Student Name
Robert Collcott

Student ID
21302939

Lecturer
Richard Jones

Session Time and Day
1PM to 4PM on Thursdays

Table of Contents

C# Console

Remaining Bits

Exercise 5.4 Body Mass

I had to make a program that was capable of defining people's weight

I had to break it down into the following sections these were the following

- BMI less than 18.5 the program will display underweight
- BMI 18.5 up to 25 the program will output desirable weight for size
- BMI 25 to 30 the program will output the person is overweight
- BMI 30 to 40 the program will output the person is Obese
- BMI 40+ the program will output the person is Severely Obese

Algorithm

Declare variables

Add a new method called get details

Get User input for entering weight

Get User input for entering height

Add another method called calcBMI

Return a value

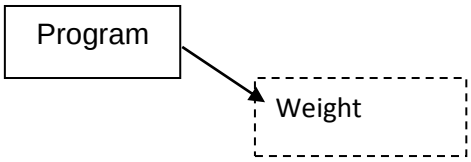
Display a message saying to the user which weight category the fall into

Source Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _5._4_BodyMass
{
    class BMI
    {
        static void Main(string[] args)
        {
            float weight;                                       // Weight has to be a
float varaiable as it weight can have decimal points in e.g. 19.24
            string input, name;

            Console.Write("Please enter person name");          // Get user input for
the person's name
            name = Console.ReadLine();



            Console.Write("Please enter weight");               // Get user input for
the person's weight
            input = Console.ReadLine();
            weight = Convert.ToInt32(input);


            if (weight <= 18.5)
            {


                Console.Write("You are under weight");          // Message will print
here to tell the person is under weight



            }


            if (weight <26)
            {

                Console.Write("You are desirable weight for size");  // Message will
print here to tell the person is desirable weight


                input = Console.ReadLine();
            }


            if (weight == 30)                                    // Message will print
here to tell the person is ober weight
            {


                Console.Write("You are overweight");


                input = Console.ReadLine();
            }


            if (weight == 40)
```

```
            {

                Console.Write("You are obse");      // Message will print here to tell
the person is obse

                input = Console.ReadLine();
            }

            if (weight >= 41)
            {

                Console.Write("You are severely obse ---- LOOSE SOME WEIGHT"); //
Message will print here to tell the person is severely obse

                input = Console.ReadLine();
            }

        }
    }
}
```

Input-Output Diagram
showing the inputs and outputs for the program



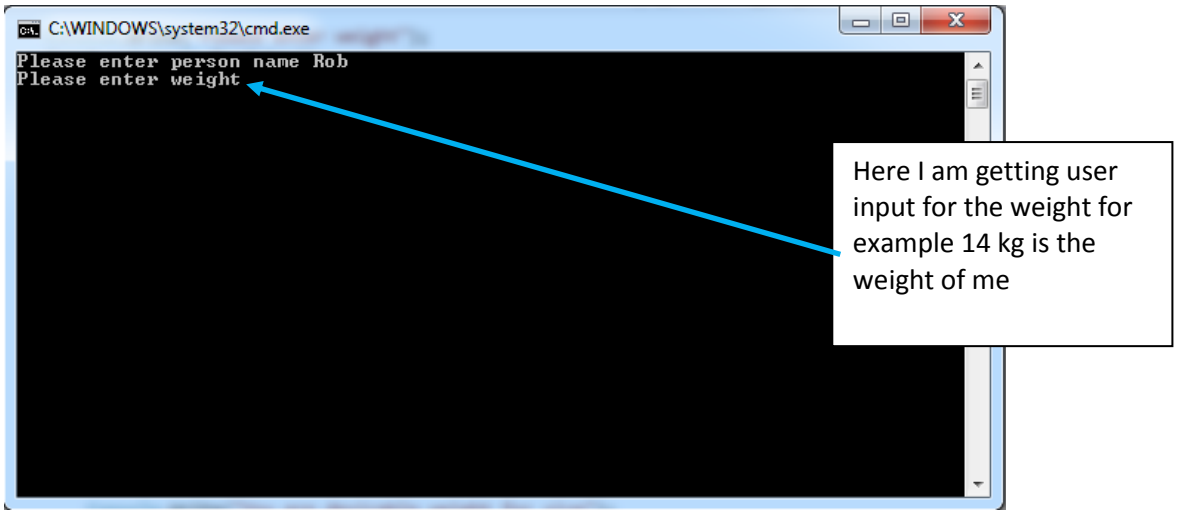
Identifier List
the variables to be used in the program

Identifier	Data Type	Meaning
Weight	Double	The weight of the person

Test Plan
actual results are left blank at this stage and filled in after the program has been written.

Test No.	Inputted weight value	Expected Outputs	Actual Outputs
1	14	Underweight	Underweight
2	55	Severely Obese	Severely Obese
3	30	Overweight	Overweight
4	22	Desirable weight for size	Desirable weight for size
5	61	Severely Obese	Severely Obese

Evidence of Body Mass sample program running [Sample Outputs]



So if someone was only weighing 14 the program would output the certain suspect was underweight

Week 2 -- Independent Study 1 -- Traffic Survey 6.4

Algorithm

Define variables

Create integer array

Define 1st method called enterCounts

Define 2nd method named show total

Define 3rd method named busiest

Define 4th method named show data

Add the final method named report that outputs the following data

- Hour
- Car Counts
- Total
- Busiest hour

Had a go trying to code the program but could not get it to work

Algorithm Logic

Declare variables

Declare functions

- Add a Booking
- Vacate a room
- Display details of All rooms
- Vacate all rooms – clear system
- Quit the program all together

Declare an integer array of 20 rooms

Get user input for room details

- Room Number
- How many guests

Output details on the screen

Source Code

On Page 10

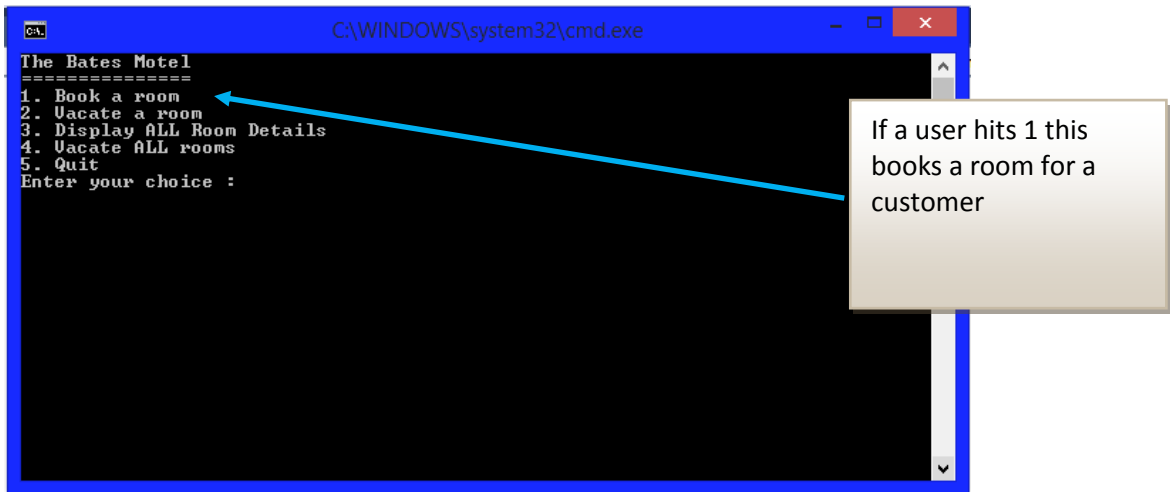
```
using System;

namespace Task7_2
{

    // author: Robert Collcott
    // ID 21302939
    // Computing
    // 22nd January 2015

    class Motel
    {
        int[] rooms;
        const int MAX = 20;                     // Creation of the aray of rooms

        static void Main()
        {
            Motel BatesMotel = new Motel();
            BatesMotel.runMotel();
        }
        //*********************************************************
        public Motel()
        {
            rooms = new int[MAX + 1];        // allow rooms from 1 to MAX which is 20
        }
        //*********************************************************
        public void runMotel()              // This is the menu screen for the motel
booking system
        {
            string choice = "";
            do
            {
                Console.Clear();
                Console.WriteLine("The Bates Motel");
                Console.WriteLine("===============");
                Console.WriteLine("1. Book a room");      // Hit 1 for booking rooms
                Console.WriteLine("2. Vacate a room");     // Hit 2 to vacate a room
                Console.WriteLine("3. Display ALL Room Details"); // Hit 3 to display
room details
                Console.WriteLine("4. Vacate ALL rooms"); // Hit 4 to clear room data
                Console.WriteLine("5. Quit program");  // Hit 5 to exit the program
all together
                Console.Write("Enter your option : ");
                choice = Console.ReadLine();
                if (choice == "1")
                {
                    bookRoom();
                }
            }
            while (choice != "5");
        }
        //*********************************************************
        public void bookRoom()
        {
            int roomNumber, guests;
            Console.WriteLine("\nThe Bates Motel");  // Menu Heading
            Console.WriteLine("===============");
            Console.WriteLine("Book a room");
            Console.Write("Enter the room number : ");   // Input for the user to type
the room number
            roomNumber = Convert.ToInt32(Console.ReadLine());
            Console.Write("How many guests : ");      // Input for the user to type
how many guests are staying in the room
            guests = Convert.ToInt32(Console.ReadLine());
```

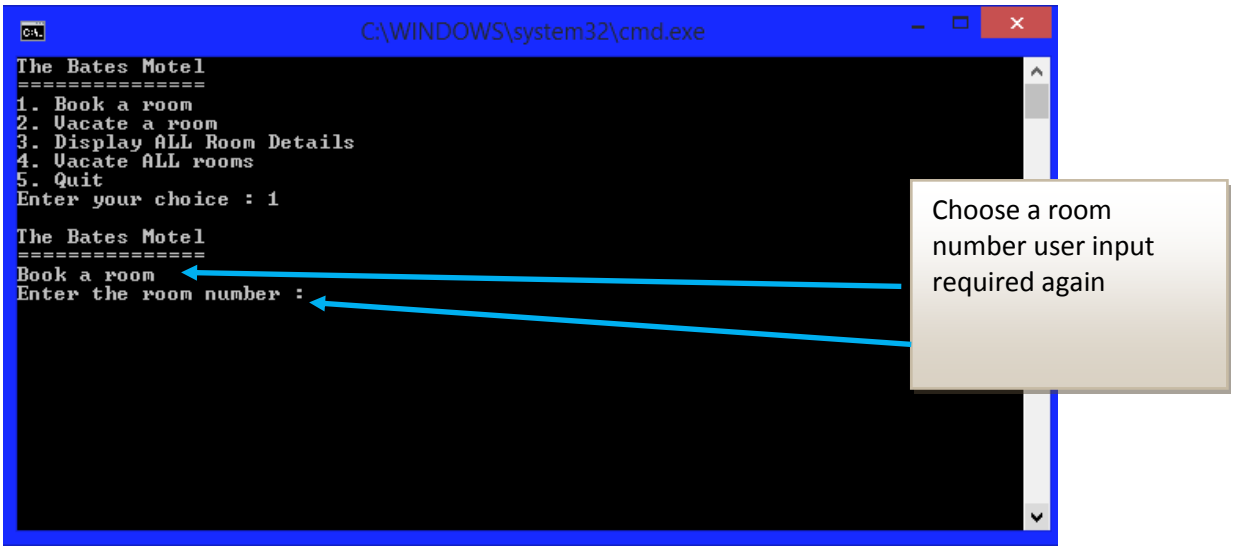
```
            rooms[roomNumber] = guests;                             // make the
booking
            Console.WriteLine("Room " + roomNumber + " booked for " + guests + "
people");
            Console.ReadKey();      // Read the information
        }
        //*****************************************************

        public void showAllRooms()
        {
            Console.WriteLine("Bates Motel Room Status");
            Console.WriteLine("======================");
            Console.WriteLine("Room " + rooms[roomNumber] + "\t" + guests + "
guests");
        }

        public void VacateAllRooms()
        {
            Console.WriteLine("Display ALL Room Details");
            Console.WriteLine("");

