text = "Atm card not present or wrong password";
        }

        reader.Close();
        }
        protected void btnWithdraw_Click(object sender, EventArgs e)
      {
       cardNo = txtCardNum.Text;
```

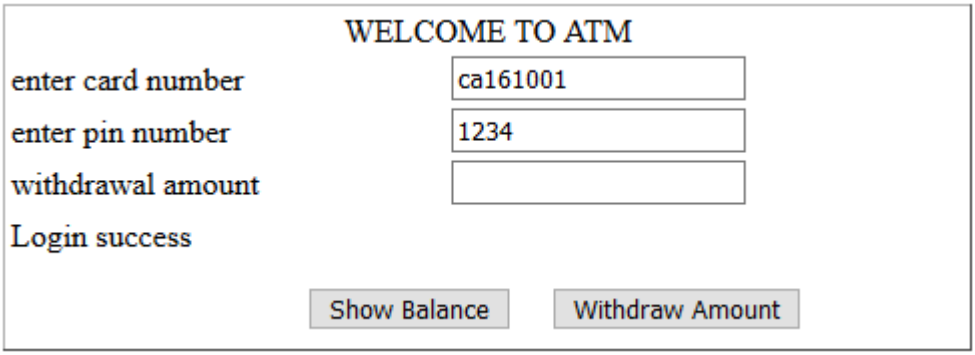
```
pin = txtPinNum.Text;
withAmount = txtWithdrawAmount.Text;
int balance;
SqlCommand checkExists = new SqlCommand("select balance from atm_entries
where atm_num = @param1", connection);
      checkExists.Parameters.AddWithValue("@param1", cardNo);
       SqlDataReader checkReader = checkExists.ExecuteReader();
       if (checkReader.Read())
       {
          balance = (int)checkReader[0];
          int withdrawal = Int32.Parse(withAmount);

          lblStatus.Text = "Balance: " + balance + " Withdrawal balance: " +withdrawal;

          if (balance < withdrawal)
          {
             lblStatus.Text = "Insufficient Balance";
          }
          else
          {
             int remainingAmount = balance - withdrawal;
             SqlCommand updateCommand = connection.CreateCommand();
 updateCommand.CommandText = "UPDATE atm_entries SET balance =
@param1 WHERE atm_num = @param2 ";
 updateCommand.Parameters.AddWithValue("@param1",  remainingAmount);
 updateCommand.Parameters.AddWithValue("@param2", cardNo);
 checkReader.Close();
  var id = updateCommand.ExecuteScalar();
 lblStatus.Text = "Transcation Successfull: Remaining Balance = " + remainingAmount;
       }
      }
      checkReader.Close();
      }
```

```
protected void btnShowBalance_Click(object sender, EventArgs e)
    {
        cardNo = txtCardNum.Text;
        SqlCommand command = new SqlCommand("select balance from atm_entries where
        atm_num = @param1", connection);
        command.Parameters.AddWithValue("@param1", cardNo);
        SqlDataReader checkReader = command.ExecuteReader();
        if (checkReader.Read())
        {
            lblStatus.Text = "Your balance: " + checkReader[0].ToString();
        }
        else
        {
            lblStatus.Text = "Error try again";
        }
    }
  }
}
```

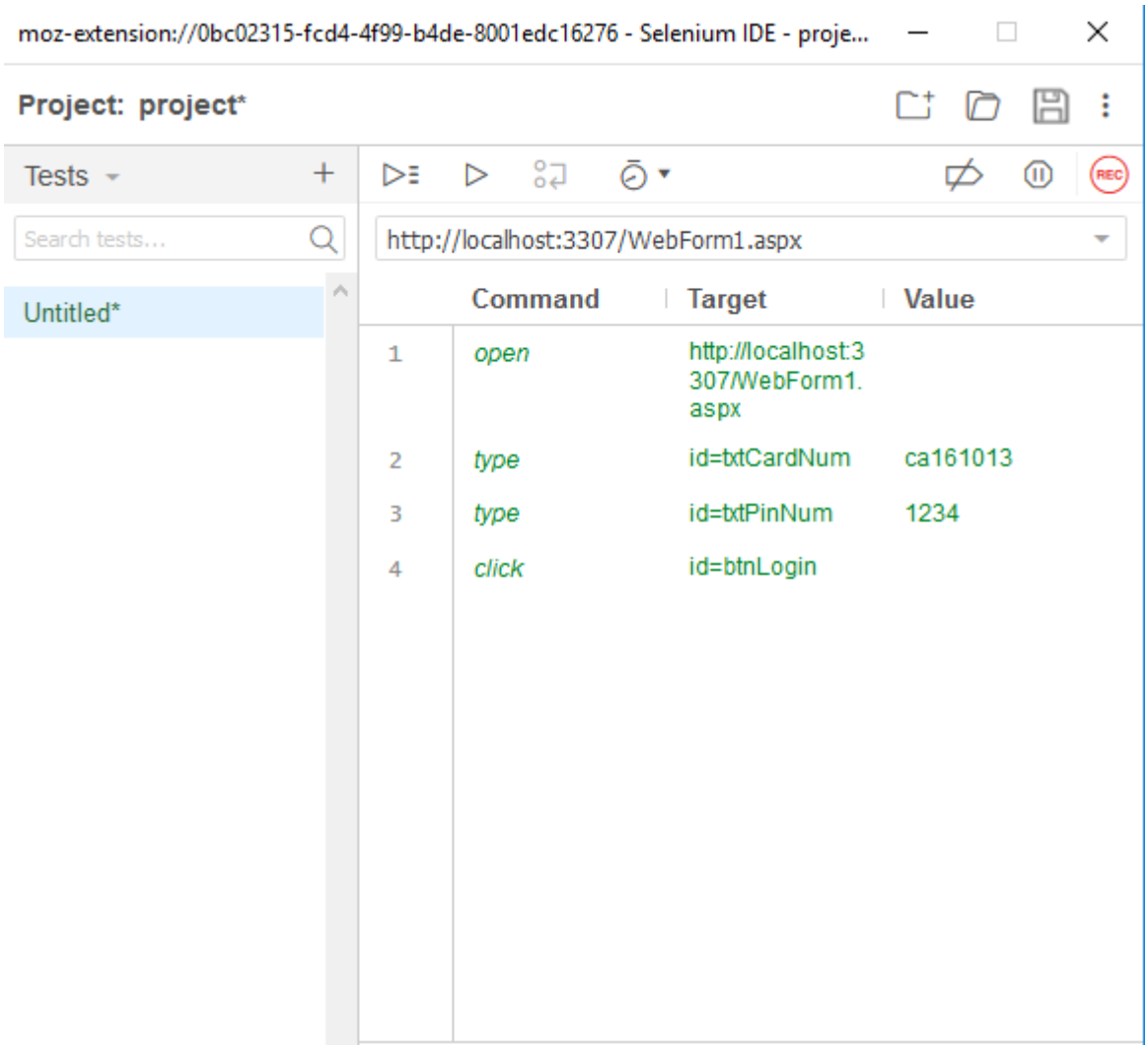
OUTPUT:

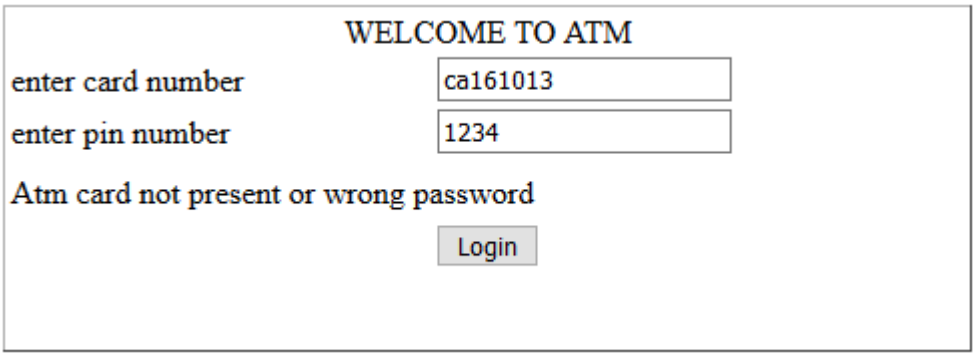
a.Machine is accepting ATM card



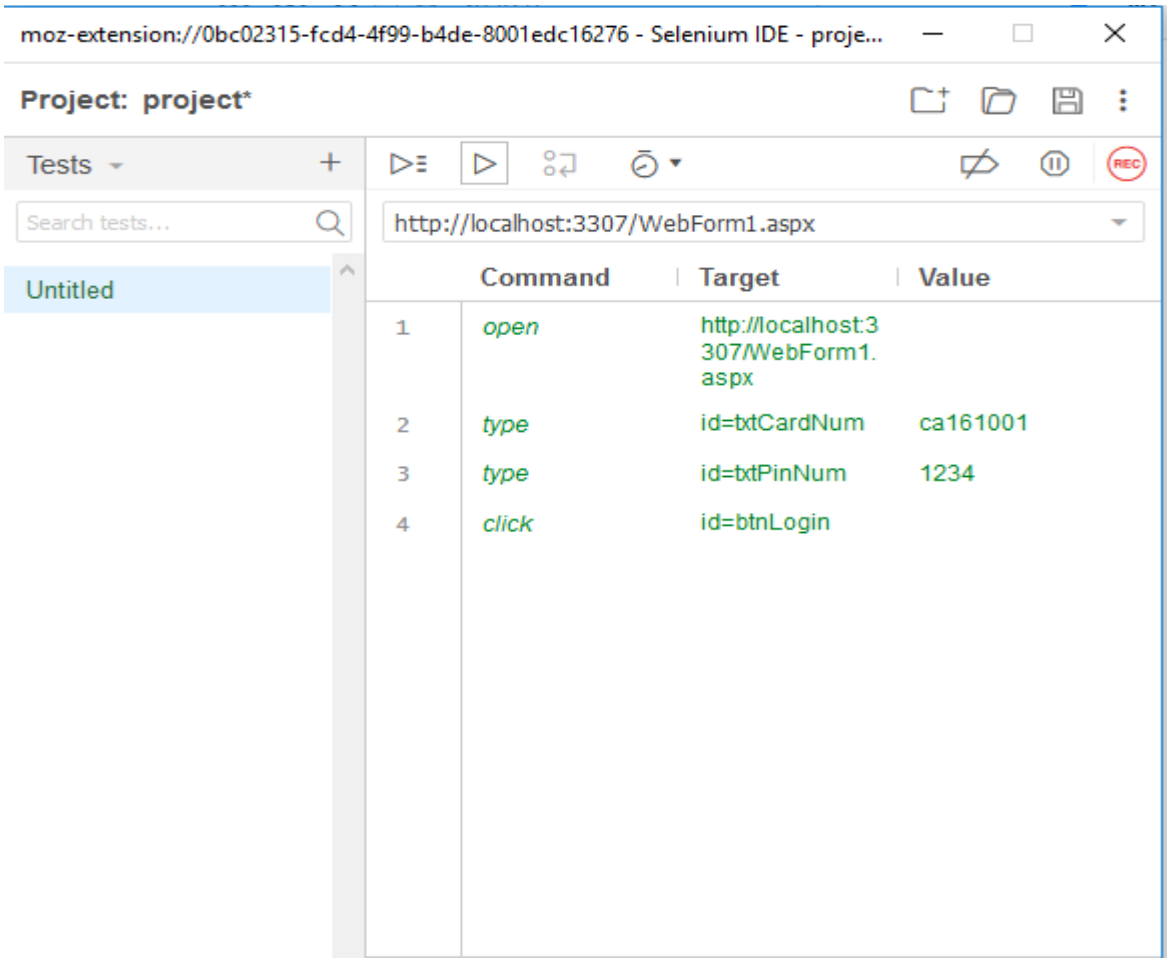


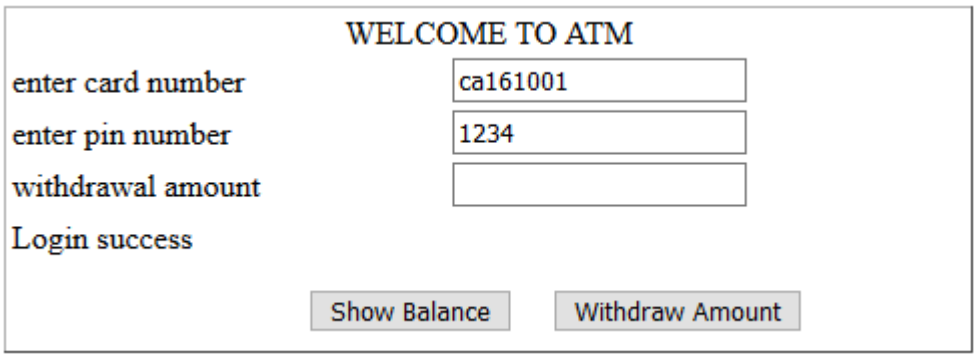
b.Machine is rejecting expired card.



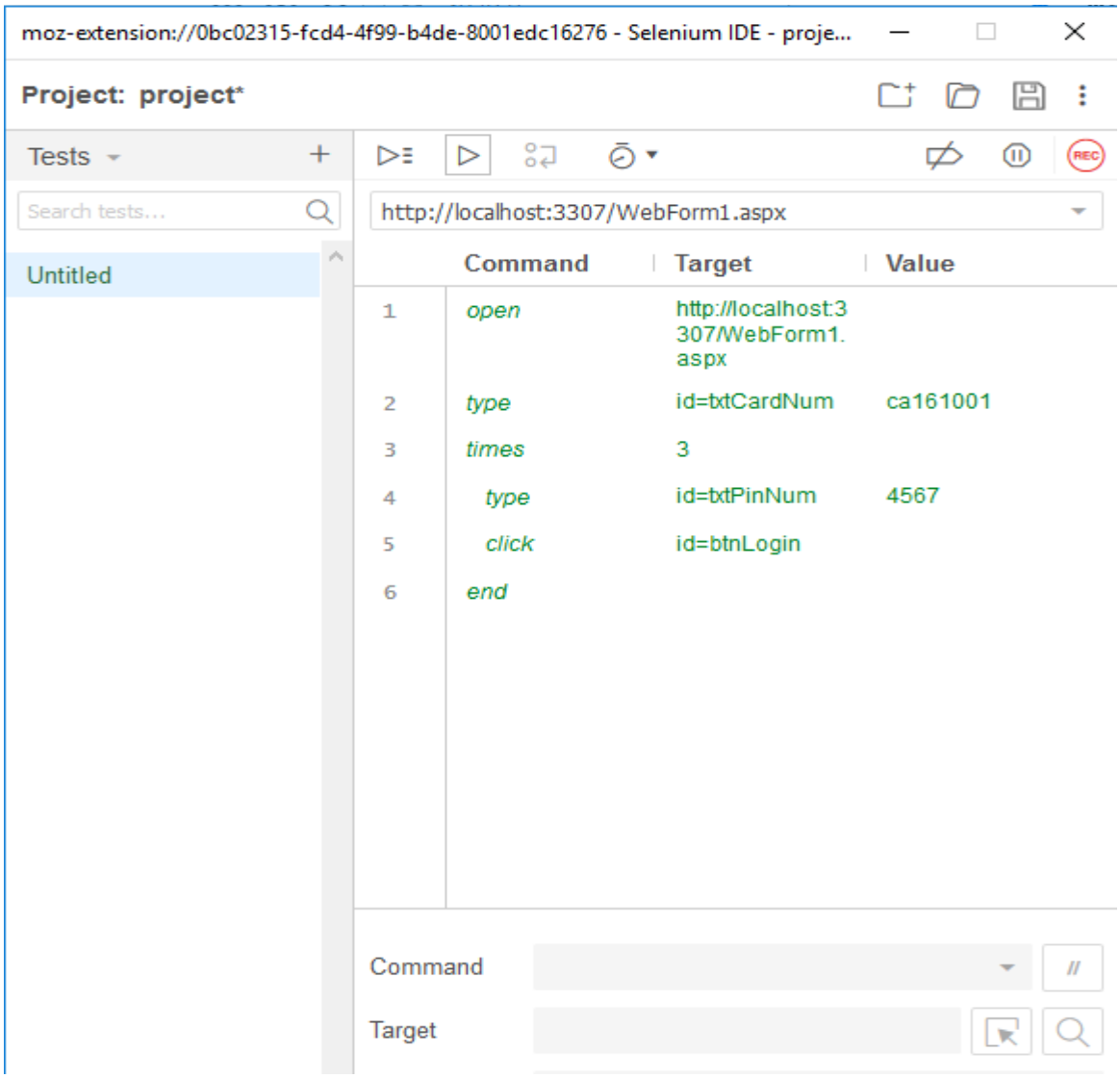


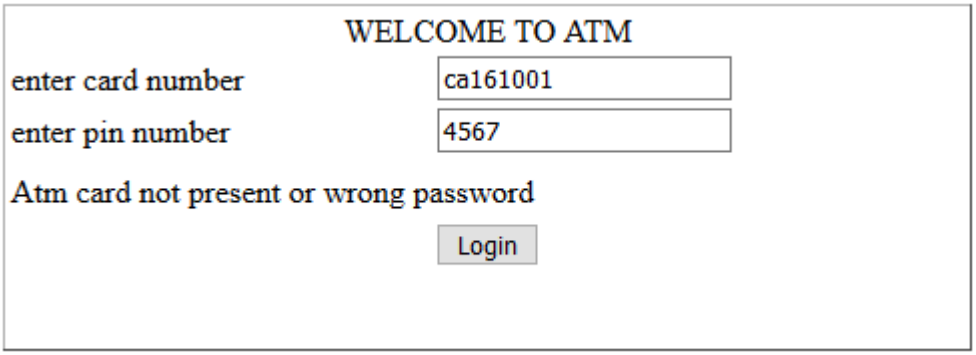
c.Successfull entry of PINnumber.





d.Unsuccessfull operation due to enter wrong PIN number 3 times.





2. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary value analysis, execute the test cases and discuss the results.

Testtriangle.java

