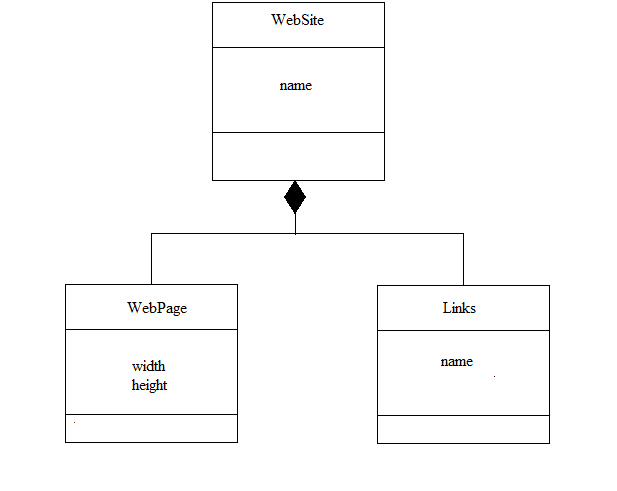
|  |  |  |
| --- | --- | --- |
|  | **NED University of Engineering and Technology** | **Object Oriented Programming(C++ and C#)** |
| **Department of Software Engineering** | **First Year(Section A and B)** |

**ASSIGNMENT # 2**

Q.1 **Problem Statement:**

We have the following part of class diagram showing composition relationship:



You are required to implement above class diagram (complete program) in C++ with all data members, constructors, member functions and concept (composition) given in the class diagram/table 1.

See the sample output to view the messages you need to print in constructors of all classes. In case of parameterize constructors, you can select any of data member(s) according to the type of constructor.

**PROGRAM**

#include<conio.h>

#include<iostream>

#include<string>

using namespace std;

class webpage

{

private:

double width;

double length;

public:

webpage(double w, double l) :width(w), length(l)

{}

void setwidth(double w)

{

width = w;

}

void setlength(double l)

{

length = l;

}

void showdata()

{

cout << " width of a webpage: " << width << endl;

cout << " length of a webpage: " << length << endl;

}

};

class link

{

private:

string name;

public:

link(string n) :name(n)

{}

void showdata()

{

cout << " link of your website: " << name << endl;

}

};

class website

{

private:

string name;

webpage w1;

link l1;

public:

website(string nd, webpage w, link l) :name(nd), w1(w), l1(l)

{}

string setname(string n)

{

name = n;

}

void info()

{

cout << "\nwebsite name is: " <<name<< endl;

cout << "webpage info: " << endl;w1.showdata();

cout << "link info: " << endl;l1.showdata();

}

};

int main()

{

webpage w1(5.45, 6.8);

w1.setlength(5);

w1.setwidth(6);

link l1("@mariamnaz");

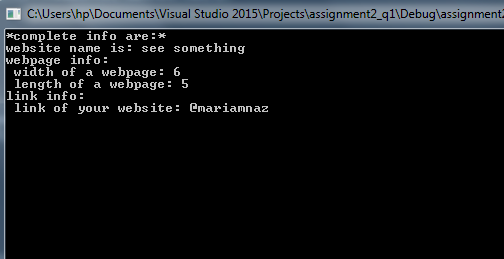
website s("see something", w1, l1);

cout << "\*complete info are:\*";s.info();

\_getch();

}

**OUTPUT**



Q.2 Create an application in C# than can generate raining effect by using Labels or Picture Box and Timer Control.

**PROGRAM**

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 raining\_effect\_in\_csharp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

private void pictureBox18\_Click(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

int j = -10;

pictureBox1.Location = new Point(pictureBox1.Location.X, pictureBox1.Location.Y+1);

pictureBox2.Location = new Point(pictureBox2.Location.X, pictureBox2.Location.Y + 1);

pictureBox3.Location = new Point(pictureBox3.Location.X, pictureBox3.Location.Y + 1);

pictureBox4.Location = new Point(pictureBox4.Location.X, pictureBox4.Location.Y + 1);

pictureBox5.Location = new Point(pictureBox5.Location.X, pictureBox5.Location.Y + 1);

pictureBox6.Location = new Point(pictureBox6.Location.X, pictureBox6.Location.Y + 1);

pictureBox7.Location = new Point(pictureBox7.Location.X, pictureBox7.Location.Y + 1);

pictureBox8.Location = new Point(pictureBox8.Location.X, pictureBox8.Location.Y + 1);

pictureBox9.Location = new Point(pictureBox9.Location.X, pictureBox9.Location.Y + 1);