        }


    }
}
```

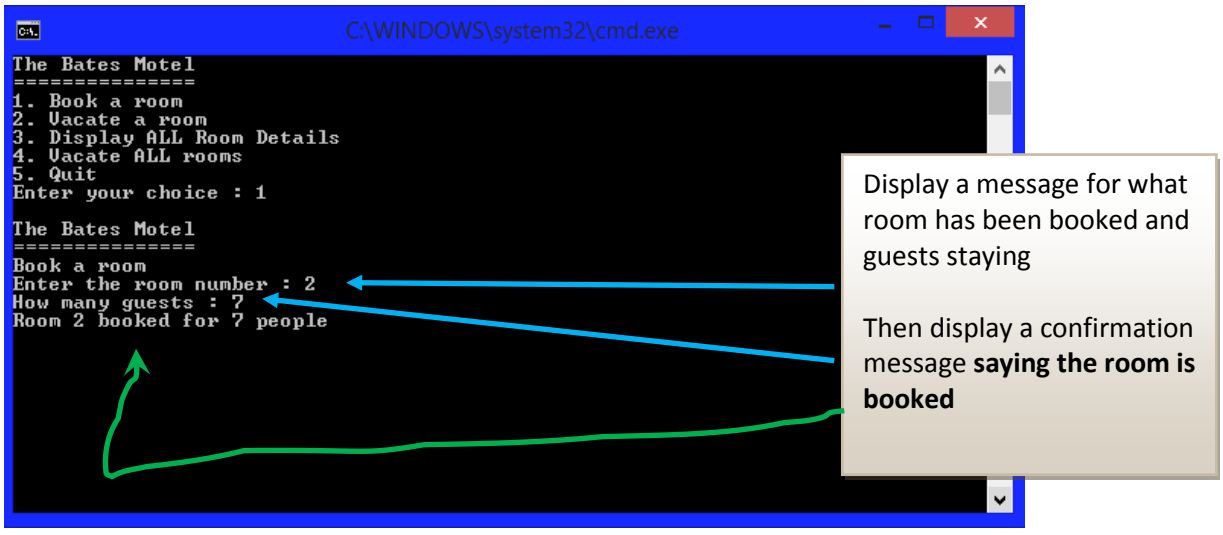
As much as I tried to get the other functions to work such as display all room details I could not get them to work

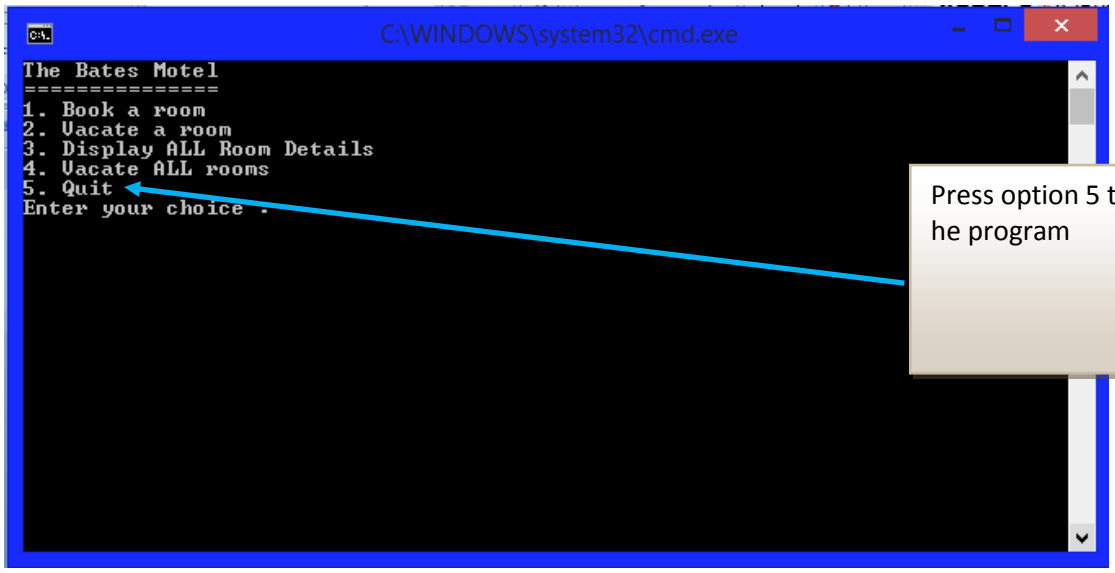
I googled help forums such as stack overflow these did not help what so ever

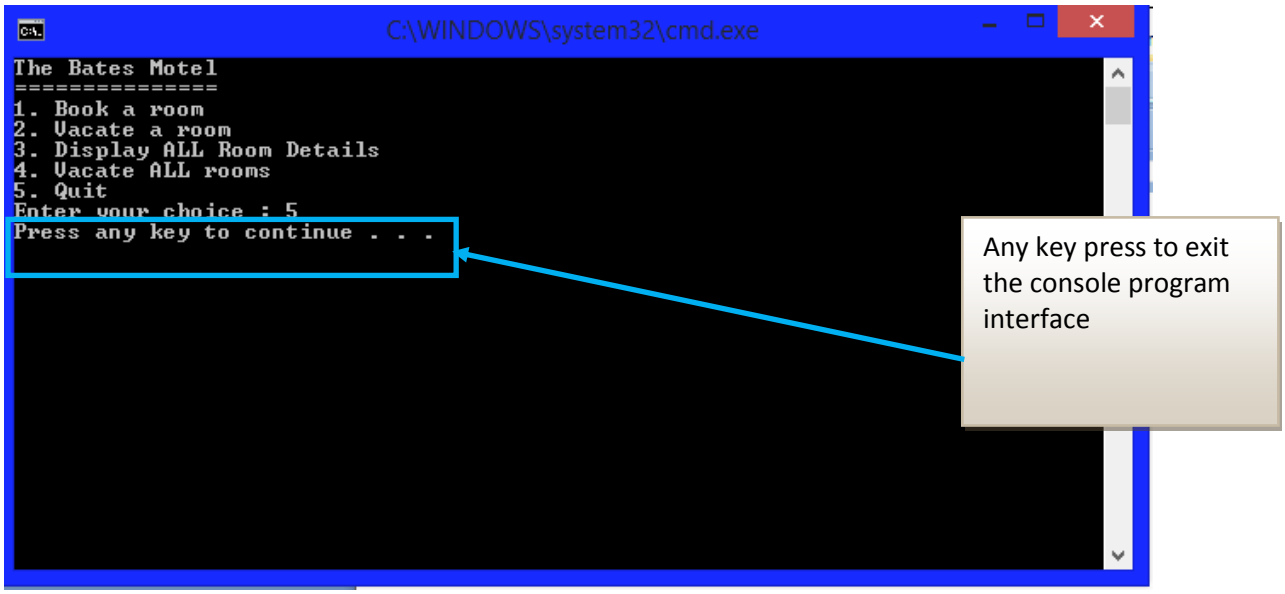
Sample Bates Motel program outputs

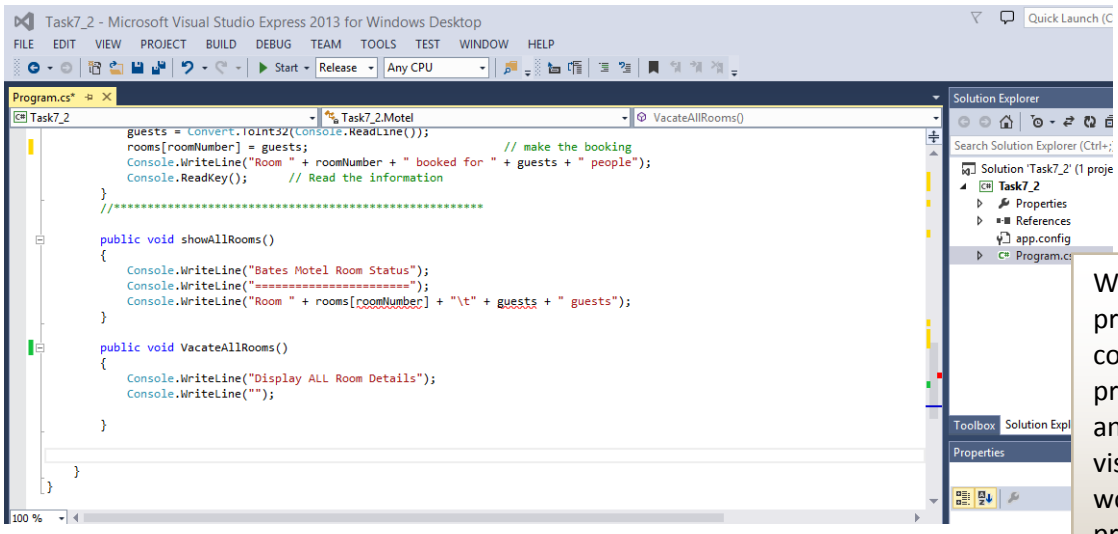







C:\WINDOWS\system32\cmd.exe
The Bates Motel
==========
1. Book a room
2. Vacate a room
3. Display ALL Room Details
4. Vacate ALL rooms
5. Quit
Enter your choice :
Press option 5 to quit he program


C:\WINDOWS\system32\cmd.exe
The Bates Motel
==========
1. Book a room
2. Vacate a room
3. Display ALL Room Details
4. Vacate ALL rooms
5. Quit
Enter your choice : 5
Press any key to continue . . .
Any key press to exit the console program interface


Task7_2 - Microsoft Visual Studio Express 2013 for Windows Desktop
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Program.cs*
Task7_2
Task7_2.Motel
VacateAllRooms()
guests = Convert.ToInt32(Console.ReadLine());
rooms[roomNumber] = guests; // make the booking
Console.WriteLine("Room " + roomNumber + " booked for " + guests + " people");
Console.ReadKey(); // Read the information
}
//***
public void showAllRooms()
{
Console.WriteLine("Bates Motel Room Status");
Console.WriteLine("======================");
Console.WriteLine("Room " + rooms[roomNumber] + "\t" + guests + " guests");
}
public void VacateAllRooms()
{
Console.WriteLine("Display ALL Room Details");
Console.WriteLine("");
}
}
}
Solution Explorer
Task7_2
Properties
References
app.config
Program.cs
Properties
When you quit the program it will shut the command line console program interface down and display in this case my visual studio as I was working on another program at the time of testing the motel program

My C# Console project has to include the following elements

1. My Source Code **(fully commented)**
2. Author name, date, project title etc.
3. Sample outputs from program execution
4. Class Diagrams
5. Completed Test Plans
6. Commentary on success (or otherwise)

Algorithm Logic

Declare variables

Define classes and methods

Define objects

Begin a loop that repeats 20 times to play the game

Get user input for the player's choice

Output the computer's choice by using the random class

Display result

IF

Player picks the same thing as computer result is a draw
Output message saying
IT'S A DRAW

IF

Player picks Scissors and computer picks paper

Player wins because Scissors cut paper

IF

Computer picks Scissors computer looses blunts Scissors

IF

Player picks paper and computer picks stone

Player wins because paper wraps stone

Output scores after each round

Output final score for player

Output final score for computer

Output final score difference

Close program

This is where the main variables are declared this helps the program have more flexibility

Sample Outputs

Here I am getting user input for the name of the player

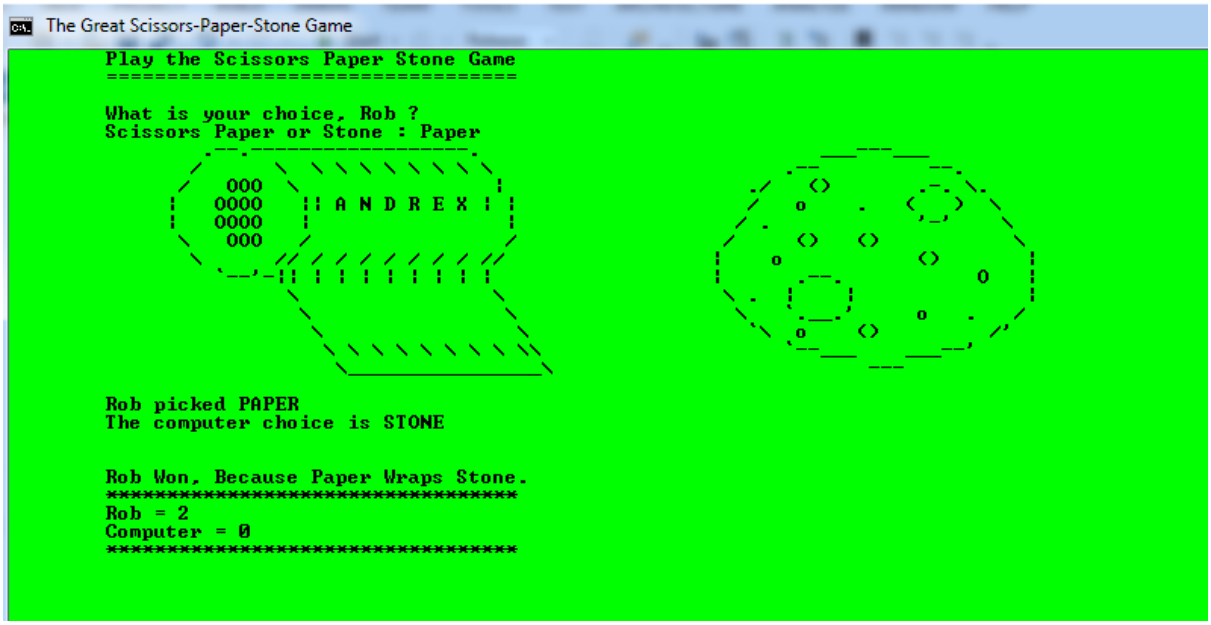
Here I inputted the name of the player now I need to get the user input of what the player wants to choose from either

- Scissors
- Paper
- Stone

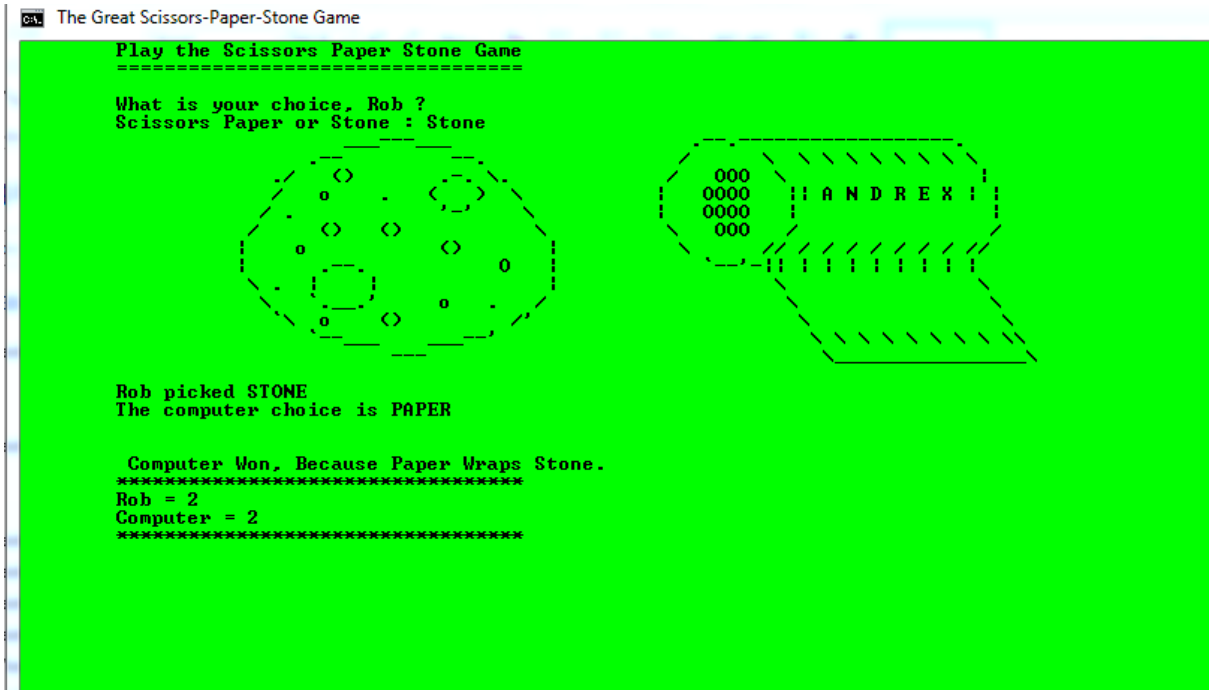
Here I am getting the message to say when I won the game, drew or lost the game

Here I am getting user input by what they want to choose to play with

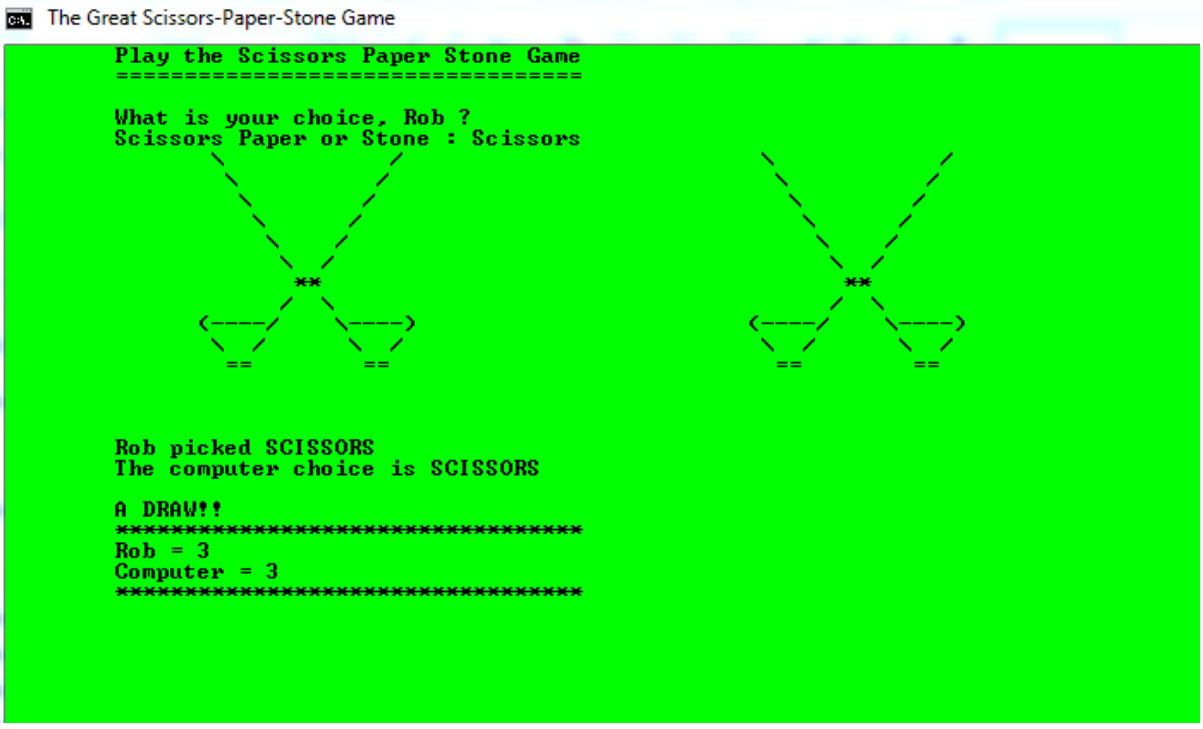
- Scissors
- Paper
- Stone



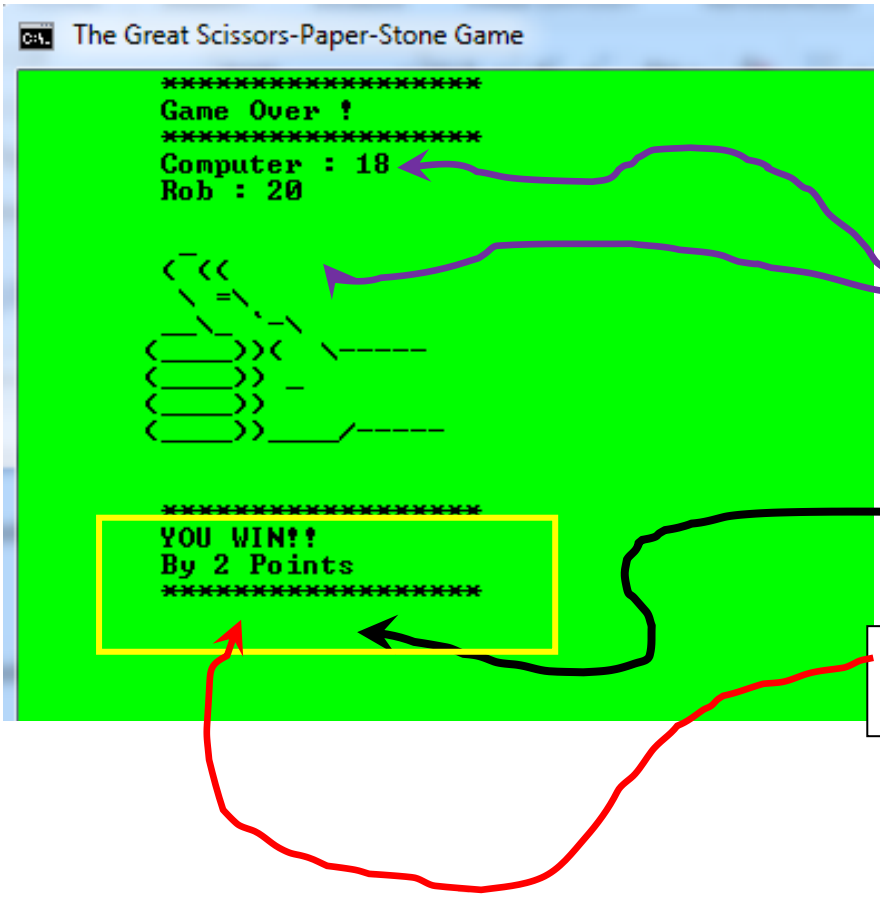
With the computer program having had a random object declared in it when I choose the computer picks at random so if I picked Paper the computer could even pick paper to draw a game stone or scissors



I picked stone here and the computer picked paper at random the Computer won here because paper wraps stone



Here the computer choice if it matches what I pick the program will give me and the computer 1 point each and output the following message **A DRAW!!!!**



When either me or the computer reaches 20 points

The game is closed off and I programmed a message to display the result draw win or loss and by how many points so users will know where to improve for next time they plan the game

Score difference

Source Code for project