```
package shri;
import java.util.Scanner;
public class testtringle {
                        public String tringle(int a,int b,int c)
                         {
                           String res1=" ";
                            if((a>10)||(b>10)||(c>10))
                            {
                                  System.out.print("Outof range");
                             }
                            if((a<b+c)&&(b<c+a)&&(c<a+b))
                            {
                                      if((a==b)&&(b==c))
                                      {
                                            res1="Equilateral";
                                       }
                                      else if((a!=b)&&(b!=a)&&(c!=a))
                                      {
                                            res1="scalene";
                                      }
                                      else
                                     {
                                            res1="Isoscales";
                                      }
```

```
            }
        else
          {
                res1="Triangle cannot be formed";

            }
          return(res1);
        }
}
```

Testequi.java

```
package mytestng;
import static org.testng.Assert.assertEquals;
import org.testng.annotations.*;
import shri.testtringle;
public class testequi
 {
        testtringle Res;
        public testequi() {
                Res=new testtringle();
        }
        @Test
        public void testE1()
        {
                assertEquals("Equilateral",Res.tringle(1,1,1));
        }
        @Test
        public void testE2()
        {
                assertEquals("Equilateral",Res.tringle(1,2,3));
        }
        @Test
    public void testE3()
   {
                assertEquals("scalene",Res.tringle(5,6,7));
        }
        @Test
        public void testE4()
        {
                assertEquals("scalene",Res.tringle(6,6,6));
        }
        @Test
        public void testE5()
```

```
        {
                assertEquals("Isoscales",Res.tringle(5,5,7));
        }
        @Test
        public void testE6()
        {
                assertEquals("Isoscales",Res.tringle(5,6,7));
        }
}
```

OUTPUT: Using TestNG

```
[RemoteTestNG] detected TestNG version 6.14.2

PASSED: teste1
FAILED: teste2
java.lang.AssertionError: expected [4 6 1987] but found [5 6 1987]



===============================================
    Default test
    Tests run: 2, Failures: 1, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 2, Failures: 1, Skips: 0
===============================================
```

3. Design, develop, code and run the program in any suitable language to implement the Next Date function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.

Nextdate.java

```
package shri;
import java.util.Scanner;
public class nextdate1 {

    public String date(int d,int m,int y)
    {
        int month[]={31,28,31,30,31,30,31,31,30,31,30,31};
        int nd,nm,ny,ndays;
        ndays=month[m-1];
        String ndate;
        if(y<=1812 && y>=2012)
        {
            System.out.println("Invalid year");

        }
        if(m==2 && y%100==0||y%400==0||y%4==0)
        {
            System.out.println("leap year");
        }
        nd=d+1;
        nm=m;
        ny=y;
        if(nd>ndays)
        {
            nd=1;
            nm++;
        }
```

```
  if(nm>12)
  {
        nm=1;
        ny++;
  }
  ndate=nd+" "+nm+" "+ny;
  return(ndate);
}}
```

Testdate.java

```
package testnextdate;
import static org.testng.Assert.assertEquals;
import org.testng.annotations.*;
import shri.nextdate1;

public class testdate
{
	nextdate1 dt;
	public testdate()
		{
			dt=new nextdate1();
		}
   @Test
   public void teste1()
	{
			assertEquals("5 6 1987",dt.date(4, 6, 1987));
		}
   @Test
   public void teste2()
	{
		assertEquals("5 6 1987",dt.date(3, 6, 1987));
	}
}
```

OUTPUT: Using TestNG

[RemoteTestNG] detected TestNG version 6.14.2

PASSED: teste1

FAILED: teste2

java.lang.AssertionError: expected [4 6 1987] but found [5 6 1987]

```
===============================================
    Default test
    Tests run: 2, Failures: 1, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 2, Failures: 1, Skips: 0
===============================================
```

4.Write a program to test the following constructs.

a. if….else

b. Nested….if

ifcond.java

```
package shri;
public class ifcond
{
	public String ev(int c)
	{
      String str;
		if(c%2==0)
			str="even";
		else
			str="odd";
	  return(str);
	}
	public String grade(int m)
	{
      String str;
		if(m<=100 && m>=75)
			str="Distinction";
		else if(m<75 && m>60)
			str="Grade A";
		else if(m<60 && m>=50)
			str="Grade B";
			else if(m<50 && m>=35)
				str="Grade c";
			else
				str="Fail";
	  return(str);
	}
}
```

Testifcond.java