pictureBox15.Location = new Point(pictureBox15.Location.X, pictureBox15.Location.Y + 1);

pictureBox10.Location = new Point(pictureBox10.Location.X, pictureBox10.Location.Y + 1);

pictureBox12.Location = new Point(pictureBox12.Location.X, pictureBox12.Location.Y + 1);

pictureBox11.Location = new Point(pictureBox11.Location.X, pictureBox11.Location.Y + 1);

pictureBox20.Location = new Point(pictureBox8.Location.X, pictureBox8.Location.Y + 1);

pictureBox21.Location = new Point(pictureBox21.Location.X, pictureBox21.Location.Y + 1);

pictureBox22.Location = new Point(pictureBox22.Location.X, pictureBox22.Location.Y + 1);

pictureBox23.Location = new Point(pictureBox23.Location.X, pictureBox23.Location.Y + 1);

pictureBox24.Location = new Point(pictureBox24.Location.X, pictureBox24.Location.Y + 1);

pictureBox25.Location = new Point(pictureBox25.Location.X, pictureBox25.Location.Y + 1);

pictureBox26.Location = new Point(pictureBox26.Location.X, pictureBox26.Location.Y + 1);

pictureBox27.Location = new Point(pictureBox27.Location.X, pictureBox27.Location.Y + 1);

pictureBox28.Location = new Point(pictureBox28.Location.X, pictureBox28.Location.Y + 1);

pictureBox29.Location = new Point(pictureBox29.Location.X, pictureBox29.Location.Y + 1);

pictureBox30.Location = new Point(pictureBox30.Location.X, pictureBox30.Location.Y + 1);

pictureBox35.Location = new Point(pictureBox35.Location.X, pictureBox35.Location.Y + 1);

pictureBox34.Location = new Point(pictureBox34.Location.X, pictureBox34.Location.Y + 1);

pictureBox33.Location = new Point(pictureBox33.Location.X, pictureBox33.Location.Y + 1);

pictureBox32.Location = new Point(pictureBox32.Location.X, pictureBox32.Location.Y + 1);

pictureBox31.Location = new Point(pictureBox31.Location.X, pictureBox31.Location.Y + 1);

if (pictureBox4.Location.Y==381)

{

pictureBox1.Location = new Point(75, 49);

pictureBox2.Location = new Point(525, 9);

pictureBox3.Location = new Point(335, 84);

pictureBox4.Location = new Point(600, j);

pictureBox5.Location = new Point(425, 34);

pictureBox6.Location = new Point(116, 49);

pictureBox7.Location = new Point(710, 71);

pictureBox8.Location = new Point(666, 29);

pictureBox9.Location = new Point(824, 71);

pictureBox15.Location = new Point(801, 12);

pictureBox10.Location = new Point(195, 43);

pictureBox12.Location = new Point(376, 34);

pictureBox11.Location = new Point(759, 29);

pictureBox20.Location = new Point(580, 25);

pictureBox21.Location = new Point(12, 12 );

pictureBox22.Location = new Point(906, 57);

pictureBox23.Location = new Point(500, -10);

pictureBox24.Location = new Point(269, 74);

pictureBox25.Location = new Point(91, -10);

pictureBox26.Location = new Point(143, 34);

pictureBox27.Location = new Point(306, 36);

pictureBox28.Location = new Point(465, 71);

pictureBox29.Location = new Point(837, 12);

pictureBox30.Location = new Point(837, 12);

pictureBox35.Location = new Point(54, -1);

pictureBox34.Location = new Point(251, 12);

pictureBox33.Location = new Point(408, 71);

pictureBox32.Location = new Point(555, -1);

pictureBox31.Location = new Point(626, 9);

}

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void pictureBox13\_Click(object sender, EventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void pictureBox7\_Click(object sender, EventArgs e)

{

}

private void pictureBox15\_Click(object sender, EventArgs e)

{

}

private void pictureBox11\_Click(object sender, EventArgs e)

{

}

private void pictureBox18\_Click\_1(object sender, EventArgs e)

{

}

private void pictureBox17\_Click(object sender, EventArgs e)

{

}

private void pictureBox6\_Click(object sender, EventArgs e)

{

}

private void pictureBox9\_Click(object sender, EventArgs e)

{

}

private void pictureBox26\_Click(object sender, EventArgs e)

{

}

private void pictureBox10\_Click(object sender, EventArgs e)

{

}

private void pictureBox22\_Click(object sender, EventArgs e)

{

}

private void pictureBox35\_Click(object sender, EventArgs e)

{

}

private void pictureBox23\_Click(object sender, EventArgs e)

{

}

}

}

**OUTPUT**



Q.3 Create an application that can display moving car horizontally and vertically on track bar using C# code.