```
using System;

namespace SPSProject
{
                    // Robert Collcott
                    // ID 21302939
                    // Computing
                    // 12th February 2015


    class Game
    {
        string compChoice;
        string playerName, playerChoice;
        int p, c, r; // To store Player's (p), Computer's (c) scores and Result (r).
        Random randy;

        static void Main()
        {
            Game myGame = new Game();           // create a new Game object
            myGame.play();                       // call its play method
        }
        //*******************************************************

        public Game()                           // Create game class
        {
            randy = new Random();               // define a random object
        }
        //*******************************************************
        public void play()
        {
            setupScreen();
            introduction();
            getPlayerName() // get user input for the name of the player playing the
game
            Console.Clear();
            while(true)        // Start a loop       // p is for player and c is for
computer
            {
                if (p >= 20 || c >= 20)                      // To stop the game when
one of the player reaches 20.
                {
                    break;
                }
                else
                {
                    introduction(); // To keep the intro on screen every time the game
repeats.
                    getPlayerChoice();
                    getComputerChoice();
                    drawPlayerChoice();
                    drawComputerChoice();
                    printChoices();
                    showResult();
                    showScore();
                    Console.ReadKey();   // wait for a key press
                    Console.Clear();     // Clear the screen
                }
```

```
            }
            finish();
        }
        //*********************************************************
        private void setupScreen()
        {
            Console.Title = " The Great Scissors-Paper-Stone Game";                  //
This is the console window title
            Console.SetWindowSize(150, 60);
            Console.SetBufferSize(150, 60);
            Console.BackgroundColor = ConsoleColor.Green;
            Console.ForegroundColor = ConsoleColor.Black;
            Console.Clear();  // clear screen in chosen colour
        }
        //*********************************************************
        private void introduction()
        {
            Console.WriteLine("\tPlay the Scissors Paper Stone Game");
       // Message to welcome the users of the game
            Console.WriteLine("\t==================================");
        }
        //*********************************************************
        private void getPlayerName()                                                //
This is the get player name function
        {
            Console.Write("\n\tPlease Enter Your Name:   ");
            playerName = Console.ReadLine();
        }
        //*********************************************************
        private void getPlayerChoice()                                              //
This is the get player choice function
        {
            Console.WriteLine("\n\tWhat is your choice, " + playerName+" ?");
            Console.Write("\tScissors Paper or Stone : ");
            playerChoice = Console.ReadLine();
            playerChoice = playerChoice.ToUpper();
        }
        //*********************************************************
        private void getComputerChoice()
                            // Generate the computer choice
        {
            int num;
            num = randy.Next(3);  // pick a random number (0, 1 or 2)
            if (num == 1)
            {
                compChoice = "SCISSORS";
            }
            else if (num == 2)
            {
                compChoice = "STONE";
            }
            else if (num == 0)
            {
                compChoice = "PAPER";
            }
            else
            {
                compChoice = "NOT YET DETERMINED";
            }
        }
        //*********************************************************
        private void printChoices()
        {
            Console.WriteLine("\n\t"+playerName+" picked " + playerChoice);
            Console.WriteLine("\tThe computer choice is " + compChoice);
```

```
        }
        //************************************************
        private void showResult()
                        // Shows the result
        {
            if (playerChoice == compChoice)
                        // Player choice matches the computers choice
            {
                Console.WriteLine("\n\tA DRAW!!");
                     // Display a message saying ITS A DRAW
                c++;
                p++;
            }
            else if (playerChoice == "PAPER" && compChoice == "STONE")
       // if the choice is a players with paper and the computer choice is stone the
player wins because paper wraps stone
            {
                 Console.WriteLine("\n\n\t"+playerName+" Won, Because Paper Wraps
Stone.");
                 p+=2;
            }
            else if (playerChoice == "PAPER" && compChoice == "SCISSORS")
       // if the choice is a players with paper and the computer choice is scissors
the computer wins because scissors cut paper
            {
                 Console.WriteLine("\n\n\t Computer Won, Because Scissors Cut
Paper.");
                 c+=2;
            }
            else if (playerChoice == "SCISSORS" && compChoice == "PAPER")
       // if the choice is a players with paper and the computer choice is scissors
the computer wins because scissors cut paper
            {
                Console.WriteLine("\n\n\t" + playerName + " Won, Because Scissors Cut
Paper.");
                p+=2;
            }
            else if (playerChoice == "SCISSORS" && compChoice == "STONE")
            {
                Console.WriteLine("\n\n\t Computer Won, Because Stone Blunts
Scissors.");
                c+=2;
            }
            else if (playerChoice == "STONE" && compChoice == "PAPER")
            {
                Console.WriteLine("\n\n\t Computer Won, Because Paper Wraps Stone.");
                c+=2;
            }
            else if (playerChoice == "STONE" && compChoice == "SCISSORS")
            {
                Console.WriteLine("\n\n\t" + playerName + " Won, Because Stone Blunts
Scissors.");
                p+=2;
            }
          }
        //*******************************************************
             // This is the draw player choice function this draws the players
choices
        private void drawPlayerChoice()
        {
            if (playerChoice == "SCISSORS")
            {
                drawScissors(10, 5);
                        // Image position to place the scissors image on the
screen
            }
```

```
            else if (playerChoice == "PAPER")
            {
                drawPaper(10, 5);
                            // Image position to place the paper image on the screen
            }
            else if (playerChoice == "STONE")
            {
                drawStone(10, 5);
                            // Image position to place the paper image on the screen
            }
        }
        //*********************************************************************
        private void drawComputerChoice()
        {
            if (compChoice == "SCISSORS")
                     // draw the stone image if the computer picked Scissors
            {
                drawScissors(50, 5);
            }
            else if (compChoice == "PAPER")
                            // draw the stone image if the computer picked Paper
            {
                drawPaper(50, 5);
            }
            else if (compChoice == "STONE")
                            // draw the stone image if the computer picked stone
            {
                drawStone(50, 5);
            }
        }

        //*********************************************************************
        private void showScore()
                                        // show the score display
        {
            Console.WriteLine("\t**********************************");
            Console.WriteLine("\t"+playerName+" = "+p);
            Console.WriteLine("\tComputer = " + c);
            Console.WriteLine("\t**********************************");

        }
        //*********************************************************************
        private void finish()
                                              // display and finish the game
        {
            Console.Clear();
                                              // Clear the screen
            Console.WriteLine("\t******************");
            Console.WriteLine("\tGame Over !");
                                    // Display a message saying game over and either
the computer wins or you as the player wins
            Console.WriteLine("\t******************");
            Console.WriteLine("\tComputer : " + c);
            Console.WriteLine("\t" + playerName + " : " + p);
            if (p > c)
            {
                r = p - c;
                drawThumbsUp();
                Console.WriteLine();
                Console.WriteLine("\t******************");
                Console.WriteLine("\tYOU WIN!!");
                Console.WriteLine("\tBy " + r + " Points");
                     // This tells the player how points they won by
                Console.WriteLine("\t******************");
            }
            else if (p < c)
```

```
            {
                r = c - p;
                drawThumbsDown();
                Console.WriteLine();
                Console.WriteLine("\t******************");
                Console.WriteLine("\tCOMPUTER WINS!!");
                Console.WriteLine("\tBy " + r + " Points");
              // This tells the player how points they lost by
                Console.WriteLine("\t******************");
            }
            else if (p == c)
            {
                drawSmile();
                Console.WriteLine();
                Console.WriteLine("\tGame Drawn! Better Luck Next Time.");         //
This tells the player how points they got and the computers points to output a message
saying the match has been drawn
            }
            Console.ReadKey();

        }
        //*********************************************************************
        private void drawScissors(int x, int y)

                        // draw scissors image
        {
            Console.SetCursorPosition(x, y++);   // set start position then increment
y to move down
            Console.Write("     \\            /");
            Console.SetCursorPosition(x, y++);
            Console.Write("      \\          /");
            Console.SetCursorPosition(x, y++);
            Console.Write("       \\        /");// Write and print out the scissors
pieces of the scissors image
            Console.SetCursorPosition(x, y++);
            Console.Write("        \\      /");
            Console.SetCursorPosition(x, y++);
            Console.Write("         \\    /");
            Console.SetCursorPosition(x, y++);
            Console.Write("          \\  /");
            Console.SetCursorPosition(x, y++);
            Console.Write("           **");
            Console.SetCursorPosition(x, y++);
            Console.Write("          /  \\");
            Console.SetCursorPosition(x, y++);
            Console.Write("    (----/    \\----)");
            Console.SetCursorPosition(x, y++);
            Console.Write("     \\  /      \\  /");
            Console.SetCursorPosition(x, y++);
            Console.Write("      ==        ==");
            Console.WriteLine("\n\n");
        }
        //*************************************************************
        private void drawStone(int x, int y)

                // draw stone image
        {
            Console.SetCursorPosition(x, y++);   // set start position then increment
y to move down
            Console.Write("                  ___---___      ");
            Console.SetCursorPosition(x, y++);
            Console.Write("              .--         --.    ");
            Console.SetCursorPosition(x, y++);
            Console.Write("           ./   ()       .-. \\.   ");
            Console.SetCursorPosition(x, y++);
            Console.Write("          /   o    .   (   )  \\  ");
```