```
package testif;
import static org.testng.Assert.assertEquals;
import org.testng.annotations.*;
import shri.ifcond;
public class Testifcond {
	ifcond e;
	public Testifcond()
	{
		e=new ifcond();
	}
	@Test
	public void teste1()
	{
		assertEquals("even",e.ev(2));
	}
	@Test
	public void teste2()
	{
		assertEquals("even",e.ev(3));
	}
	@Test
	public void testo1()
	{
		assertEquals("odd",e.ev(3));
	}
	@Test
	public void testo2()
	{
		assertEquals("odd",e.ev(4));
	}
  }
```

OUTPUT: Using TestNG

```
[RemoteTestNG] detected TestNG version 6.14.2
PASSED: teste1
PASSED: testo1
FAILED: teste2
java.lang.AssertionError: expected [odd] but found [even]


===============================================
    Default test
    Tests run: 4, Failures: 2, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 4, Failures: 2, Skips: 0
===============================================
```

Testnestif.java

```
package testnestif;
import static org.testng.Assert.assertEquals;
import org.testng.annotations.*;
import shri.ifcond;
public class testnestif
{
	ifcond e;
	public testnestif()
	{
		e=new ifcond();
	}
	@Test
	public void testg1()
	{
		assertEquals("Distinction",e.grade(85));
	}
	@Test
	public void testg2()
	{
		assertEquals("Distinction",e.grade(50));
	}
	@Test
	public void testg3()
	{
		assertEquals("Grade A",e.grade(70));
	}
	@Test
	public void testg4()
	{
		assertEquals("Grade A",e.grade(40));
	}
	@Test
```

```
	public void testg5()
	{
		assertEquals("Grade B",e.grade(55));
	}
	@Test
	public void testg6()
	{
		assertEquals("Grade B",e.grade(40));
	}
	@Test
	public void testg7()
	{
		assertEquals("Grade c",e.grade(45));
	}
	@Test
	public void testg8()
	{
		assertEquals("Grade c",e.grade(30));
	}
	@Test
	public void testg9()
	{
		assertEquals("Fail",e.grade(30));
	}
	@Test
	public void testg10()
	{
		assertEquals("Fail",e.grade(90));
	}
}
```

OUTPUT: Using TestNG

```
[RemoteTestNG] detected TestNG version 6.14.2
PASSED: testg1
PASSED: testg3
PASSED: testg5
PASSED: testg7
PASSED: testg9
FAILED: testg10
java.lang.AssertionError: expected [Distinction] but found [Fail]


===============================================
    Default test
    Tests run: 10, Failures: 5, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 10, Failures: 5, Skips: 0
===============================================
```

5. Program to test a count number of items present on a desktop.

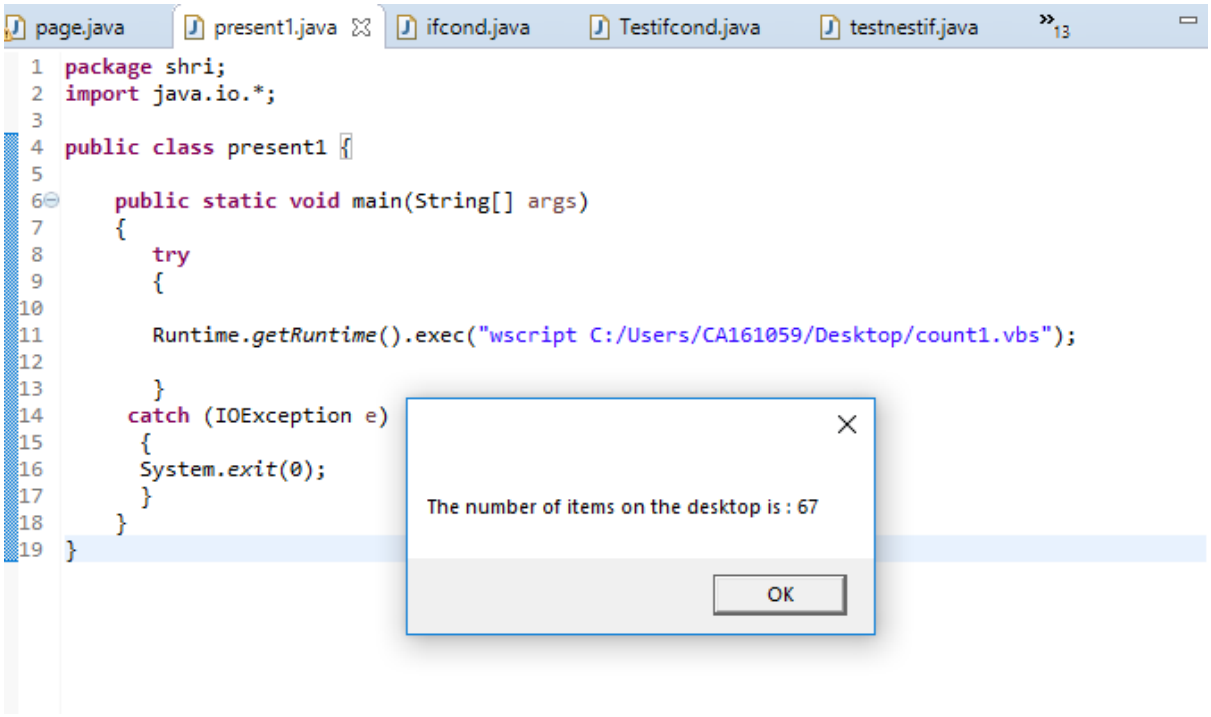
Present.java

```
package shri;
import java.io.*;
public class present1 {
        public static void main(String[] args)
        {
          try
          {
          Runtime.getRuntime().exec("wscript C:/Users/CA161059/Desktop/count1.vbs");
          }
         catch (IOException e)
          {
          System.exit(0);
          }
      }
}
```

Count1.vbs

```
Set fso = createobject("Scripting.FileSystemObject")

DesktopPath = CreateObject("WScript.Shell").SpecialFolders("Desktop")

'Files count

 a=fso.GetFolder(DesktopPath).Files.Count

 b=fso.GetFolder(DesktopPath).SubFolders.Count

c=a+b

msgbox "The number of items on the desktop is : " &c
```

OUTPUT : Using Eclipse

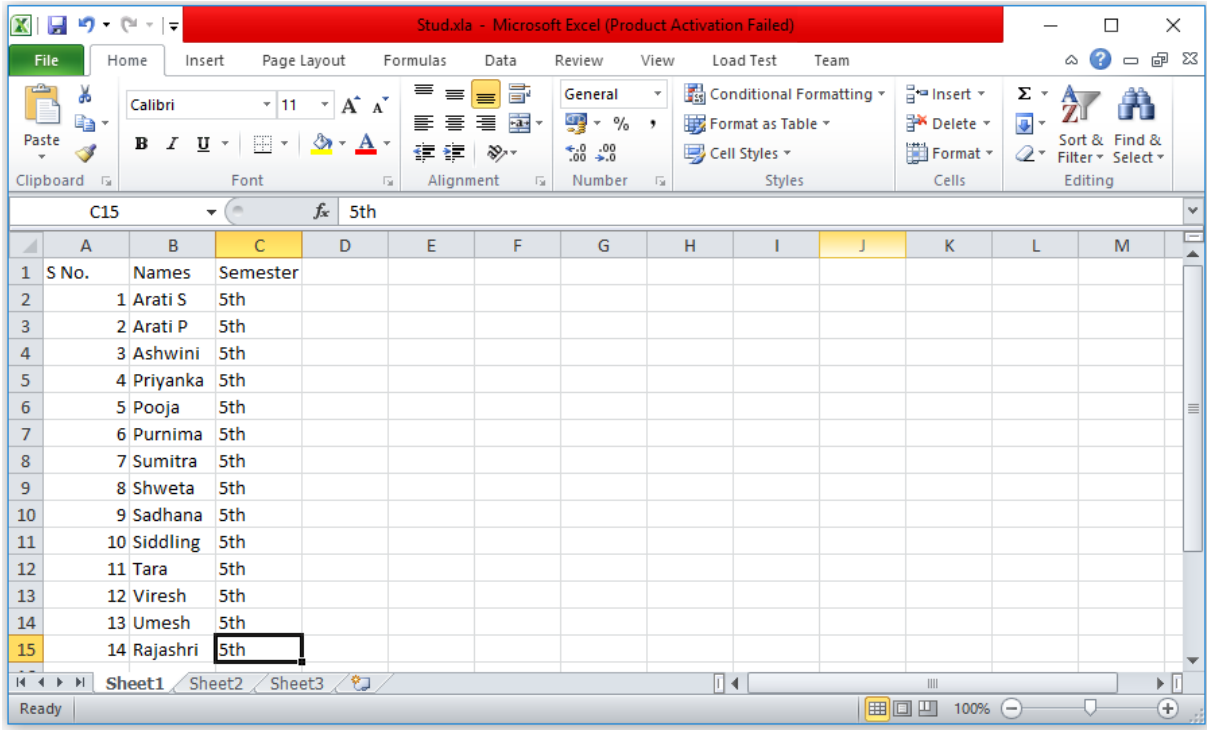


6. Program test a to update 10 student records into table into Excel file.

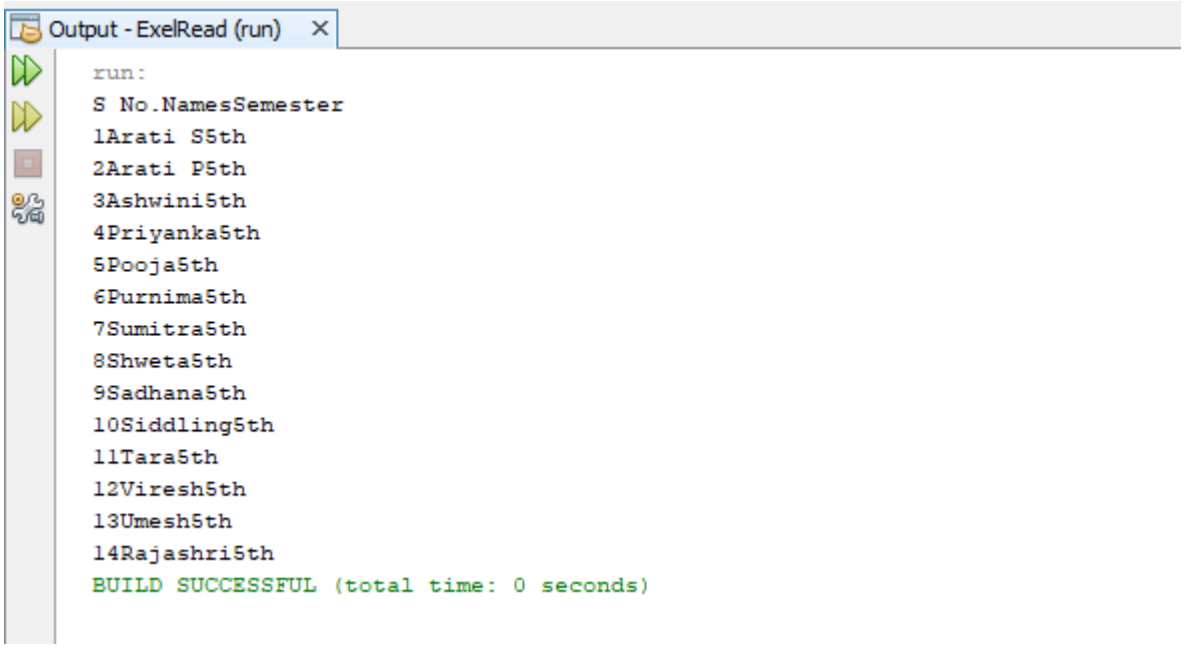
ExelRead.java