**PROGRAM**

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 car\_track\_bar\_csharp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int i = 1;

int j = 313;

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

{

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

}

private void trackBar3\_Scroll(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (i < 719 && j == 313)

{

pictureBox1.Location = new Point(i, 312);

i++;

}

else if (i == 719 && j > 2)

{

pictureBox1.Location = new Point(i, j);

j--;

}

else if (i > -4 && j == 2)

{

pictureBox1.Location = new Point(i, 2);

i--;

}

else if (i == -4 && j >= 2)

{

pictureBox1.Location = new Point(i, j);

j++;

}

else if (i == -4 && j == 293)

{

pictureBox1.Location = new Point(i, j);

j++;

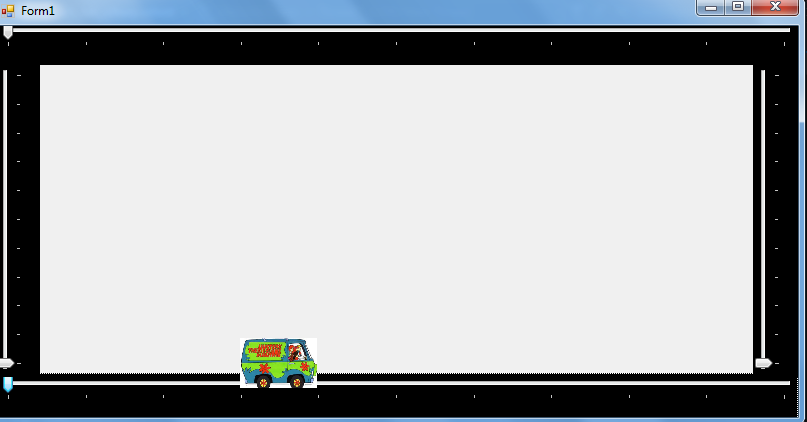
}

}

}

}

**OUTPUT**





Q.4. Create an application in C sharp that will place different objects at random locations over the form and they are moving with random speed and would be killed by

some other object. Application must be able to count the score that has been produced by the player.

**PROGRAM**

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 enemy\_game\_csharp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

int score = 0;

int x = 76;

int y = 90;

int i;

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Right)

{

i = 10;

}

if (e.KeyCode == Keys.Left)

{

i = 15;

}

if (e.KeyCode == Keys.Up)

{

i = 20;

}

if (e.KeyCode == Keys.Down)

{

i = 25;

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (i == 10)

{

x=x+10;

pictureBox5.Location = new Point(x, y);

}

if (i == 15)

{

x = x - 10;

pictureBox5.Location = new Point(x, y);

}

if (i == 20)

{

y = y - 10;

pictureBox5.Location = new Point(x, y);

}

if (i == 25)

{

y = y +10;

pictureBox5.Location = new Point(x, y);

}

if(pictureBox5.Bounds.IntersectsWith(pictureBox1.Bounds)&&(pictureBox1.Visible))

{

pictureBox1.Hide();

score += 1;

label1.Text = "Score is" + score;

}

if (pictureBox5.Bounds.IntersectsWith(pictureBox2.Bounds) && (pictureBox2.Visible))

{

pictureBox2.Hide();

score += 1;

label1.Text = "Score is" + score;

}

if (pictureBox5.Bounds.IntersectsWith(pictureBox3.Bounds) && (pictureBox3.Visible))

{

pictureBox3.Hide();

score += 1;

label1.Text = "Score is" + score;

}

if (pictureBox5.Bounds.IntersectsWith(pictureBox4.Bounds) && (pictureBox4.Visible))

{

pictureBox4.Hide();

score += 1;

label1.Text = "Score is" + score;

}

}

private void timer2\_Tick(object sender, EventArgs e)

{

Random ran = new Random();

int j = ran.Next(300, 900);

int k = ran.Next(-2, 272);

int z = ran.Next(-2, 272);

int m = ran.Next(300, 900);

int n = ran.Next(300, 900);

int o = ran.Next(300, 900);

int u = ran.Next(-2, 272);

int v = ran.Next(-2, 272);

pictureBox1.Location = new Point(j, k);

pictureBox2.Location = new Point(m, z);

pictureBox3.Location = new Point(n, u);

pictureBox4.Location = new Point(o, v);

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

}

}

}

**OUTPUT**