```
            Console.SetCursorPosition(x, y++);
            Console.Write("          / .             '-'    \\  ");
            Console.SetCursorPosition(x, y++);
            Console.Write("         /     ()   ()           \\ ");
            Console.SetCursorPosition(x, y++);
            Console.Write("        |    o           ()       | ");
            Console.SetCursorPosition(x, y++);
            Console.Write("        |      .--.           o   | ");
            Console.SetCursorPosition(x, y++);
            Console.Write("         \\ .  |    |              |  ");
            Console.SetCursorPosition(x, y++);
            Console.Write("          \\    `.__.'     o   .   /    ");
            Console.SetCursorPosition(x, y++);
            Console.Write("           `\\  o    ()         /'    ");
            Console.SetCursorPosition(x, y++);
            Console.Write("             `--___    ___--'    ");
            Console.SetCursorPosition(x, y++);
            Console.Write("                   ---         ");
            Console.WriteLine();
        }
        //**************************************************************************
        private void drawPaper(int x, int y)

                    // draw paper image
        {
            Console.SetCursorPosition(x, y++);    // set start position then increment
y to move down
            Console.Write("      .--.------------------.");
            Console.SetCursorPosition(x, y++);
            Console.Write("     /      \\  \\ \\ \\ \\ \\ \\ \\ \\ \\");
            Console.SetCursorPosition(x, y++);
            Console.Write("    /   OOO  \\                |");
            Console.SetCursorPosition(x, y++);
            Console.Write("   |   OOOO   || A N D R E X | |");
            Console.SetCursorPosition(x, y++);
            Console.Write("   |   OOOO   |                |");
            Console.SetCursorPosition(x, y++);
            Console.Write("    \\   OOO   /                /");
            Console.SetCursorPosition(x, y++);
            Console.Write("     \\      // / / / / / / //");
            Console.SetCursorPosition(x, y++);
            Console.Write("       `--'-|| | | | | | | |");
            Console.SetCursorPosition(x, y++);
            Console.Write("             \\\                \\\");
            Console.SetCursorPosition(x, y++);
            Console.Write("              \\\                \\\");
            Console.SetCursorPosition(x, y++);
            Console.Write("               \\\                \\\");
            Console.SetCursorPosition(x, y++);
            Console.Write("                \\\ \\\ \\\ \\\ \\\ \\\ \\\ \\\ \\\\");
            Console.SetCursorPosition(x, y++);
            Console.Write("                 \\\_________________\\\");
            Console.WriteLine();
        }
        //**************************************************************************
        private void drawSmile()
                                // draw a smiley if you as the player wins the game
        {
            Console.WriteLine("\n                    .-\"\"\"\"\"-.-\"\"\"\"\"-. ");
            Console.WriteLine("                  _.`                 `'._   ");
            Console.WriteLine("              .-'  __..,.___.___.,..__  '-.    ");
            Console.WriteLine("            '-.-;` |  |    |    |  | `;-.-'    ");
            Console.WriteLine("             \\'-\\_/\\__|    |    |__/\\_/-'/   ");
            Console.WriteLine("              \\, _      '---'---'     _ ,/    ");
            Console.WriteLine("               \\'./`'.--.--.--,--.'`\\.'/    ");
            Console.WriteLine("                \\ `'-;__|__|__|__;-'` /    ");
```

```
            Console.WriteLine("                '.                 .'   ");
            Console.WriteLine("                 `'-....---....-'`    ");
        }
        //************************************
        private void drawThumbsUp()
                                        // draw a thumbs up to show you won the game
        {
            Console.WriteLine();
            Console.WriteLine("          _ ");
            Console.WriteLine("        ( ((  ");
            Console.WriteLine("         \\ =\\   ");
            Console.WriteLine("        __\\_ `-\\   ");
            Console.WriteLine("       (____))(  \\-----  ");
            Console.WriteLine("       (____)) _    ");
            Console.WriteLine("       (____))   ");
            Console.WriteLine("       (____))____/-----  ");
            Console.WriteLine();
        }
        //************************************
        private void drawThumbsDown()
                            // draw a thumbs down image if you as the player looses
        {
            Console.WriteLine();
            Console.WriteLine("          ______ ");
            Console.WriteLine("       ((____  \\-----  ");
            Console.WriteLine("       ((_____        ");
            Console.WriteLine("       ((_____      ");
            Console.WriteLine("       ((____   -----   ");
            Console.WriteLine("           /  /    ");
            Console.WriteLine("          (_((     ");
            Console.WriteLine();
        }
    }
}
```

C# Console Project Test Plan				
Test No	Description of test	Expected Result	Actual Result	Additional Comments
1	**User input of player = Robert**	**Robert should be the player asked**	**Robert should was the player asked**	The Console application stored my name ready to ask for scissors paper or stone
2	**When the computer or I reaches 20 points the game stops**	**Game stops**	**Game stops**	The console screen displayed a message
3	**When the player of the game chooses Scissors and the computer chooses scissors**	**Draw**	**Draw**	There was a message displayed saying ITS A DRAW
4	**When the player of the game chooses stone and the computer chooses paper**	**Computer should win**	**Computer did win**	Computer did win because paper wraps stone
5	**Player picks paper and computer picks stone**	**Player wins because paper wraps stone**	**Player wins because paper wraps stone**	Player did win
6	**Drawings of stone paper and stone are drawn correctly**	**All drawings should come out correctly**	**All drawings came out correctly**	No additional comments were required here as it's just a simple drawing
7	**Displaying a result message when the game is over**	**Result message should display**	**Result message did display**	Result message did display with the score summary e.g. me 20 points the computer got 18

C# Console Project Evaluation [Scissors Paper Stone Game]

I created a project that was based on the scissors paper stone game concept

First to make the program more efficient to use I broke the program up into several functions these were

- Game
- Play
- get player name
- get player choice
- get computer choice
- print choices
- show result
- draw player choice
- draw computer choice
- show score
- finish

All these functions helped me break the program project into manageable chunks if I had the program all in one function problems would happen such as the updates of scores and choices the player and the computer choose at random by creating a random object named randy.

When you see Console.WriteLine this were the pieces of code that wrote the details on the screen such as the player name

Console.Clear(); this would clear the screen each time so the users want to do something such as choose scissors paper or stone to choose to play the game

One of the problems I had with the project was creating a background colour back drop but this was quickly resolved by searching for the correct syntax on MSDN knowledge base site this told me how to code it in C# with the colour code and colour name.

I also had a problem with trying to align the text on the console command prompt that the program would run on as I am programming old console DOS programs here.

This took some time to resolve as I was searching multiple sites some forums such as stack overflow however, I found the correct syntax that worked on the MSDN site in C# format.

Overall I thought the project was very manageable to complete as it had a small basic program and 3 chunks of extension work which some were easy some were hard to complete

Project Rating

8/10

C#
Windows
.NET
PROGRAMMING

C# Windows Week 1 -- Exercise No 5.4 Independent Study 1.5 Super Surfer

It's just a simple form this week that hides shows a pic and the form changes colour

Algorithm Logic

Declare Variables

Add Buttons in for

- Change Form colour to Blue
- Change Form colour to Red
- Change form colour to Yellow

Get user input to hide or show the picture

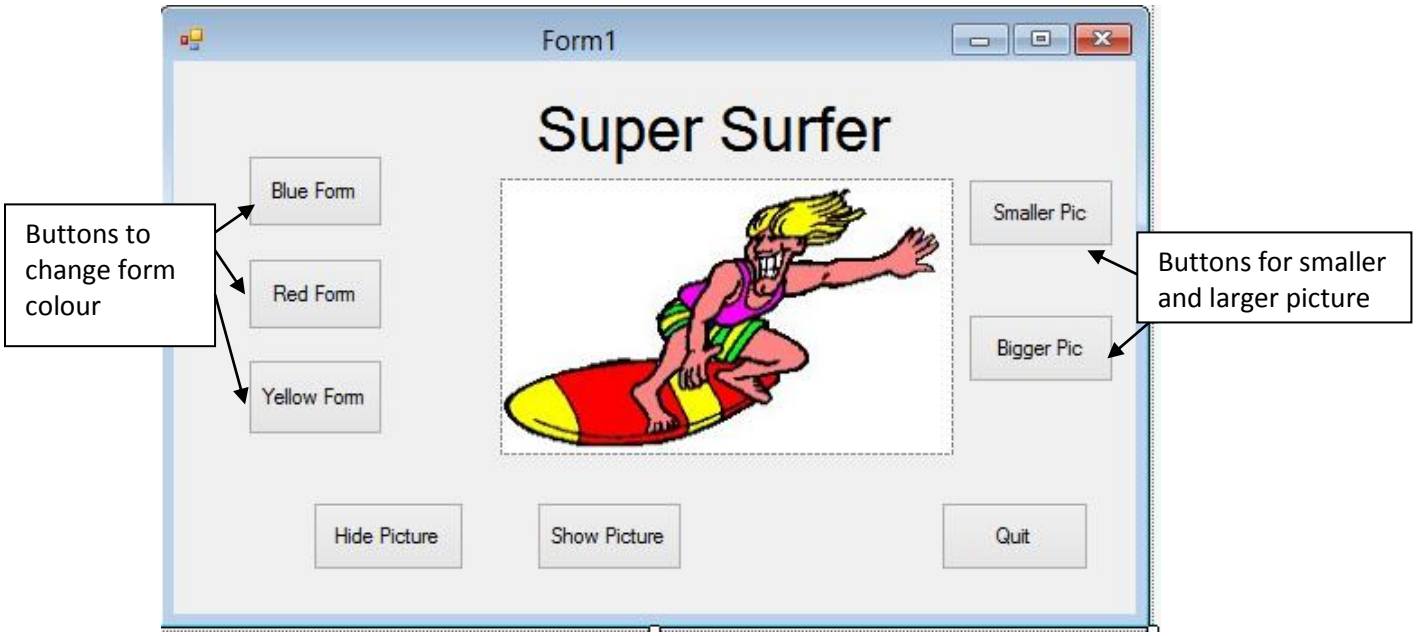
Get user press input change the form colour to blue

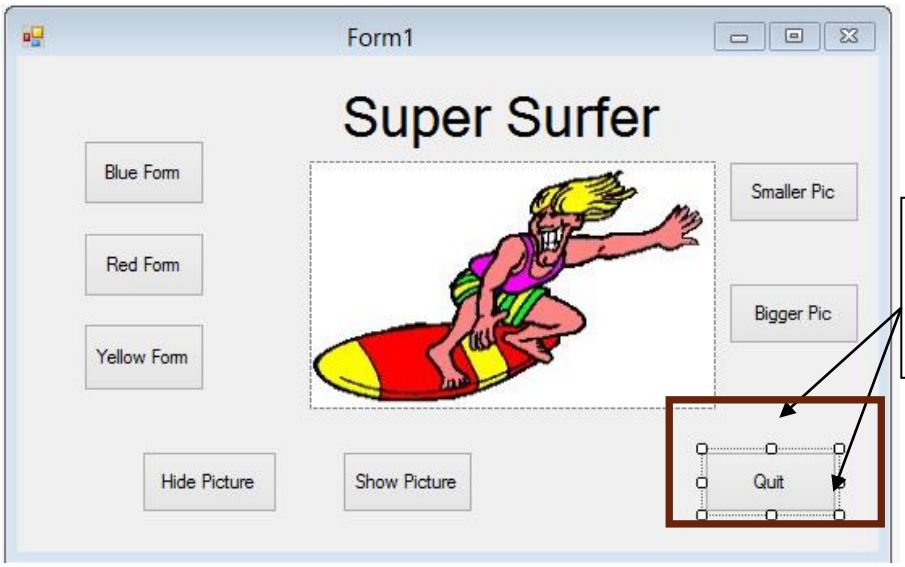
Get user press input change the form colour to yellow

Get user press input change the form colour to red

Get user input to quit the program when ready

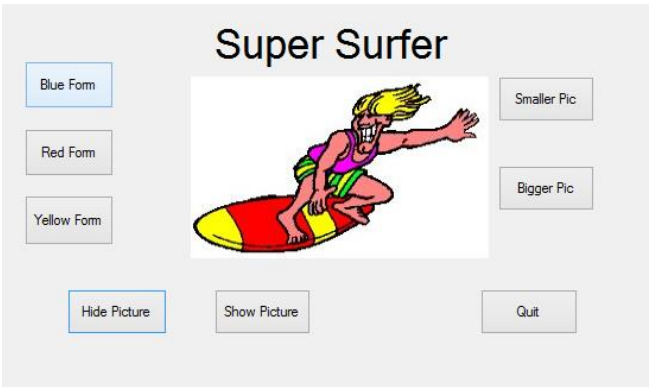
Sample Form Prototype in action

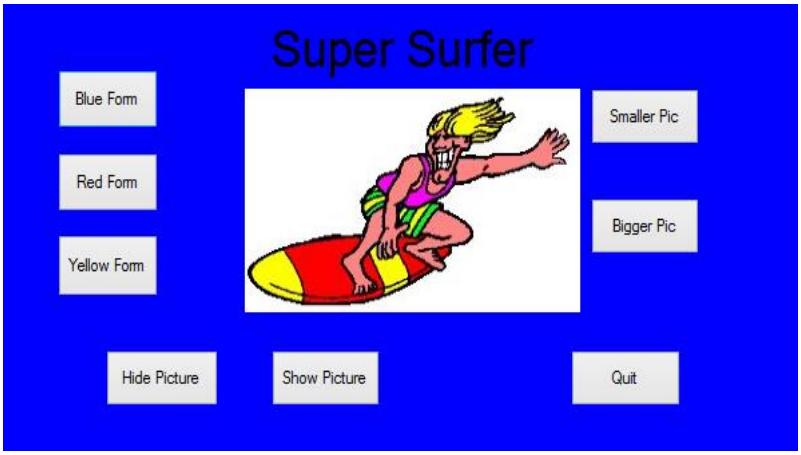




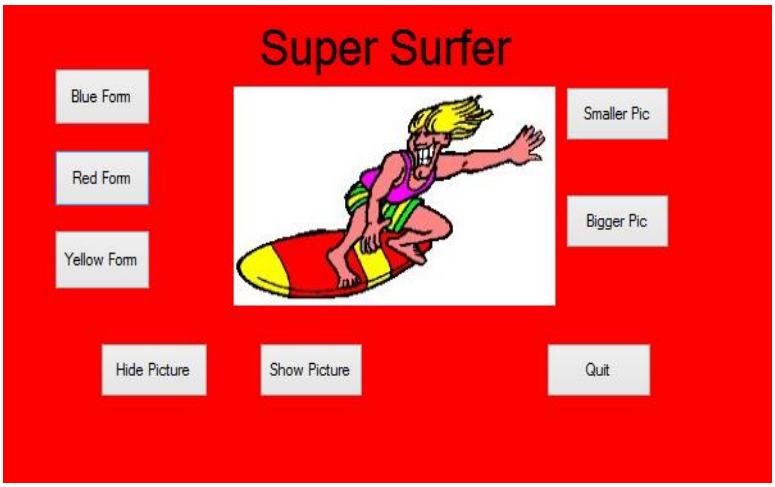
If the user presses on the quit button the form will do quit and shut down

The Form starts off a grey colour but there are buttons to change its Colour

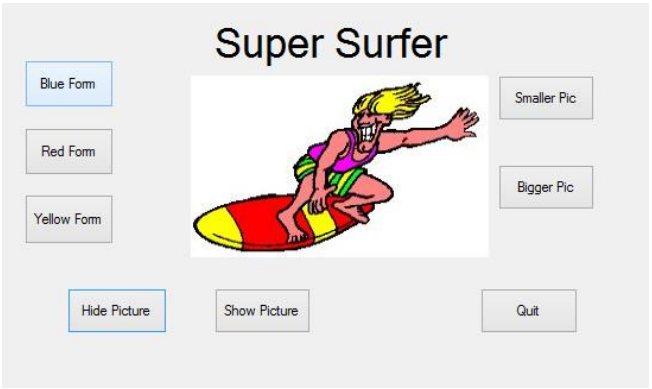


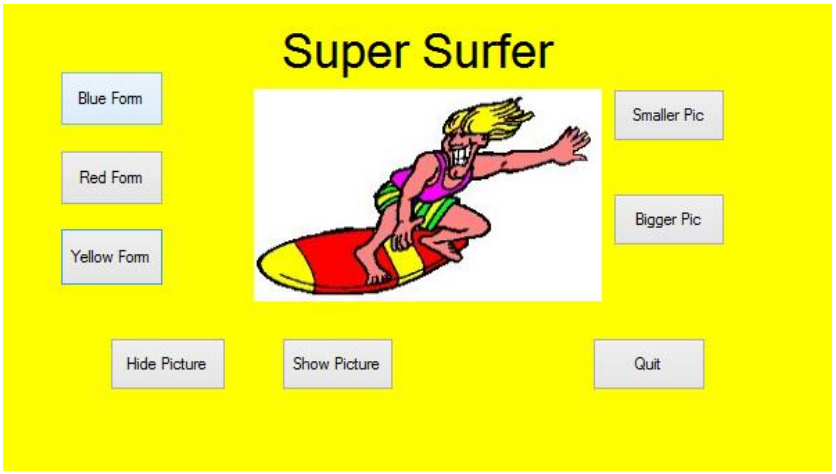


When running the form if a user presses on the blue form button the form colour background changes to Blue



When running the form if a user presses on the red form button the form colour background changes to Red





When running the form this time if a user presses on the Yellow form button the form colour background changes to Yellow

Super Surfer 1.5 Source code