```
package exelread;
import java.io.File;
import jxl.Cell;
import jxl.Sheet;
import jxl.Workbook;
public class ExelRead
{
   public static void main(String[] args) throws Exception
        {
                        File f = new File("C:\\Users\\CA161055\\Desktop\\Exel.xla");
                        Workbook wb = Workbook.getWorkbook(f);
                Sheet S = wb.getSheet(0);
                        int row = S.getRows();
                int col = S.getColumns();
                        for(int i=0;i<row;i++)
                        {
                        for(int j=0;j<col;j++)
                        {
                        Cell c = S.getCell(j,i);
                        System.out.print(c.getContents());
                        }
        System.out.println("");
                        }
        }

}
```

INPUT:Using Exel

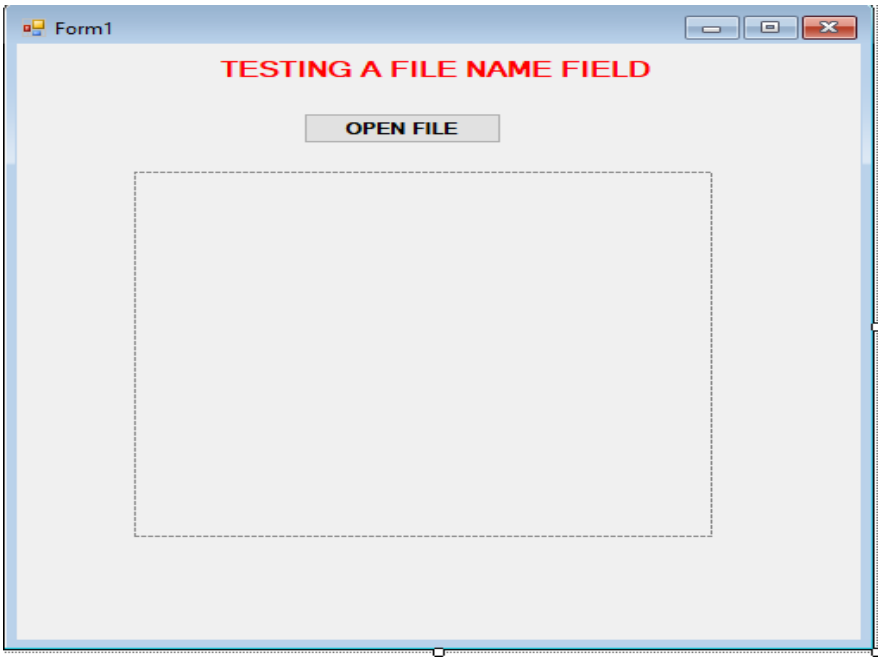


OUTPUT: Using NetBeans



7. Imagine testing a file name field. For example, go to an open file dialog, you can enter something into the field. Do a domain testing analysis: List a risk, equivalence classes appropriate to the risk, and best representatives of the equivalence classes. For each test case(use a best representative), briefly explain why this is a best representative. Keep doing this until you have listed 12 best-representative test cases.

Design:



Form1.cs

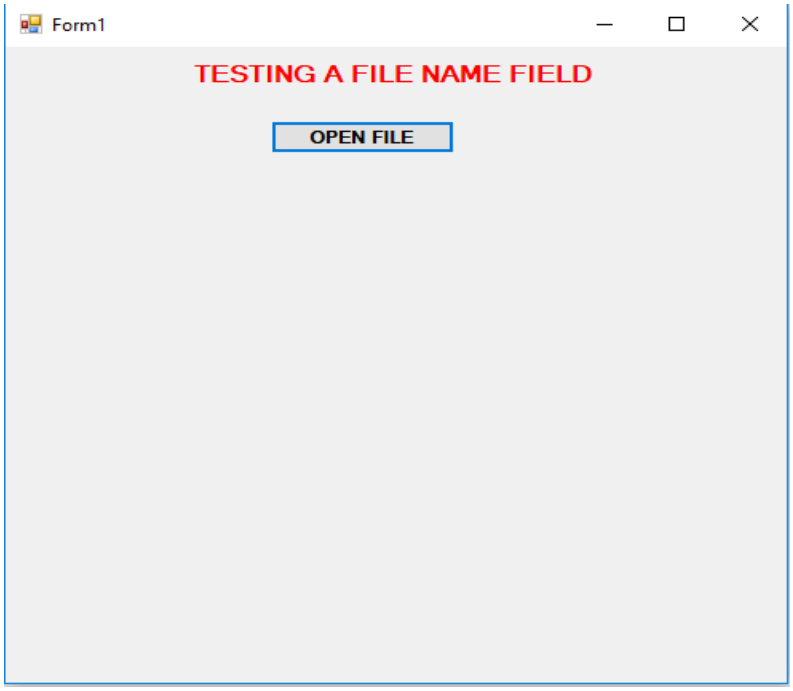
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace WindowsFormsApplication18
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
         public void button1_Click(object sender, EventArgs e)
      {
        openFileDialog1.InitialDirectory="C:\\FIELD";
          if(openFileDialog1.ShowDialog()==DialogResult.OK)
          {
```

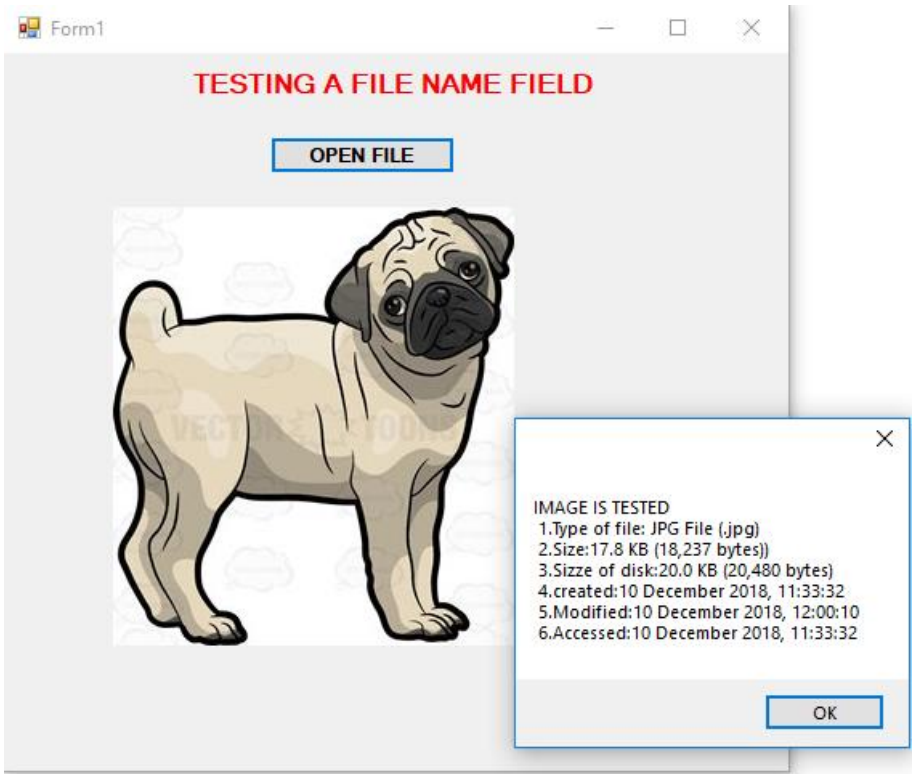
```
        pictureBox1.Image=Image.FromFile(openFileDialog1.FileName);

         MessageBox.Show("IMAGE IS TESTED\n 1.Type of file: JPG File (.jpg)\n 2.Size:17.8 KB (18,237 bytes))\n 3.Sizze of disk:20.0 KB (20,480 bytes)\n 4.created:10 December 2018, 11:33:32\n 5.Modified:10 December 2018, 12:00:10\n 6.Accessed:10 December 2018, 11:33:32");

      }

   }

  public void Form1_Load(object sender, EventArgs e)

      {

       }

          public void pictureBox1_Click(object sender, EventArgs e)

      {

      }

   }

}
```

Domain Testing Table:

Variable	Risk (Potential Failure)	Classes that Should Not Trigger the Failure	Classes that Might Trigger the Failure	Test (Best Representative)	Notes
String Input	Character outside of the range of a-z to A-Z might be accepted	Character A-Z to a-z	Characters length 0-10 with extension	False or -1	Large signed character loop to negative string, so -1 may cover both above and below the range of accepted inputs
Numeric Input	No integer or other special characters should be accepted	Numbers	Numeric characters such as number or symbols	True or A	Any numeric integer should present the same ability to uncover a bug
White space	What if nothing is entered at all?	Characters	Pressing the 'Enter' key	Force the program to accept a null space	Clear the input line and press 'Enter'

OUTPUT:


Form1
TESTING A FILE NAME FIELD
OPEN FILE

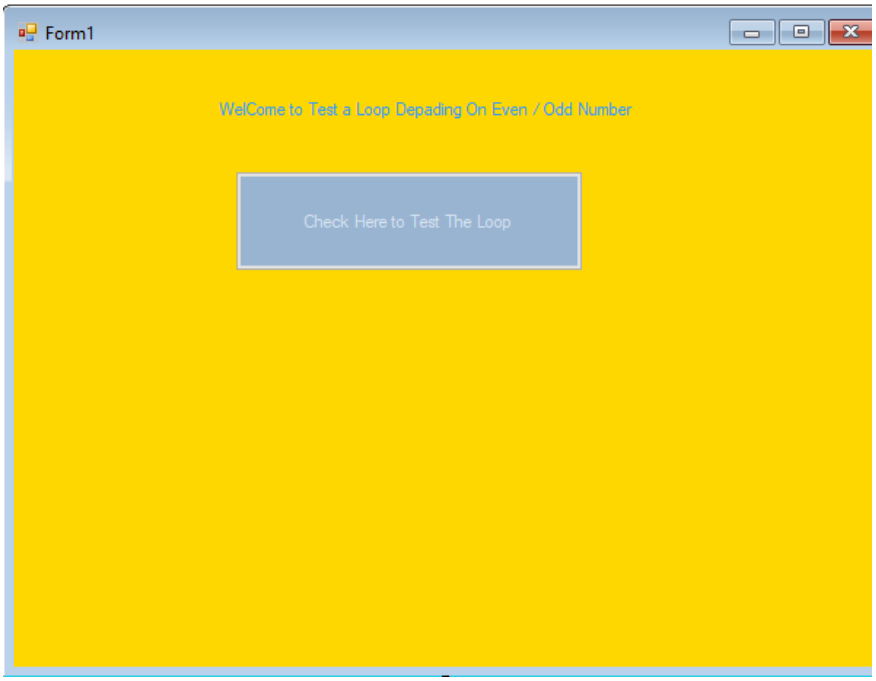

Form1
TESTING A FILE NAME FIELD
OPEN FILE
IMAGE IS TESTED
1.Type of file: JPG File (.jpg)
2.Size:17.8 KB (18,237 bytes))
3.Sizze of disk:20.0 KB (20,480 bytes)
4.created:10 December 2018, 11:33:32
5.Modified:10 December 2018, 12:00:10
6.Accessed:10 December 2018, 11:33:32
OK

8. A program is structured as follows:

- **It starts with a loop; the index variable can run from 0 to 20. The program can exit the loop normally at any value of the index.**
- **Coming out of the loop, there is a case statement that will branch to one of 10 places depending on the value of X. X is a positive, non-zero integer. It can have any value from 1 to MaxInt.**
- **In 9 of the 10 cases, the program executes X statements and then goes into another loop. If X is even, the program can exit the loop normally at any value of its index, from 1 to X. If X is odd, the program goes through the loop 666 times and then exits. In the 10th case, the program exits.**

Ignore the possibility of invalid values of the index variable on X. How many paths are there through this program? Please show and/or explain your calculations.

Design:



Form1.cs

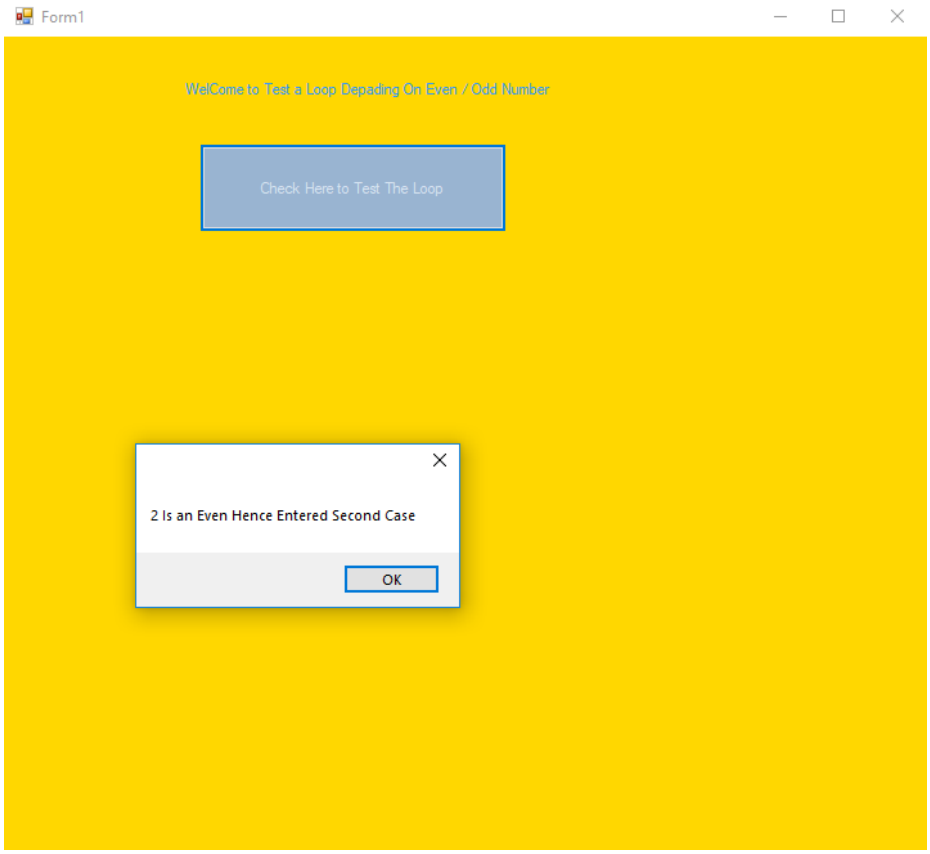
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication31
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }
          public void button1_Click(object sender, EventArgs e)
      {
          int i;
          do
         {
             for (i = 1; i < 20; i++)
```

```
            {
                if (i == 1 || i == 3 || i == 5 || i == 7 || i == 9)
            {
                    MessageBox.Show(i + " is a Odd Number Hence You Have Entered Loop");
                     }
                    else
                {
                    switch (i)
                    {
case 1:
            MessageBox.Show("First Case  ");
            break;
 case 2:
            MessageBox.Show(i + " Is an Even Hence Entered Second Case ");
            break;
case 3:
            MessageBox.Show("Three Case");
            break;
case 4:
            MessageBox.Show(i + " Is an Even Hence Entered Four Case ");
            break;
case 5:
             MessageBox.Show("Fifth Case");
```

```
            break;
 case 6:
            MessageBox.Show(i + " Is an Even Hence Entered Six Case ");
             break;
case 7:
            MessageBox.Show("Seven Case ");
            break;
case 8:
            MessageBox.Show(i + " Is an Even Hence Entered Eight Case ");
             break;
case 9:
            MessageBox.Show("Nine Case ");
            break;
             default:
            MessageBox.Show(i + " is a Above 10 case not Declare ");
            break;
                        }
                       }
                   }
                   Console.ReadLine();
            }
            while (i > 666);
 }
```

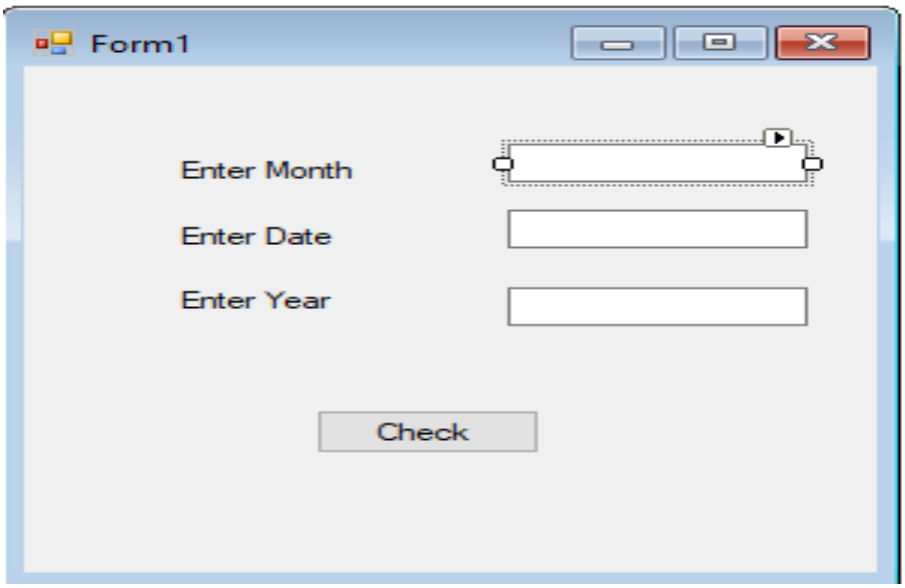
```
        private void Form1_Load(object sender, EventArgs e)
    {
        }
  }
}
```

OUTPUT:


Form1
WelCome to Test a Loop Depading On Even / Odd Number
Check Here to Test The Loop
2 Is an Even Hence Entered Second Case
OK

9. Imagine testing a date field. The field is of the form MM/DD/YYYY (two digit month, two digit day, 4 digit year). Does equivalence class analysis and boundary tests that you would run in order to test the field. (Don't bother with non-numeric values for these fields).

Design:



Form1.cs

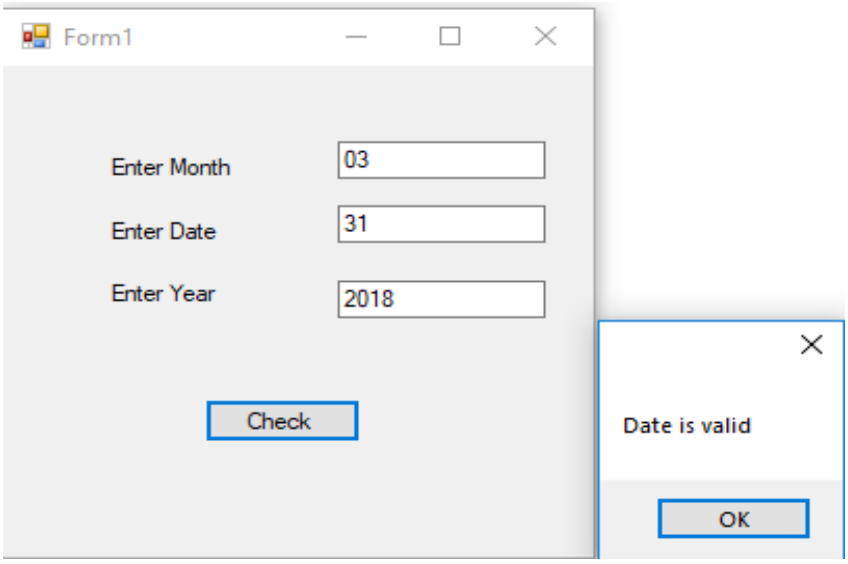
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Globalization;
using System.Text.RegularExpressions;
namespace DateFormat
{
   public partial class Form1 : Form
   {
      int dd, mm, yy,t,u,s;
      public Form1()
      {
         InitializeComponent();
      }
          private void button1_Click(object sender, EventArgs e)
      {
         if(yy>=1900 && yy<=9999)
               {
                  if(mm>=1 && mm<=12)
                   {
              if((dd>=1 && dd<=31) && (mm==1 || mm==3 || mm==5 || mm==7 || mm==8
                        || mm==10 || mm==12))
                     MessageBox.Show("Date is valid");
             else if((dd>=1 && dd<=30) && (mm==4 || mm==6 || mm==9 || mm==11))
                     MessageBox.Show("Date is valid");
            else if((dd>=1 && dd<=28) && (mm==2))
                     MessageBox.Show("Date is valid");
```

```
            else if(dd==29 && mm==2 && (yy%400==0 ||(yy%4==0 && yy%100!=0)))
                    MessageBox.Show("Date is valid.\n");
             else
                    MessageBox.Show("Day is invalid.\n");
                }
            else
            {
                    MessageBox.Show("Month is not valid.\n");
             }
      }
         else
          {
               MessageBox.Show("Year is not valid.\n");
          }
    }
     private void textBox1_TextChanged(object sender, EventArgs e)
{

    mm = int.Parse(textBox1.Text);

    t = int.Parse(textBox1.Text);
    if (t < 0 )
    {
       MessageBox.Show("Enter Month in Numaric form only!!! ");
    }
}
    private void textBox2_TextChanged(object sender, EventArgs e)
{
   dd = int.Parse(textBox2.Text);
    u = int.Parse(textBox1.Text);
    if (u < 0)
    {
       MessageBox.Show("Enter Date in Numaric form only!!! ");
```