```
// Robert Collcott
// ID 21302939
// Computing
// Super Surfer exercise 1.5
// Unit 1 and Week 1 of C# Windows Programming
// 19th Febaury 2015


namespace _1._5_Super_Surfer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void pictureBox1_Click(object sender, EventArgs e)      // Picture of
the super surfer
        {

        }

        private void button4_Click(object sender, EventArgs e)  // Change colour of
form to blue
        {
            this.BackColor = Color.Blue;

        }

        private void button3_Click(object sender, EventArgs e)      // quit the
program
        {
            Application.Exit();
        }

        private void button5_Click(object sender, EventArgs e)      // Change colour
of form to red
        {
            this.BackColor = Color.Red;

        }

        private void button6_Click(object sender, EventArgs e)      // Change colour
of form to yellow
        {
            this.BackColor = Color.Yellow;

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

C# Windows Week 2 – 26th February – Calculator Projects

This week I am building a simple calculator for a person to use on a PC or an app to use

Calculator Project Version 1 2.6

Algorithm Logic

Declare Variables

Create form

Drag objects onto form

Create buttons

Get user Input for first number

Choose method to use with first number

- Add
- Subtract
- Multiply
- Divide

Store first number in calculator

Get user input to press = to store the second number

Store second number

Add Subtract Multiply or Divide numbers

Output the result of the two numbers

Display final result

Create calculator functions

- Add
- Subtract
- Multiply
- Divide

Perform certain calculations

Perform add sum
Perform subtraction sum
Perform multiplication sum

Perform division sum

Output results of sums done by pressing the = button on the calculator

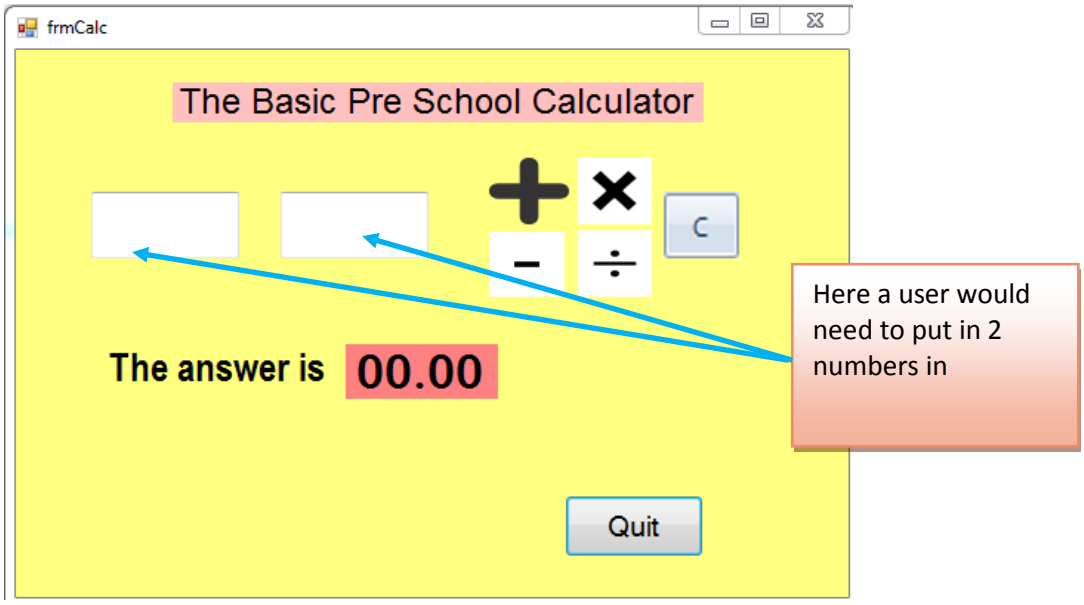
Source Code

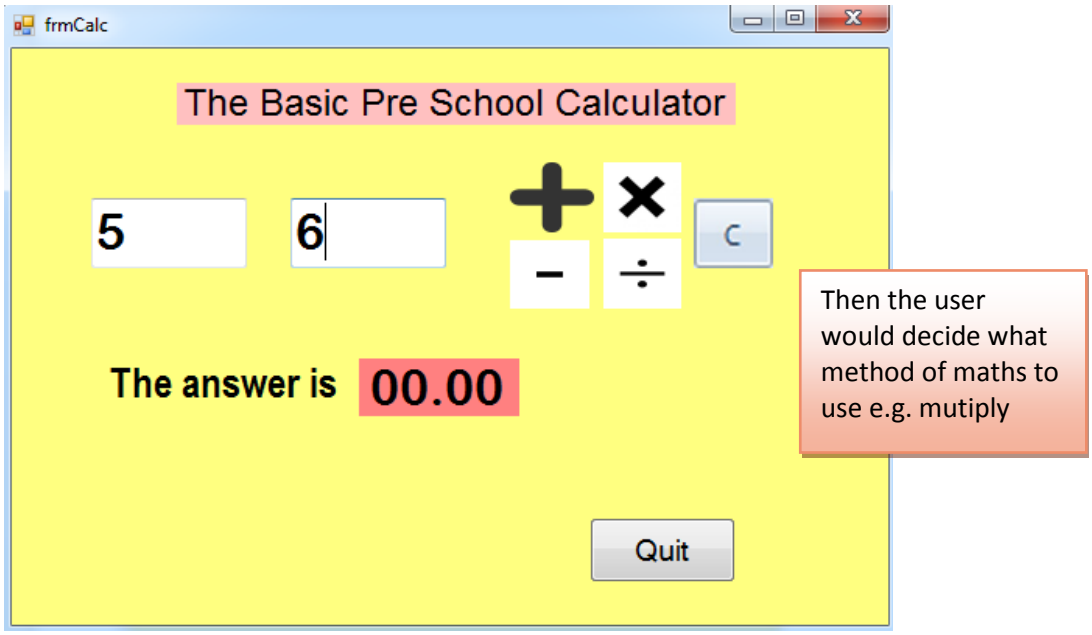
```
//Robert Collcott
    //ID:21302939
    // Computing
    // May 2015
       // Basic Calculator for preschool kids

namespace Calculator
{
    public partial class Form1 : Form
    {
        double n1, n2, answer;                  // These are the variables

        public Form1()
        {
            InitializeComponent();
        }

        private void label4_Click(object sender, EventArgs e)
        {

        }

        private void btnMul_Click(object sender, EventArgs e)
        {
            answer = n1 * n2;                                       //This multipplys the
two numbers together
            lblResult.Text = answer.ToString();
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            n2 = Convert.ToDouble(txtSecondNum.Text);           // convert the second
number to double
        }

        private void lblResult_Click(object sender, EventArgs e)
        {

        }

        private void btnAdd_Click(object sender, EventArgs e)
        {
            answer = n1 + n2;                                   //This will add 2 numbers
together
        }

        private void txtFirstNum_TextChanged(object sender, EventArgs e)
        {
            n1 = Convert.ToDouble(txtFirstNum.Text); // convert the first number to
double
        }

        private void button1_Click(object sender, EventArgs e)
        {


        }

        private void btnSub_Click(object sender, EventArgs e)
        {
            answer = n1 - n2; //This subtracts the two numbers together
            lblResult.Text = answer.ToString();
        }
```

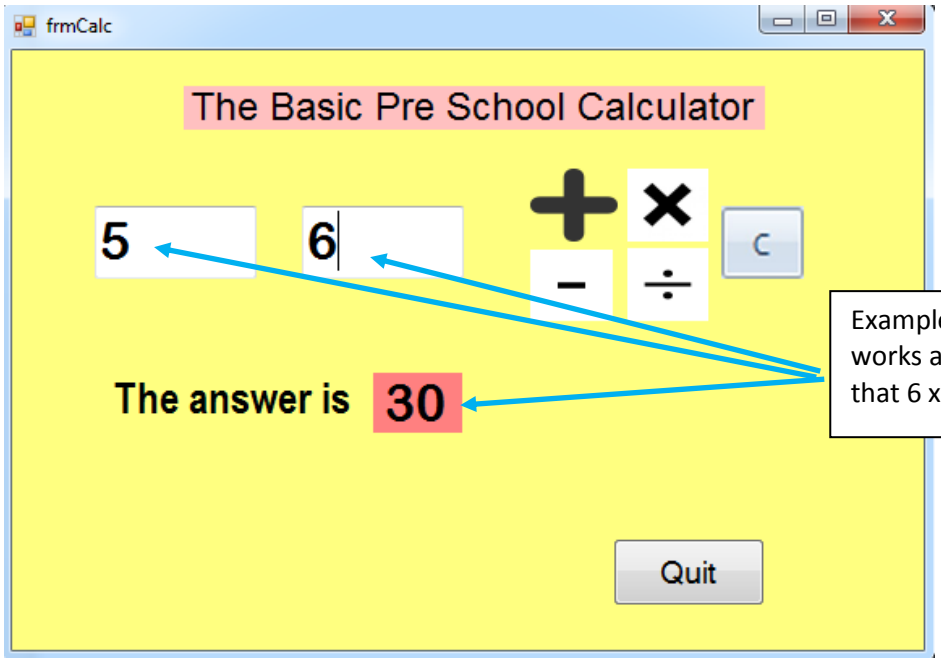
```
        private void btnQuit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void lblInfinity_Click(object sender, EventArgs e)
        {

        }

        private void timer1_Tick(object sender, EventArgs e)
        {

        }

        private void btnDiv_Click_1(object sender, EventArgs e)
        {
            if (n1 == 0 | n2 == 0) // If any number is divided by zero this error
message will come up
            {
                lblResult.Text = "You cannot divide by \"0\"";
            }
            else // Else continue operating
            {
                answer = n1 / n2;
                lblResult.Text = answer.ToString();
            }

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void btnPow_Click(object sender, EventArgs e)
        {
            answer = (n1 + n2) / 2; //Calculate the average
            lblResult.Text = answer.ToString(); //Show the result
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            answer = Math.Pow(n1, n2);                                      //Calculate
the first number with the second number
            lblResult.Text = answer.ToString();                             //Show the
result
        }

        private void btnClear_Click(object sender, EventArgs e)
        {
            lblResult.Text = "";

        }

        private void lblR_Click(object sender, EventArgs e)
        {

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {
            answer = n1 + n2;                                                   //This
addds the two numbers together
```

```
            lblResult.Text = answer.ToString();
        }

        private void pictureBox2_Click(object sender, EventArgs e)
        {
            answer = n1 * n2;                                           //This
multipplys the two numbers together
            lblResult.Text = answer.ToString();
        }

        private void pictureBox3_Click(object sender, EventArgs e)
        {
            answer = n1 - n2;                                          //This
subtracts the two numbers together
            lblResult.Text = answer.ToString();
        }

        private void pictureBox4_Click(object sender, EventArgs e)
        {
            if (n1 == 0 | n2 == 0)                                     // If any
number is divided by zero show the error message
            {
                lblResult.Text = "You canot divide by \"0\"";
            }
            else                                                       // Else
continue operating the numbers
            {
                answer = n1 / n2;
                lblResult.Text = answer.ToString();
            }

        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            lblResult.Text = "";
        }
    }
}
```

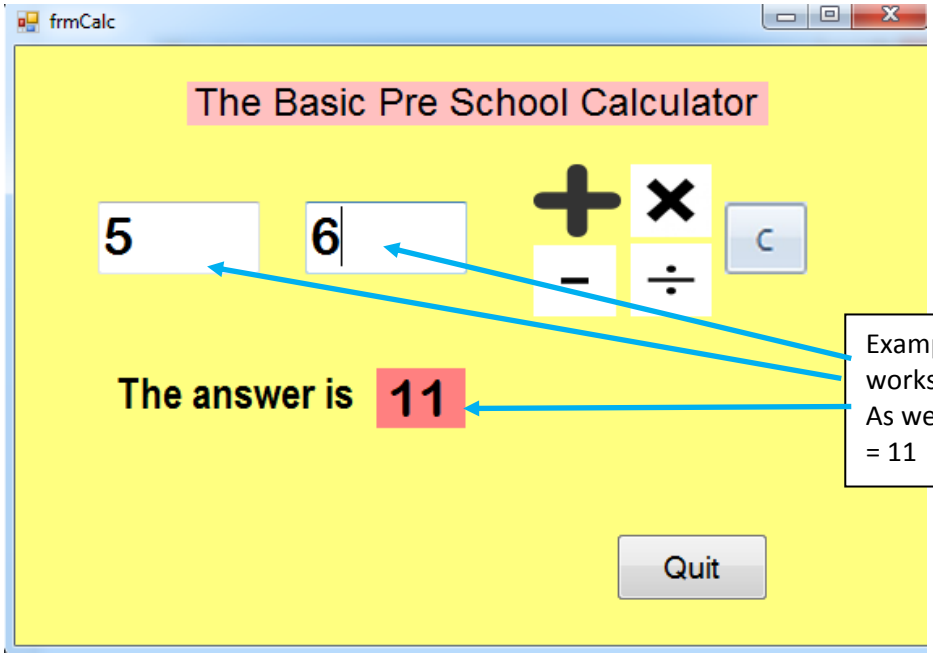
Basic Calculator sample outputs







Example multiplication sum works as everyone knows that 6 x 5 = 30



Example addition sum works
As we all know that 6 add 5 = 11

Simple Calculator Program TOE chart [Task Object Event Chart]			
Task No	**Task**	**Object**	**Event**
1	Display the interface	frmCalc	
2	Enter First Number	txtFirstNum	
3	Enter Second Number	txtSecNum	
4	Perform Addition	btnAdd	Click
5	Perform Subtraction	btnSub	Click
6	Perform Multiplication	btnMult	Click
7	Perform Division	btnDiv	Click
8	Display Result	lblResult	
9	Store First Number		
10	Store Second Number		
11	End Program	btnQuit	Click

Here is the TOE chart for the Calculator program I am about to build

Key and Note
Black Fill in a box means the thing does not apply in other words **N/A [Not Applicable]**

Advanced Calculator Project Version 2 Task No 2.7

This calculator is similar to the one I had designed in the pervious independent study but this time the calculator has 10 buttons which are going to be labelled one to 9 this will make a number pad, in addition they will be an extra for buttons for basic functionality these are

- Add
- Subtract
- Multiply
- Divide

There will also be a label to display the number as its entered by the user using the number pad.