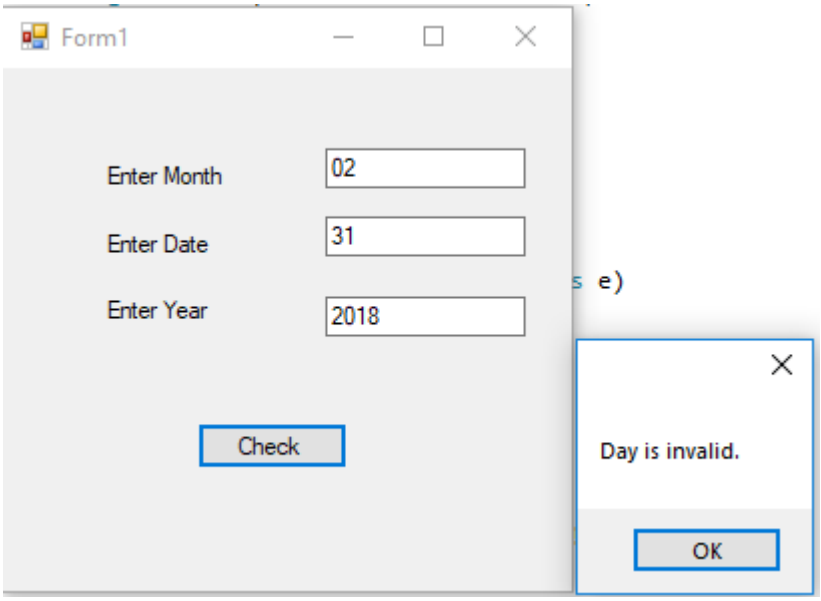
```
        }
    }
        private void textBox3_TextChanged(object sender, EventArgs e)
    {
        yy = int.Parse(textBox3.Text);
        s = int.Parse(textBox1.Text);
        if (s < 0)
        {
            MessageBox.Show("Enter Year in Numaric form only!!! ");
        }
         }
    }
}
```

OUTPUT:

1.

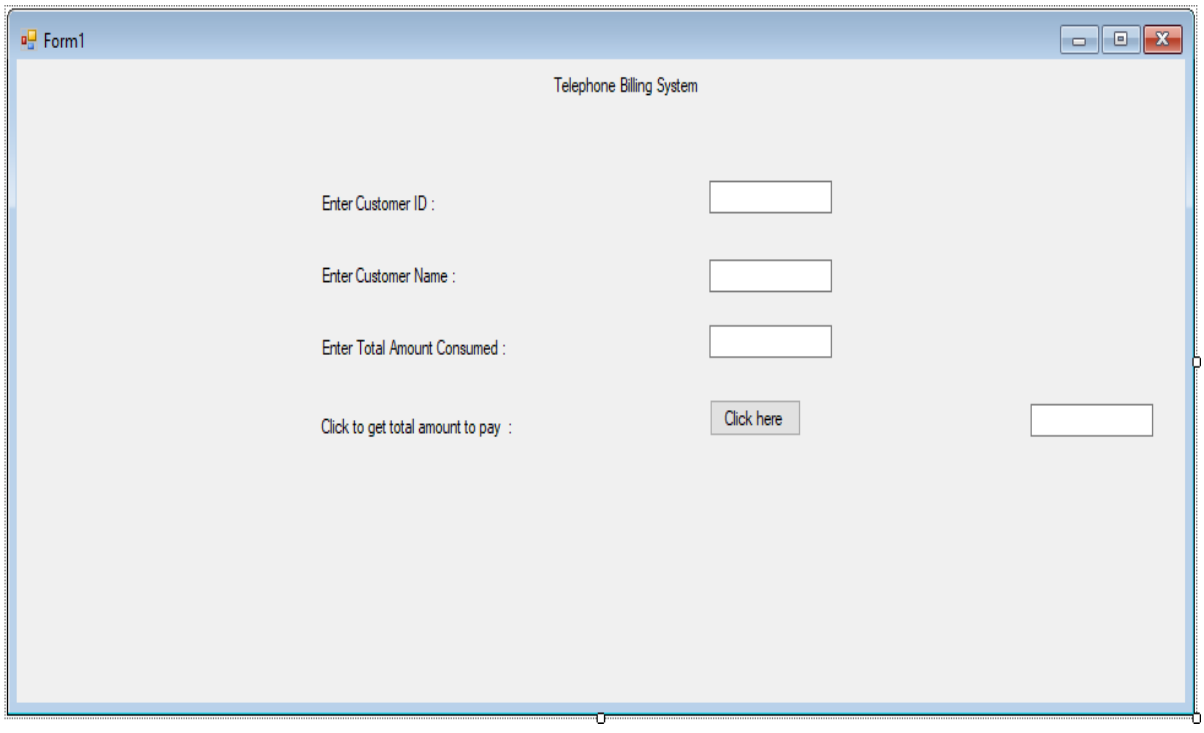

Form1
Enter Month
03
Enter Date
31
Enter Year
2018
Check
Date is valid
OK

2.


Form1
Enter Month
02
Enter Date
31
Enter Year
2018
Check
Day is invalid.
OK

10. The spring and fall change between Standard and Daylight savings time creates an interesting problem for telephone bills. Focus your thinking on the complications arising from the daylight saving time transitions. Create a table that shows risks, equivalence classes, boundary cases, and expected results for a long distance telephone service that bills calls at a flat rate of Rs.5.05 per minute. Assume that a chargeable time of a call begins when the called party answers, and ends when the calling party disconnects.

Design:



Form1.cs

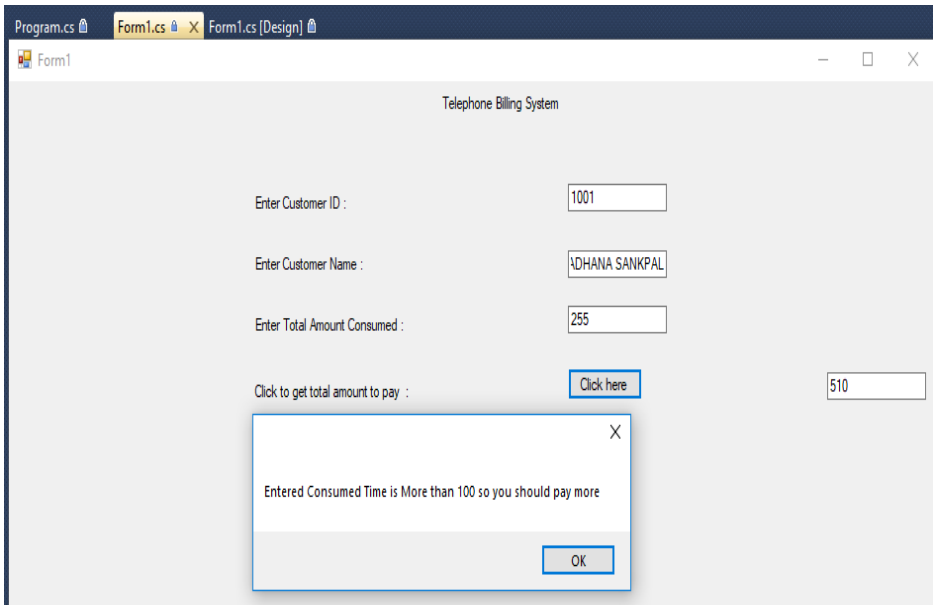
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace TelephoneBilling
{
   public partial class Form1 : Form
   {
      int r, s, t,u;
      public Form1()
      {
         InitializeComponent();
          }
          private void button1_Click(object sender, EventArgs e)
      {
         if (r < 100)
         {
            textBox4.Text = (int.Parse(textBox3.Text) * 1.0).ToString();
            MessageBox.Show("Entered Consumed Time is less than 100 so no extra pay");

         }
          else if (r >= 200 && r < 400)
         {
            textBox4.Text = (int.Parse(textBox3.Text) * 2.0).ToString();
            MessageBox.Show("Entered Consumed Time is More than 100 so you should pay
            more");
         }
         else if (r >= 400 && r < 600)
```

```
        {
            textBox4.Text = (int.Parse(textBox3.Text) * 3.0).ToString();
            MessageBox.Show("Entered Consumed Time is More than 100 so you should pay
            more");
        }
        else
        {
            textBox4.Text = (int.Parse(textBox3.Text) * 5.05).ToString();
            MessageBox.Show("Entered Consumed Time is More than 100 so you should pay
            more");
        }
      }
         private void textBox3_TextChanged(object sender, EventArgs e)
    {
        r = int.Parse(textBox3.Text);
    }
         private void textBox4_TextChanged(object sender, EventArgs e)
    {
        s = int.Parse(textBox4.Text);
    }
        private void textBox1_TextChanged(object sender, EventArgs e)
    {
        MessageBox.Show("Enter Customer ID in numeric form only!!! ");
        t = int.Parse(textBox1.Text);
        if (t <= 0)
        {
            }
    private void textBox2_TextChanged(object sender, EventArgs e)
    {
        }
  }
}
```

Telephone Billing Table :

Customer Id	Customer Name	(T)Called Time(in hours)	Boundary Cases	Equivalent Class	Additional Charge	Expected Charge (T*5.05)	Actual Charge	Status
1001	Miss.Sadhana Sankpal	100	T<1	(T<1&&T<=100),	NO CHARGE	100*1.0 =100	100	Pass
1002	Mr.Viresh Patil	00	T<1	(T<1&&T<=100),	Invalid	0*1.0 =0.0	0.0	Fail
1003	Miss.Priya Waganavar	200	T<1	(T<200&&T<=400)	Charges will Apply.	200*2.0 =400	400	Pass
1004	Miss.Shweta Kannakanwadi	300	T<1	(T<400&&T<=600	Charges will Apply.	300*3.0 =900	900	Pass

OUTPUT:


Program.cs
Form1.cs
Form1.cs [Design]
Form1
Telephone Billing System
Enter Customer ID :
1001
Enter Customer Name :
ADHANA SANKPAL
Enter Total Amount Consumed :
255
Click to get total amount to pay :
Click here
510
Entered Consumed Time is More than 100 so you should pay more
OK

11. Simulation of paging Technique in operating System.

Page.java

```
Package shri;
import java.util.Scanner;
public class page {
        public static void main(String[] args)
        {
         Scanner in=new Scanner(System.in);
      int memsize;
      int pagesize,nofpage;
      int[] p=new int[100];
      int frameno,offset;
      int logadd,phyadd;
      int i;
      int choice=0;
      System.out.println("\nEnter your memory size");
      memsize=in.nextInt();
      System.out.println("\nEnter page size");
      pagesize=in.nextInt();
      nofpage=memsize/pagesize;
      for(i=0;i<nofpage;i++)
      {
         System.out.println("\nEnter the frame of page"+(i+1)+":");
         p[i]=in.nextInt();
      }
      do
      {
         System.out.println("Enter a logical address:");
         logadd=in.nextInt();
         frameno=logadd/pagesize;
         offset=logadd%pagesize;
         phyadd=(p[frameno]*pagesize)+offset;
         System.out.println("Physical address is:"+phyadd);
```