For example hitting the multiply key will store the first number inputted by the user on the pad this then clears the number

While pressing the = button will store the second number and carry out the necessary method such as a subtraction sum.

Now I am going to write the complex algorithm for the calculator version 2

Algorithm Logic

Declare Variables

Create form

Create buttons

Create functions for each button

Get user Input for first number

Choose method to use with first number from the following

- Add
- Subtract
- Multiply
- Divide

Store first number in calculator

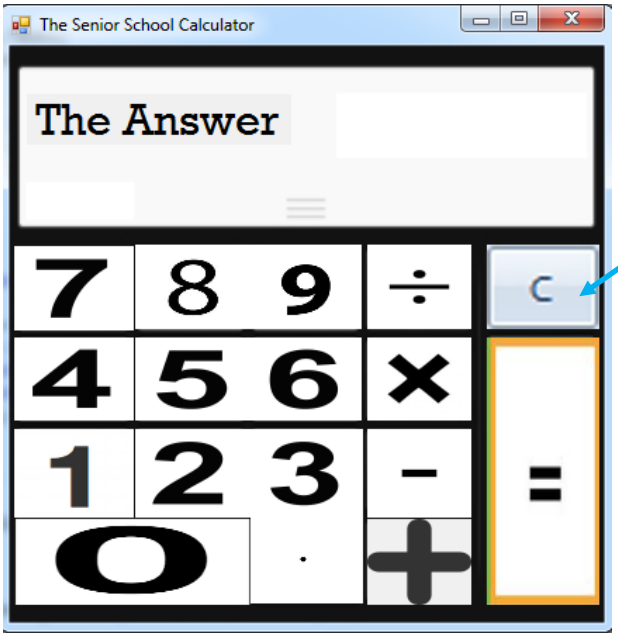
Get user input to press to store the second number

Store second number

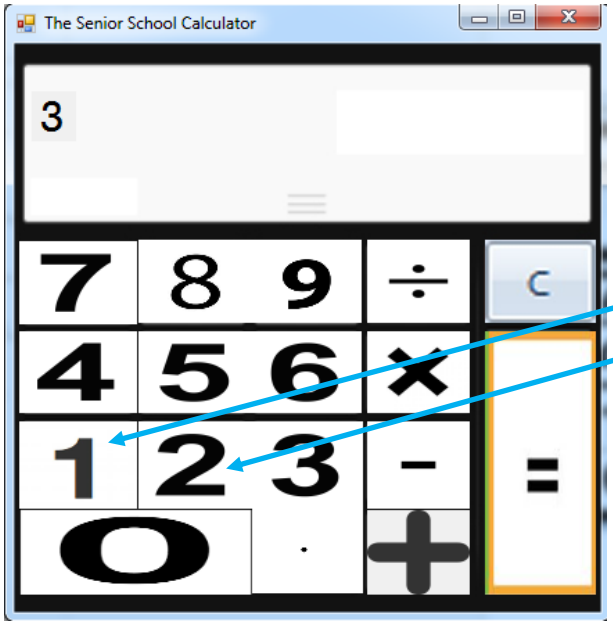
Output the result of the two numbers

Display final result

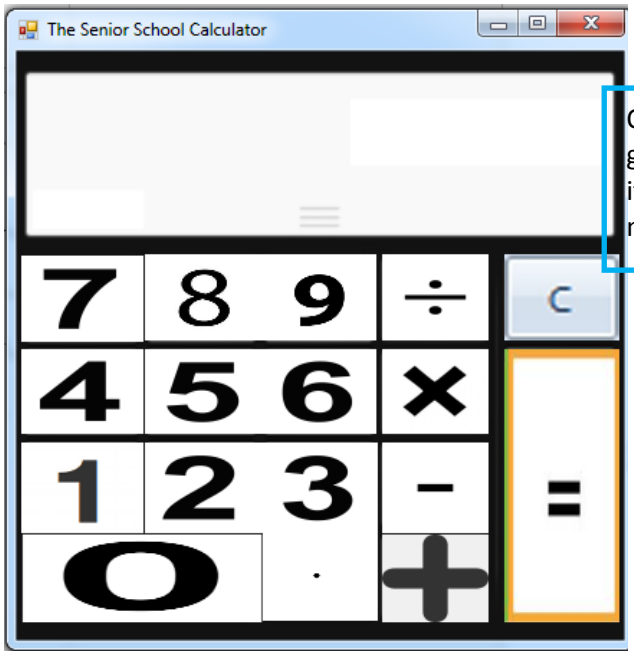
Advanced Calculator 2.7 Sample Outputs



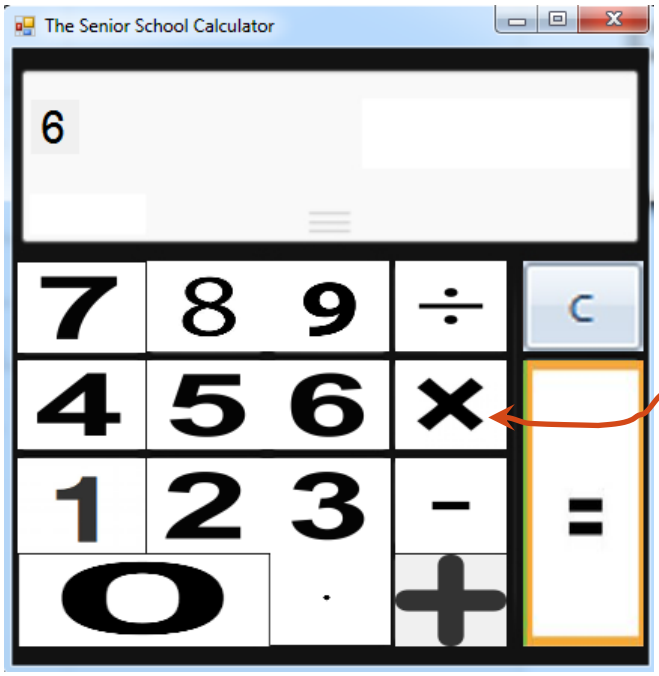
This is a more advanced calculator



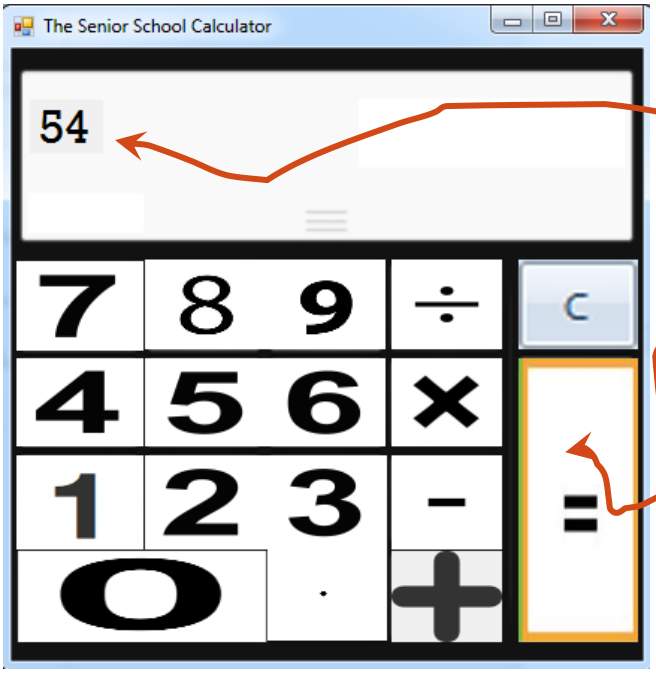
Pressing 12 by pressing 1 and a 2 on the calculator



Ok the number is gone good because that means its stored it in the memory of it



Now the for the method which in this case multiply



Method processed and by hitting the equals button we will get an output of an answer

Answer in this case is 54 as when maths was invented years ago someone said 12 x 5 = 54 and they are still correct today

Source Code

```
  //  Robert Collcott
                    // ID 21302939
                    // Computing
                    // 28th May 2015
                     // Advanced Calculator

namespace Calculator
{
    public partial class Form1 : Form
    {
        double result = 0.0;
        string input = string.Empty;   //all verables
        string num1 = string.Empty;
        string num2 = string.Empty;
        char operation;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e) //load the form
        {

        }
        //************************************************************** all
calculation
        private void btn0_Click(object sender, EventArgs e)                 // This is
the number 0 on the calculator
        {
            this.lblResult.Text = "";
            input += "0";
            this.lblResult.Text += input;
        }

        private void btnPoint_Click(object sender, EventArgs e)             // This is
the point button on the calculator
        {
            this.lblResult.Text = "";
            input += ".";
            this.lblResult.Text += input;
        }

        private void btn1_Click(object sender, EventArgs e)                 // This is
the number 1 on the calculator
        {
            this.lblResult.Text = "";
            input += "1";
            this.lblResult.Text += input;
        }

        private void btn2_Click(object sender, EventArgs e)                 // This is
the number 2 on the calculator
        {
            this.lblResult.Text = "";
            input += "2";
            this.lblResult.Text += input;
        }

        private void btn3_Click(object sender, EventArgs e)                 // This is
the number 3 on the calculator
        {
            this.lblResult.Text = "";
```

```
            input += "3";
            this.lblResult.Text += input;
        }

        private void btn4_Click(object sender, EventArgs e)                 // This is
the number 4 on the calculator
        {
            this.lblResult.Text = "";
            input += "4";
            this.lblResult.Text += input;
        }

        private void btn5_Click(object sender, EventArgs e)              // This is the
number 5 on the calculator
        {
            this.lblResult.Text = "";
            input += "5";
            this.lblResult.Text += input;
        }

        private void btn6_Click(object sender, EventArgs e)             // This is the
number 6 on the calculator

        {
            this.lblResult.Text = "";
            input += "6";
            this.lblResult.Text += input;
        }

        private void btn7_Click(object sender, EventArgs e)              // This is the
number 7 on the calculator

        {
            this.lblResult.Text = "";
            input += "7";
            this.lblResult.Text += input;
        }

        private void btn8_Click(object sender, EventArgs e)          // This is the
number 8 on the calculator

        {
            this.lblResult.Text = "";
            input += "8";
            this.lblResult.Text += input;
        }

        private void pictureBox4_Click(object sender, EventArgs e)
// This is the number 9 on the calculator

        {
            this.lblResult.Text = "";
            input += "9";
            this.lblResult.Text += input;
        }

        private void btnAdd_Click(object sender, EventArgs e)               // the add
button code addition
        {
            num1 = input;
            operation = '+';
            input = string.Empty;
            this.lblResult.Text = "";
        }
```

```
        private void btnMinus_Click(object sender, EventArgs e)             // this
button performs subtraction sums

        {
            num1 = input;
            operation = '-';
            input = string.Empty;
            this.lblResult.Text = "";
        }

        private void btnMulti_Click(object sender, EventArgs e)             // this
button performs mutiplaction sums
        {
            num1 = input;
            operation = '*';
            input = string.Empty;
            this.lblResult.Text = "";
        }

        private void btnDiv_Click(object sender, EventArgs e)               // this
button divides sums
        {
            num1 = input;
            operation = '/';
            input = string.Empty;
            this.lblResult.Text = "";
        }

        private void btnClear_Click(object sender, EventArgs e)        // this button
clears the calculator
        {
            this.lblResult.Text = "";
            this.input = string.Empty;
            this.num1 = string.Empty;
            this.num2 = string.Empty;
        }

        private void btnResult_Click(object sender, EventArgs e)        // calculates
the result
        {
            num2 = input;
            double n1, n2;
            double.TryParse(num1, out n1);
            double.TryParse(num2, out n2);
            this.lblResult.Text = "";
            this.input = string.Empty;
            this.num1 = string.Empty;
            this.num2 = string.Empty;

            if (operation == '+')
            {
                result = n1 + n2;
                lblResult.Text = result.ToString();
            }

            else if (operation == '-')
            {
                result = n1 - n2;
                lblResult.Text = result.ToString();
            }

            else if (operation == '*')
            {
                result = n1 * n2;
                lblResult.Text = result.ToString();
            }
```

```
            else if (operation == '/')
            {
                if (n2 != 0)
                {
                    result = n1 / n2;
                    lblResult.Text = result.ToString();
                }
            }
        }

        private void lblResult_Click(object sender, EventArgs e)
        {

        }
    }
}
```

Algorithm Logic

Declare Variables

Create the form

Create buttons

Get user input for the player name

Get user choice

Scissors
OR
Paper
OR
Stone

Declare random class for computer to pick

Scissors
OR
Paper
OR
Stone

IF

Player and Computer pick the same thing the result is a draw

IF

Player picks Scissors and computer picks paper
Player wins because Scissors cut paper

IF

Player chooses Scissors
Computer chooses Stone
Scissors loose because stone will blunt the Scissors