```
        System.out.println("Do you want to continue(1/0)?:");
        choice=in.nextInt();
    }while(choice==1);
    }
}
```

OUTPUT: Using Eclipse

Enter your memory size

20

Enter page size

5

Enter the frame of page1:

1

Enter the frame of page2:

2

Enter the frame of page3:

3

Enter the frame of page4:

4

Enter a logical address:

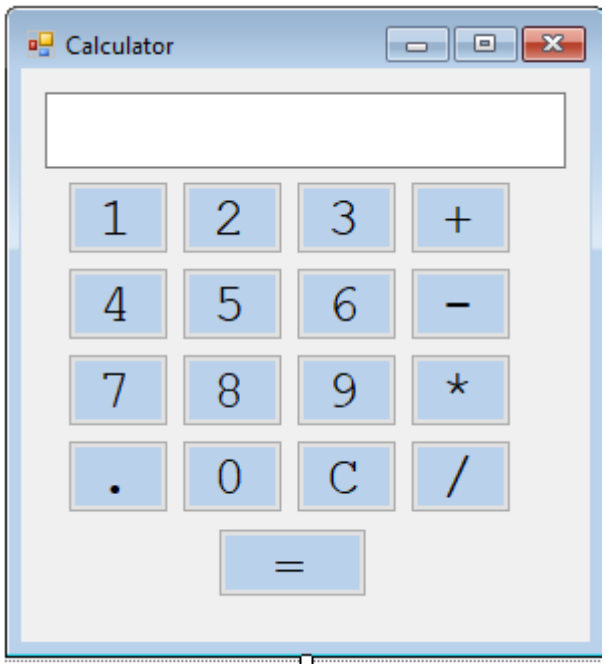
6

Physical address is:11

Do you want to continue(1/0)?:

12. Simulate a simple Calculator.

Calculator: FormDisign



Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication14
{
   public partial class Form1 : Form
   {
      double FirstNumber;
      string Operation;
      public Form1()
      {
         InitializeComponent();
      }
      private void n1_Click(object sender, EventArgs e)
      {
         if (textBox1.Text == "0" && textBox1.Text != null)
         {
            textBox1.Text = "1";
         }
         else
         {
            textBox1.Text = textBox1.Text + "1";
         }
      }
      private void n2_Click(object sender, EventArgs e)
      {
         if (textBox1.Text == "0" && textBox1.Text != null)
```

```
    {
        textBox1.Text = "2";
    }
    else
    {
        textBox1.Text = textBox1.Text + "2";
    }
}
private void n3_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
    {
        textBox1.Text = "3";
    }
    else
    {
        textBox1.Text = textBox1.Text + "3";
    }
}
private void n4_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
    {
        textBox1.Text = "4";
    }
    else
    {
        textBox1.Text = textBox1.Text + "4";
    }
}
private void n5_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
```

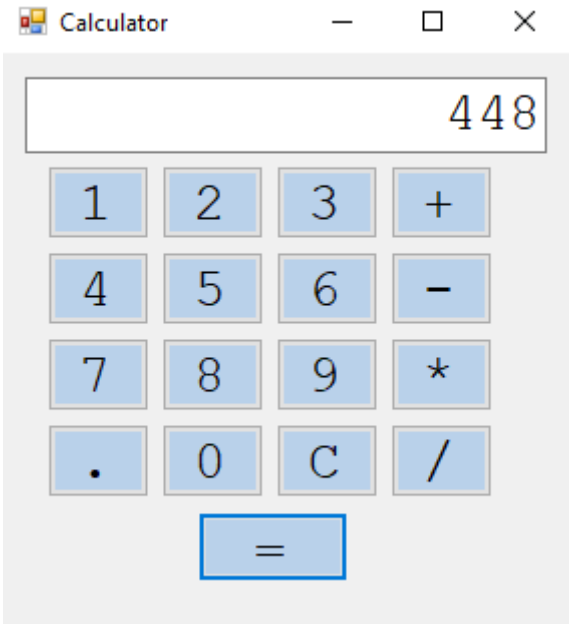
```
    {
        textBox1.Text = "5";
    }
    else
    {
        textBox1.Text = textBox1.Text + "5";
    }
}
private void n6_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
    {
        textBox1.Text = "6";
    }
    else
    {
        textBox1.Text = textBox1.Text + "6";
    }
}
private void n7_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
    {
        textBox1.Text = "7";
    }
    else
    {
        textBox1.Text = textBox1.Text + "7";
    }
}
private void n8_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
```

```
    {
        textBox1.Text = "8";
    }
    else
    {
        textBox1.Text = textBox1.Text + "8";
    }
}
private void n9_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "0" && textBox1.Text != null)
    {
        textBox1.Text = "9";
    }
    else
    {
        textBox1.Text = textBox1.Text + "9";
    }
}
private void n0_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + "0";
}

private void ndot_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox1.Text + ".";
}
private void bad_Click(object sender, EventArgs e)
{
    FirstNumber = Convert.ToDouble(textBox1.Text);
    textBox1.Text = "0";
    Operation = "+";
```

```
}
private void bsub_Click(object sender, EventArgs e)
{
   FirstNumber = Convert.ToDouble(textBox1.Text);
   textBox1.Text = "0";
   Operation = "-";
}
private void bmul_Click(object sender, EventArgs e)
{
   FirstNumber = Convert.ToDouble(textBox1.Text);
   textBox1.Text = "0";
   Operation = "*";
}
private void button4_Click(object sender, EventArgs e)
{
   FirstNumber = Convert.ToDouble(textBox1.Text);
   textBox1.Text = "0";
   Operation = "/";
}
private void bc_Click(object sender, EventArgs e)
{
   textBox1.Text = "0";
}
private void bequal_Click(object sender, EventArgs e)
{
   double SecondNumber;
   double Result;
   SecondNumber = Convert.ToDouble(textBox1.Text);
   if (Operation == "+")
   {
      Result = (FirstNumber + SecondNumber);
      textBox1.Text = Convert.ToString(Result);
      FirstNumber = Result;
```

```
        }
        if (Operation == "-")
        {
            Result = (FirstNumber - SecondNumber);
            textBox1.Text = Convert.ToString(Result);
            FirstNumber = Result;
        }
        if (Operation == "*")
        {
            Result = (FirstNumber * SecondNumber);
            textBox1.Text = Convert.ToString(Result);
            FirstNumber = Result;
        }
        if (Operation == "/")
        {
            if (SecondNumber == 0)
            {
                textBox1.Text = "0";
            }
            else
            {
                Result = (FirstNumber / SecondNumber);
                textBox1.Text = Convert.ToString(Result);
                FirstNumber = Result;
            }
        }
    }
  }
}
```

OUTPUT:



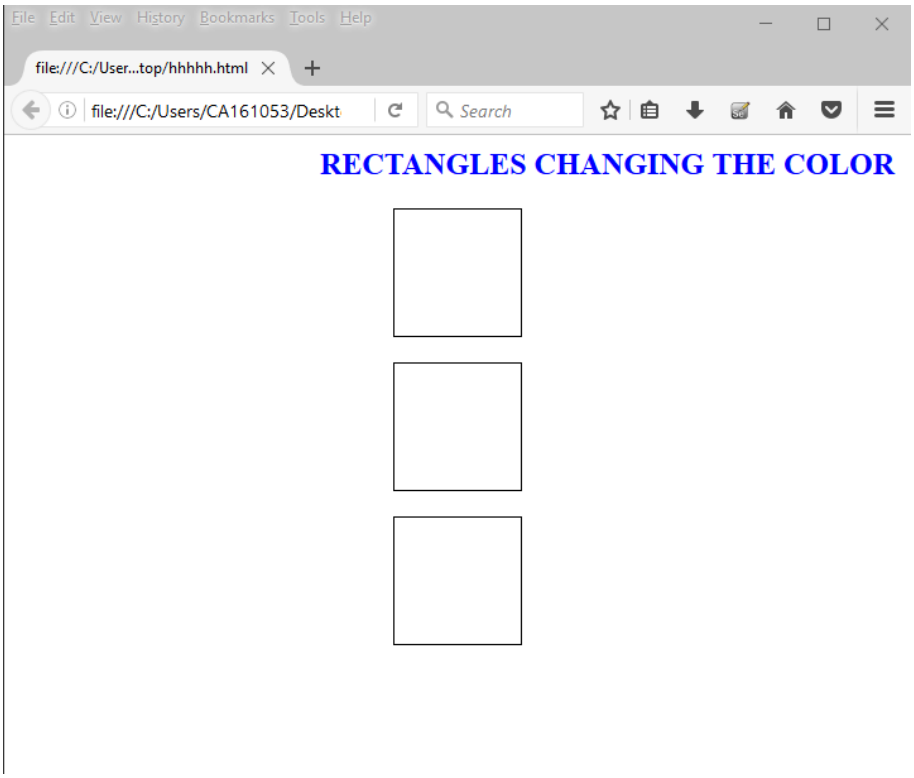
13.Students are given a program that draws a sequence of differently coloured rectangles and are asked to modify the code so that the result will be a sequence of rectangles that gradually change in colour from the colour of the first to the colour of the last. The algorithm to blend the correct colour for each rectangle requires the students to use proportions based on the distance each rectangle is from the first and last rectangles.

Rectangle.html

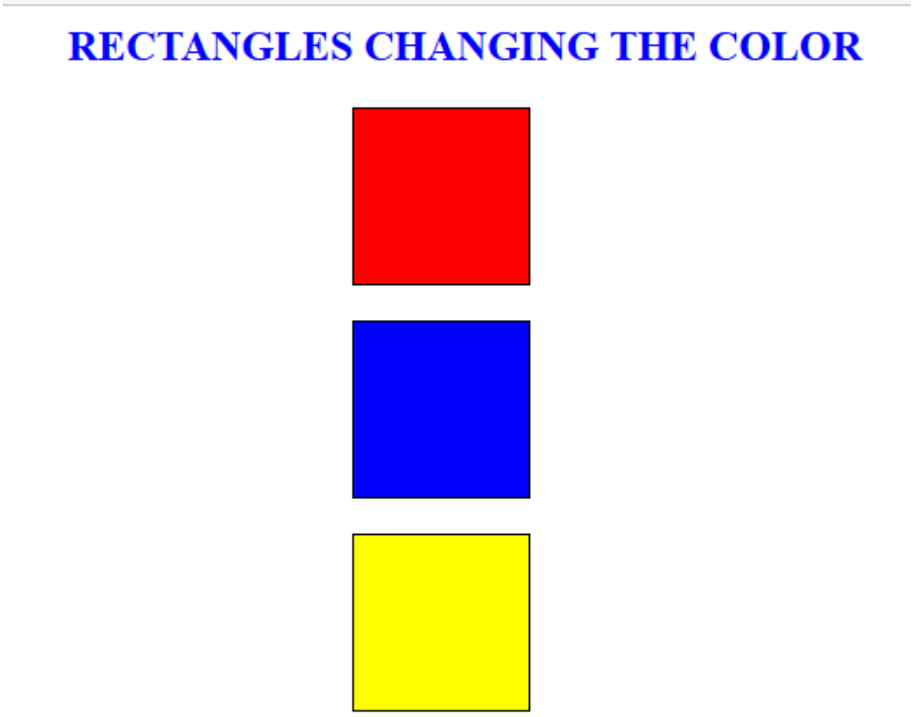
```
<html>
<head>
<style>
div#box
{
   width:100px;
   height:100px;
   background-color: white;
   border-color: black;
   border-style: solid;
   border-width: 1px 1px 1px 1px;
}
div#box1
{
   width:100px;
   height:100px;
   background-color: white;
   border-color: black;
   border-style: solid;
   border-width: 1px 1px 1px 1px;
}
div#box2
{
   width:100px;
   height:100px;
```

```
    background-color: white;
    border-color: black;
    border-style: solid;
    border-width: 1px 1px 1px 1px;
}
</style>
</head>
<script>
var colors = ["red","blue","green","yellow"];//
var index = 0;
function button_click() {
   index = (index ) % colors.length;
   document.getElementById("box").style.backgroundColor = colors[index];
   index = (index + 1) % colors.length;
   document.getElementById("box1").style.backgroundColor = colors[index];
   index = (index + 2) % colors.length;
   document.getElementById("box2").style.backgroundColor = colors[index];
 //  index = (index + 3) % colors.length;
   //document.getElementById("box3").style.backgroundColor = colors[index];
   }
</script>
<h2>
<marquee><B><font color="blue">RECTANGLES CHANGING THE COLOR
<b></font></marquee>
</h2><center>
<div id="box" onclick="button_click();"></div><br>
<div id="box1" onclick="button_click();"></div><br>
<div id="box2" onclick="button_click();"></div><br>
</center>
 </html>
```