IF

Player chooses Paper
Computer chooses Stone

Paper wins because paper wraps stone

Output result on the screen from one of the following

- You win [Player]
- Computer
- Match ends in draw

Display player score

Display computer score

When games played comes up to 20

STOP THE GAME

IF

Computer reaches 20 points display a splash screen saying keep calm you loose

IF

Player reaches 20 points display a splash screen with a ha ha I win image this means the player wins

IF

Game ends in a draw display a splash screen Its a draw

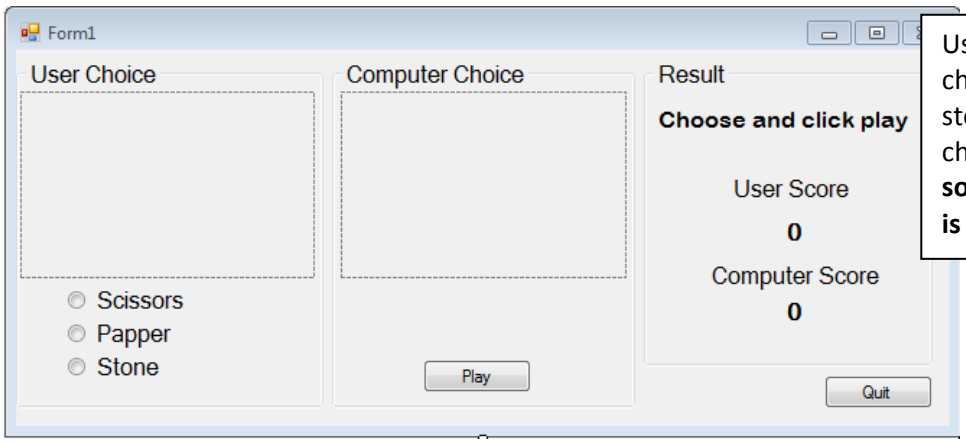
Source Code

```
        //  Robert Collcott
        //  ID:21302939
        //  Computing
        //  2014-2015 Academic Year


namespace SPS_Game
{
    public partial class Form1 : Form
    {
        int compChoice, userChoice;                 // 2 variables here 1 for user
player choice and the other one for computer choice
        int userScore = 0, compScore = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void btnQuit_Click(object sender, EventArgs e)              // This
exits the program
        {
            Application.Exit();
        }

        private void rbnScissors_CheckedChanged(object sender, EventArgs e)
        {
            pbxUser.Image = Image.FromFile("Scissors.jpg");    // Get a scissors image
from a file and display it
            userChoice = 1;
        }

        private void rbnPapper_CheckedChanged(object sender, EventArgs e) // Get a
Paper image from a file and display it
        {
            pbxUser.Image = Image.FromFile("Paper.jpg");
            userChoice = 2;
        }

        private void rbnStone_CheckedChanged(object sender, EventArgs e) // Get a
Stone image from a file and display it
        {
            pbxUser.Image = Image.FromFile("Stone.jpg");
            userChoice = 3;
        }

        private void btnPlay_Click(object sender, EventArgs e) // This is the button
that starts the game off
        {
            Random r = new Random(); // Generate a random choice for the computer
player
            compChoice = r.Next(3) + 1;

            if (compChoice == 1)
            {
                pbxComputer.Image = Image.FromFile("Scissors.jpg");             //
Display the scissors image if computer choice is 1
            }
            else if (compChoice == 2)
            {
                pbxComputer.Image = Image.FromFile("Paper.jpg");            // Display
the paper image if computer choice is 2
            }
            else if (compChoice == 3)
```

```
            {
                pbxComputer.Image = Image.FromFile("Stone.jpg");        // Display the
stone image if computer choice is 3
            }
            chkResult();
            lblCompScore.Text = compScore.ToString();
            lblUserScore.Text = userScore.ToString();


            if (userScore >= 20) // If the user score gets up to 20 display an image
on a screen saying the player has won
            {
                new YouWin().Show();
                this.Hide();
                System.Threading.Thread.Sleep(2000);
                this.Close();
            }
            else if (compScore >= 20)  // If the computer score gets up to 20 display
an image on a screen saying the computer has won
            {
                new CompWin().Show();
                this.Hide();
                System.Threading.Thread.Sleep(2000);
                this.Close();
            }
            else if (userScore >= 20 & compScore >= 20)         // if player and
computer score match call the game and display a message saying ITS A DRAW.
            {
                new Draw().Show();
                this.Hide();
                System.Threading.Thread.Sleep(2000);
                this.Close();
            }

//*************************************************************************************
**********
        }

        private void pbxComputer_Click(object sender, EventArgs e)
        {

        }                                                       // check result function
        private void chkResult()
        {
            if (userChoice == compChoice) // Call it a draw if the player choice and
the computer choice
            {                                                               // match
                lblResult.Text = "DRAW!";
                this.BackColor = Color.White;
                this.ForeColor = Color.Black;
                userScore++;   // Increment the player score up by 1 each time
                compScore++;    // Increment the computer score up by 1 each time
            }
            else if (userChoice == 2 & compChoice == 3)             // if player
choice is 2 and computer choice is 3
            {                                                       // the player wins
because paper blunts paper
                lblResult.Text = "You Win!!";
                this.BackColor = Color.LightBlue;
                this.ForeColor = Color.Orange;
                userScore = userScore + 2;
            }
            else if (userChoice == 2 & compChoice == 1) // if player choice is 2 and
computer choice is 1
            {                   // the computer wins because scissors cut paper
                lblResult.Text = "The Computer Wins this time!!";
```

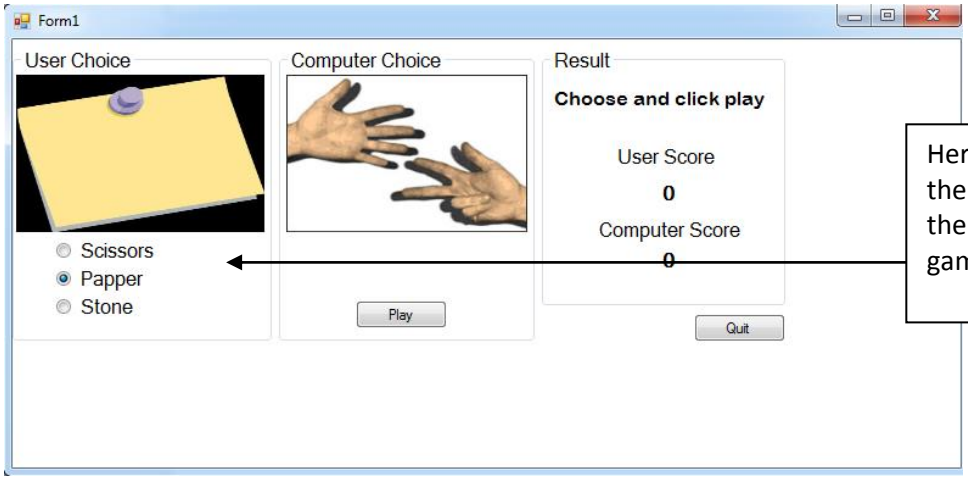
```
                this.BackColor = Color.White;
                this.ForeColor = Color.Black;
                compScore = compScore + 2;
            }
            else if (userChoice == 1 && compChoice == 2)
            {
                lblResult.Text = "You Win!!";
                this.BackColor = Color.LightBlue;
                this.ForeColor = Color.Orange;
                userScore = userScore + 2;
            }
            else if (userChoice == 1 && compChoice == 3)
            {
                lblResult.Text = "Computer Wins!!";
                this.BackColor = Color.White;
                this.ForeColor = Color.Black;
                compScore = compScore + 2;
            }
            else if (userChoice == 3 && compChoice == 2)
            {
                lblResult.Text = "Computer Wins!!";
                this.BackColor = Color.White;
                this.ForeColor = Color.Black;
                compScore = compScore + 2;
            }
            else if (userChoice == 3 && compChoice == 1)
            {
                lblResult.Text = "You Win!!";
                this.BackColor = Color.LightBlue;
                this.ForeColor = Color.Orange;
                userScore = userScore + 2;
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            frmSplash Sscreen = new frmSplash();         // Load up the splash screen
            Sscreen.Show();
            System.Threading.Thread.Sleep(10000);
            Sscreen.Close();
            this.BackColor = Color.Black;           // Black is the background colour
            this.ForeColor = Color.White;       // White is foreground colour
            pbxComputer.Image=Image.FromFile("sps3.jpg");
            pbxUser.Image=Image.FromFile("sps1.jpg");
        }
    }
}
```

Sample Outputs

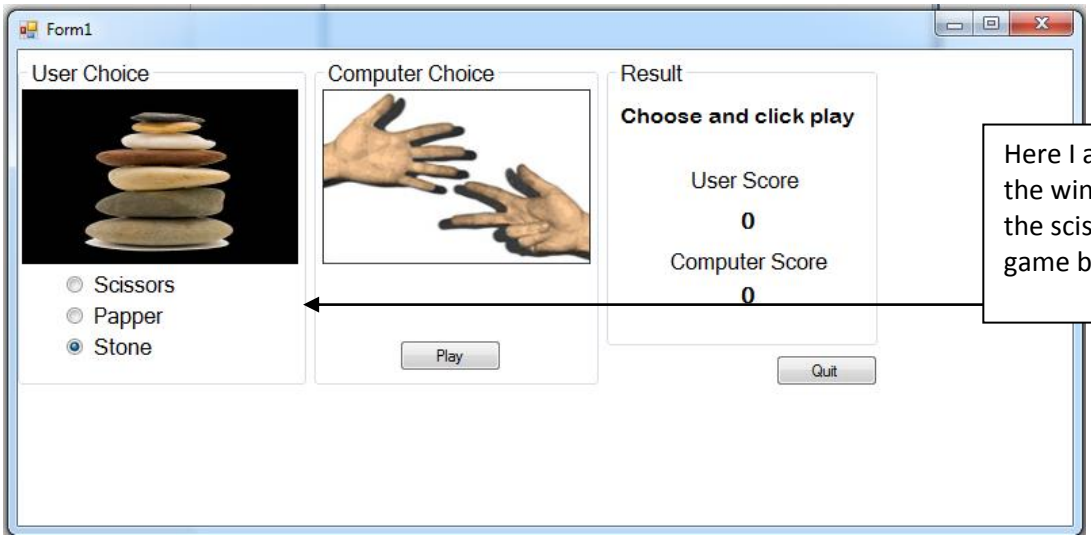


Users have the option to choose scissors paper or stone the computer choice is chosen at Random - **see source code to see how that is done**

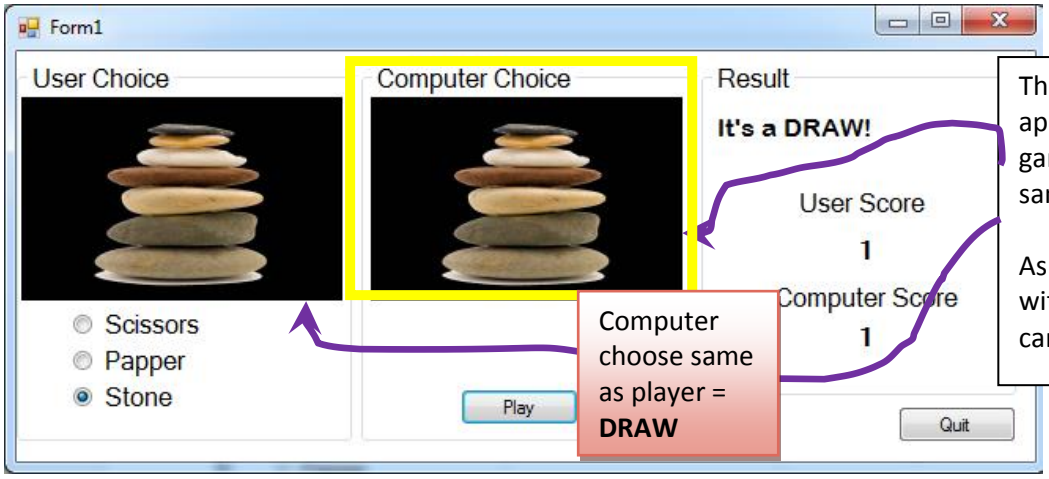
Here I am about to play the windows version of the scissors paper stone game by choosing scissors



Here I am about to play the windows version of the scissors paper stone game by choosing Paper



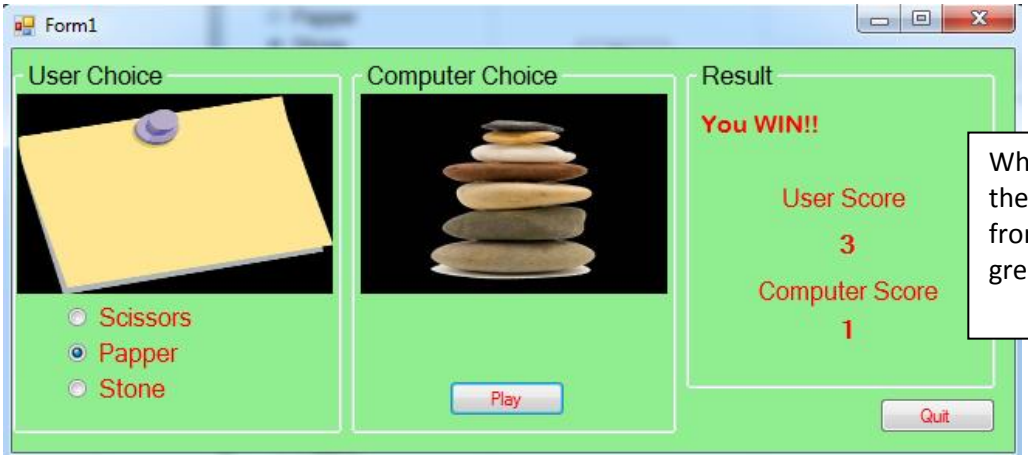
Here I am about to play the windows version of the scissors paper stone game by choosing Stone