INPUT:


File Edit View History Bookmarks Tools Help
file:///C:/User...top/hhhhh.html
file:///C:/Users/CA161053/Deskt
Search
RECTANGLES CHANGING THE COLOR

OUTPUT:



14. Simulate of the M/G/1 queue.

Queue.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define BUSY 1
#define IDLE 0
#define MAXIMUM_NUMBER_OF_CUSTOMERS 1000
/*--- Function prototypes ---------------------------------------------*/
double Exponential(double x);
void write_file(double a[][2],int m);
void Add_to_queue(int i,double time);
double arrival(int i,double current_time_index,int lamda);
double bernoulli(float a);
double start_service(double current_time_index,int mu,int lamda);
double departure(double current_time_index,int mu,int lamda);
double arrival_time[1000];
double departure_time[1000];
double service_start_time[1000];
double waiting_time[1000];
double service_time [1000];
double filearray[5000][2];
int main_counter=0;
struct queue

{
int customer_number;
double arrival_time_in_system;
double start_service_time;
double service_time;
}
```

```
customer_queue[1000];
typedef struct queue CUST_QUEUE;
struct queue2

{
int customer_number;
double arrival_time_in_system;
double start_service_time;
double service_time;
}

ultimate_queue[1000];
int number_in_queue=0;
int server_status=0;
int queue_head=0;
int main_count=0;
int number_in_system=0;
int customer_count=0;
int customer_no=0;
int k=0,d=0;
//*************************************************************************//
//    Main program
//*************************************************************************//
void main()
{
 int lamda;// Mean time between arrivals
 int mu; // Mean service time
 int i=0,j=0,m=0,z=0;
 double time = 0.0,new_time_index=0.0;// Simulation time
 double sum=0;
 double average_waiting_time;
 long int n = 0;// Number of customers in the system
 // Number of customers that have come to the system till now
 double current_time_index=0;
 int max=MAXIMUM_NUMBER_OF_CUSTOMERS;
```

```
 int number_before_service_is_started;
 printf("Enter the arrival rate\n");
 scanf("%d",&lamda);
 printf("Enter the service rate\n");
 scanf("%d",&mu);
 printf("enter the value of K (the number of people required in the queue before the service
starts\n");
 scanf("%d",&number_before_service_is_started);
 // Main simulation loop
 while (customer_count<max)
 {
  if(server_status==IDLE){
  /* put the first k customers in the queue */
   queue_head=0;
   filearray[d][0]=number_in_system;
   filearray[d++][1]=current_time_index;
   for(i=0;i<number_before_service_is_started;i++)
   {
    /* arrival event ...initially k people have to come in */
    new_time_index=arrival(customer_count,current_time_index,lamda);
    Add_to_queue(customer_count-1,new_time_index);
    //-1 because it had been incremented in arrival function
    current_time_index=new_time_index;
   }
  /* start service now */
  server_status=BUSY;
  current_time_index=start_service(current_time_index,mu,lamda);
  }
 } //end while
for(i=0;i<MAXIMUM_NUMBER_OF_CUSTOMERS;i++)
{
 departure_time[i]=ultimate_queue[i].service_time+ultimate_queue[i].start_service_time;
```

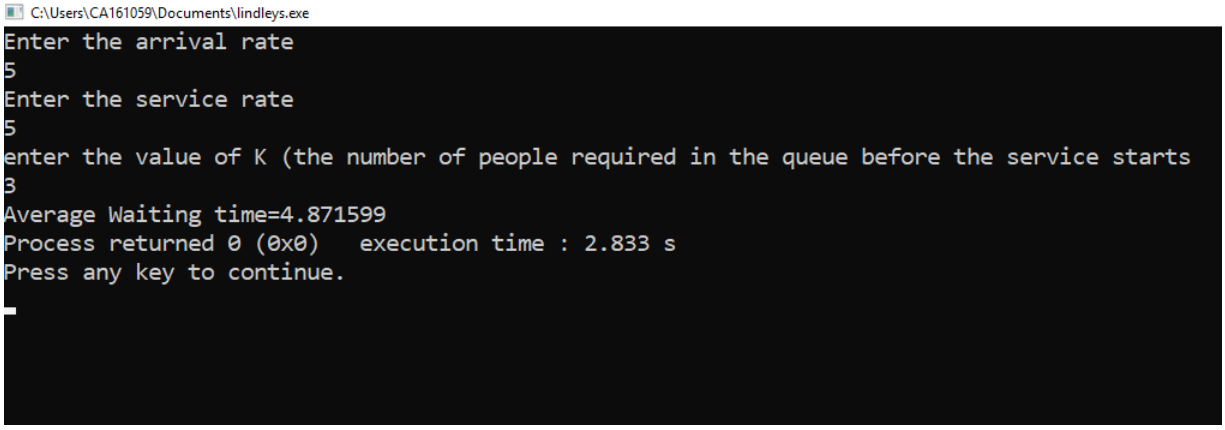
```
waiting_time[i]= ultimate_queue[i].start_service_time-
ultimate_queue[i].arrival_time_in_system;
 sum+=waiting_time[i];
}
average_waiting_time=sum/MAXIMUM_NUMBER_OF_CUSTOMERS;
printf("Average Waiting time=%f",average_waiting_time);
write_file(filearray,MAXIMUM_NUMBER_OF_CUSTOMERS);
}

//end main
//*********************************************************************//
// Begin function to generate an exponential random number
//*********************************************************************//
double Exponential(double x)
{
 double z;
 // a uniform RV (0 < z < 1)
 do
 {
 z = ((double) rand()/RAND_MAX);
 }
 while ((z==0) || (z==1));
 return((-1/x)*log(z));// formula for exponential Random Variable
}
//*********************************************************************//
// Begin function to simulate an arrival event
//*********************************************************************//
//pass the current_time_index and return the arrival time
double arrival(int customer_no,double current_time_index,int lamda)
{
 static int k=0;
 int i;
 double copy=0;
 double random_time;
```

```
//set the arrival time for the 1000 customers in one go
 if(k==0)

{
  for(i=0;i<MAXIMUM_NUMBER_OF_CUSTOMERS;i++)

{
  random_time=Exponential(lamda);
  arrival_time[i]=copy+random_time;
  copy=arrival_time[i];
  }
 }
 ultimate_queue[customer_count].customer_number=customer_count;
 ultimate_queue[customer_no].arrival_time_in_system=arrival_time[customer_no];
 customer_no++;
 customer_count++;
 filearray[d][0]=number_in_system;
 filearray[d++][1]=current_time_index;
 number_in_system++;
 main_counter++;
 k++;
 return(arrival_time[customer_no-1]);
}
//*************************************************************************//
//   Begin function to add a customer to queue
//*************************************************************************//
void Add_to_queue(int customer_nomber,double time)
{
 customer_queue[number_in_queue].customer_number=customer_nomber ;
 customer_queue[number_in_queue].arrival_time_in_system=time;
 number_in_queue++;
}
//*************************************************************************//
//  Begin function to Start the service
//*************************************************************************//
```

```
double start_service(double current_time_index,int mu,int lamda)
{
 double dept_time_index;
 int i;
 while(number_in_queue)
 {
 /* start servicing the first guy in the queue set his service start_time */
  i=customer_queue[queue_head].customer_number;
  service_start_time[i]=current_time_index;
  customer_queue[queue_head].start_service_time=current_time_index;
  ultimate_queue[main_count].start_service_time=current_time_index;
  //call the departure
  dept_time_index=departure(current_time_index,mu,lamda);
  current_time_index=dept_time_index;
 }
 server_status=IDLE;
 return(current_time_index);
}
//*************************************************************************//
//    Function to start the departure
//*************************************************************************//
double departure(double current_time_index,int mu,int lamda)
{
 int i;
 double dept_time_index,serv_time,new_time_index;
 serv_time=Exponential(mu);
 customer_queue[queue_head].service_time=serv_time;
 ultimate_queue[main_count].service_time=serv_time;
 dept_time_index=serv_time+current_time_index;
 //if an arrival occurs while the customer is being served then signal an arrival
 //and add him to the queue
 while(arrival_time[customer_count]<dept_time_index &&
customer_count<MAXIMUM_NUMBER_OF_CUSTOMERS)
```

```
{
 new_time_index=arrival(customer_count,current_time_index,lamda);
 Add_to_queue(customer_count-1,new_time_index);
 }
 number_in_queue--;
 filearray[d][0]=number_in_system;
 filearray[d++][1]=current_time_index;
 number_in_system--;
 //shift the queue up by one
 for(i=1;i<MAXIMUM_NUMBER_OF_CUSTOMERS;i++)

{
 customer_queue[i-1].arrival_time_in_system=customer_queue[i].arrival_time_in_system;
 customer_queue[i-1].customer_number=customer_queue[i].customer_number;
 customer_queue[i-1].service_time=customer_queue[i].service_time;
 customer_queue[i-1].start_service_time=customer_queue[i].start_service_time;
 }
 main_count++;
 return(dept_time_index);
}
//*************************************************************************//
//   Begin function to write to a file
//*************************************************************************//
void write_file(double filearray[][2],int m)
{
 FILE *fp;
 int i;
 fp=fopen("output.txt","w");
 for(i=0;i<m;i++)
 {
fprintf(fp,"%f\t\t %f\n",filearray[i][1],filearray[i][0]);
 }
 fclose(fp);
}
```

OUTPUT:


C:\Users\CA161059\Documents\lindleys.exe
Enter the arrival rate
5
Enter the service rate
5
enter the value of K (the number of people required in the queue before the service starts
3
Average Waiting time=4.871599
Process returned 0 (0x0) execution time : 2.833 s
Press any key to continue.