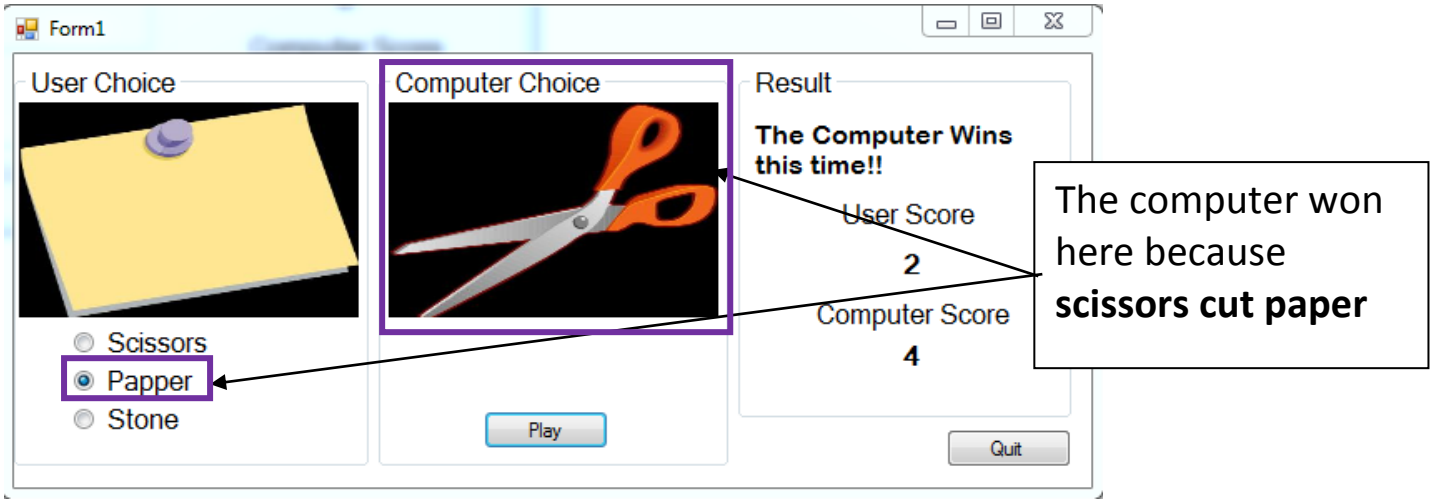
This is the message that will appear if you as a player of the game and the computer chose the same thing

As shown in the example here with Stone well 2 sorts of stone cannot do anything

Computer choose same as player = **DRAW**



When you beat the computer the background changes from grey to a light pale green sea colour



Here is an example of the computer winning a round of the windows version of the scissors paper stone game

Scissors Paper Stone Game Windows Version
Sample splash screens that appear when the game ends

This is the winning splash screen if you manage to beat the computer

This is the splash screen that you as the player will see if the game ends in a draw this can only happen if both scores are at 20

This is the splash screen that you as the player will see if you loose the game

C# Windows Week 4 – Hit The Target– [Message box week]

This week's Independent study is to create a simple basic game called hit the target

Algorithm Logic

Declare Variables

Creation of Form

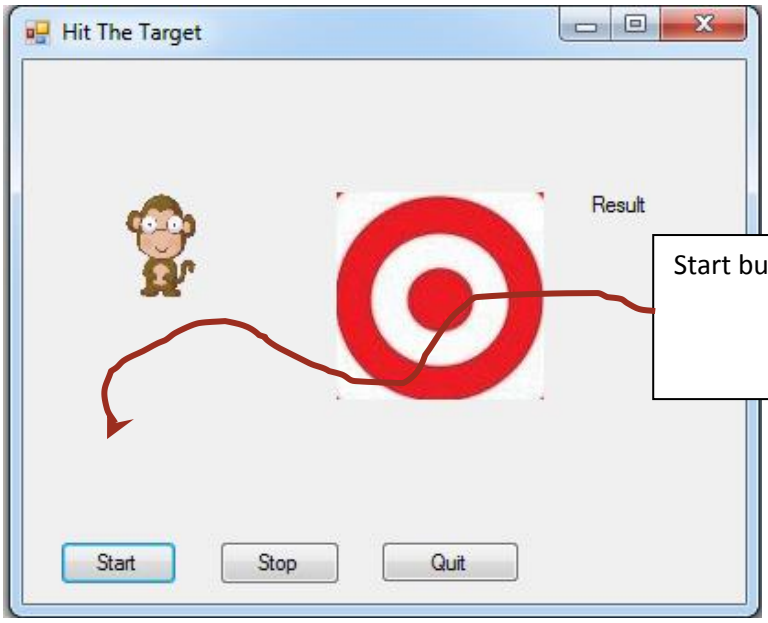
Declare functions

Start a loop

- Calculate the distance via a screen root
- Get user interaction to hit the target by a click
- Display result
- Output score and result when the target is hit
- Store scores for future reference
- Output final score total
- Store final score total

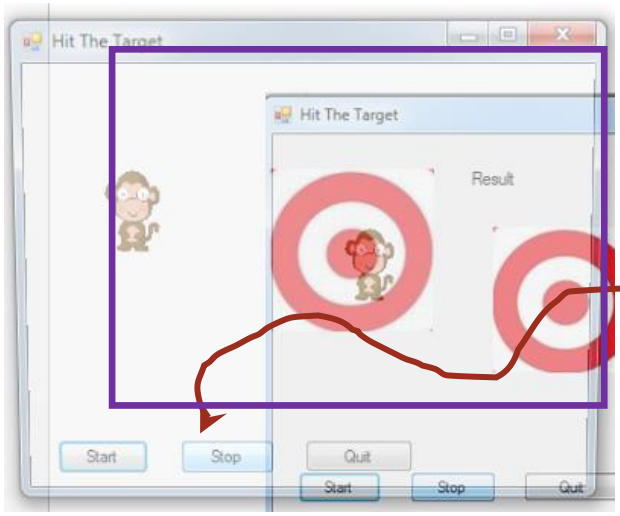
4.7 Hit the Target Sample outputs

Here is the main hit the target form

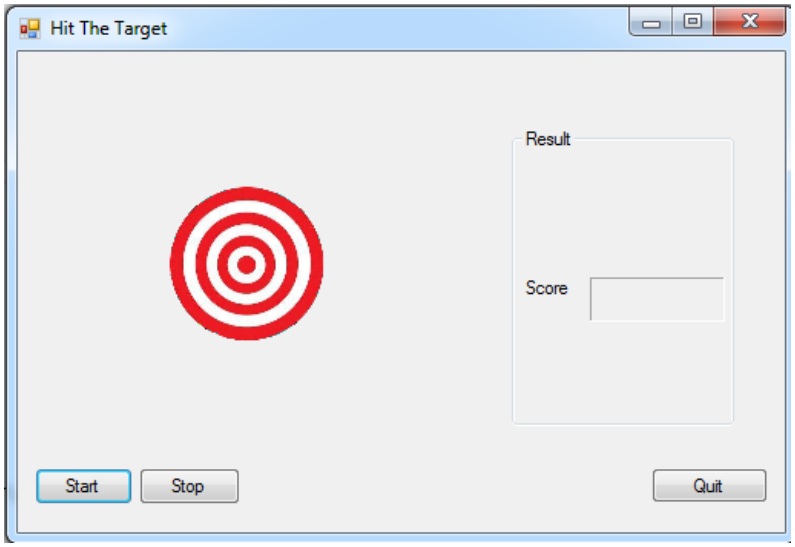


Start button will start the game

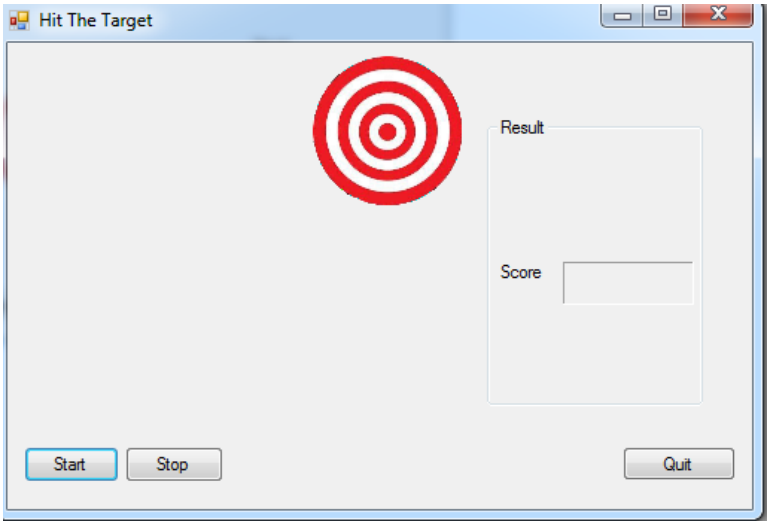
Start button will stop the game and restart the game



Here is the program being restarted

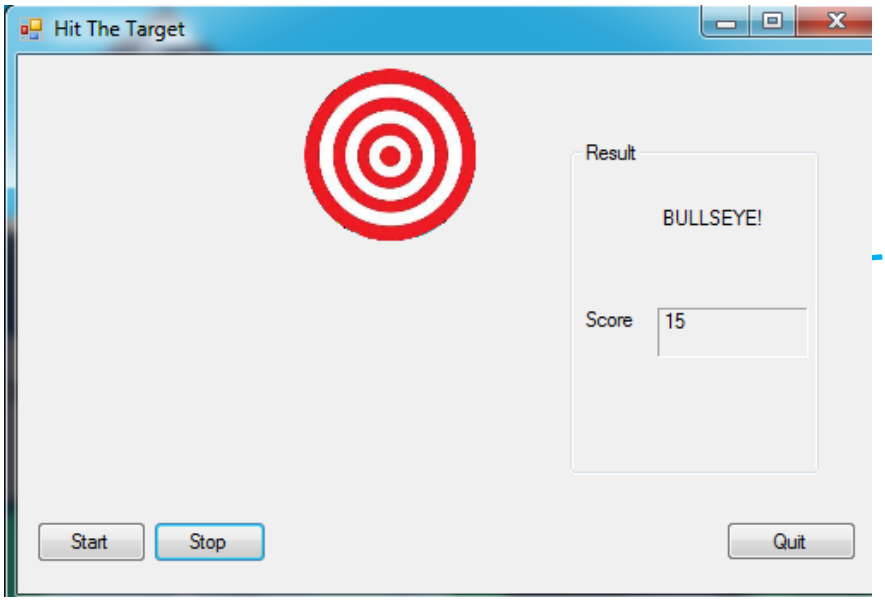


Here the user will press start and the target will move



Here the target is moving and the user of the program will try and hit it

Each time the user hits the target he or she will score a point and if they keep hitting the score will go up



- Each time the user will hits the centre of the target

Points go up by 10 as its known as a Bulls eye hit just like in normal darts

Source Code

```
// Robert Collcott
// ID 21302939
// Computing
// 2014-15 Academic Year
// 28th May 2015


namespace Monkey
{
    public partial class Form1 : Form
    {                                                   // This is the distance varirable
        double distance;
        double Hit=1;

        int x, y;                                       // These are the cordnatinte
varirables
        Random rand = new Random();

        public Form1()
        {
            InitializeComponent();
        }


        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            x = rand.Next(this.Width - 200);
            y = rand.Next(this.Height - 200);
            pbxPicture.Left = x;                        // set the picture to the left
as x which is 200
            pbxPicture.Top = y;                         // set the picture to the left
as y which is 200
            Refresh();                                        // This will re draw the
image in a new postion
        }

        private void btnStart_Click(object sender, EventArgs e)
        {
            timer1.Enabled = true;                                  // start the timer
control
        }

        private void btnStop_Click(object sender, EventArgs e)
        {
            timer1.Enabled = false;                                 // stop the timer
control
        }

        private void button1_Click(object sender, EventArgs e)      // exit the
program completly
        {
            Application.Exit();
        }

        private void pbxPicture_Click(object sender, EventArgs e)           // This is
the picture box that has the target image in
        {
```

```
            lblResult1.Text = Hit.ToString();                           // this is the
hit score
            Hit = Hit + 0;
            Hit++;                                                      // increment
the hits by one each time e.g. 1 to 2
        }

        private void pbxPicture_MouseDown(object sender, MouseEventArgs e)
        {
        }

        private void pbxPicture_MouseClick(object sender, MouseEventArgs e)         //
This is the picture box for the target image
        {
            distance = Math.Sqrt(Math.Pow((e.X - 50), 2) + Math.Pow((e.Y - 50), 2));
            MessageBox.Show("The mouse X position is " + e.X + "\n The mouse Y
position is " + e.Y + "\nThe distance is" + dist);

            if(distance<=20)
            {
                lblB.Text = "Bullseye!";                // This is what will output on
the screen if the distance is less than 20 meters from hitting the target
            }

        }
    }
}
```

Algorithm Logic

Declare Variables

Build menu's

Build menu child links

Make toolbars
- File
- Edit
- Format

Put drop down choices on the following menu tabs
- File
  - New
  - Open
  - Save
  - Exit

- Edit
  - Cut
  - Copy
  - Paste
  - Undo

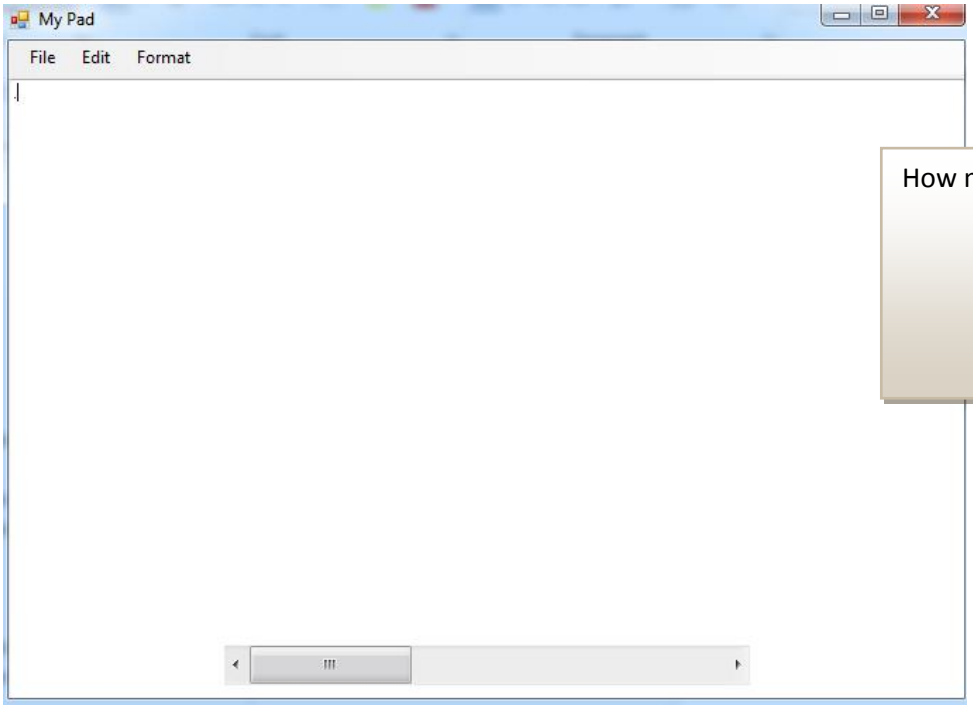
- Format
  - Font
  - Background Colour
  - Foreground Colour

Get user input to use all the functions for the program

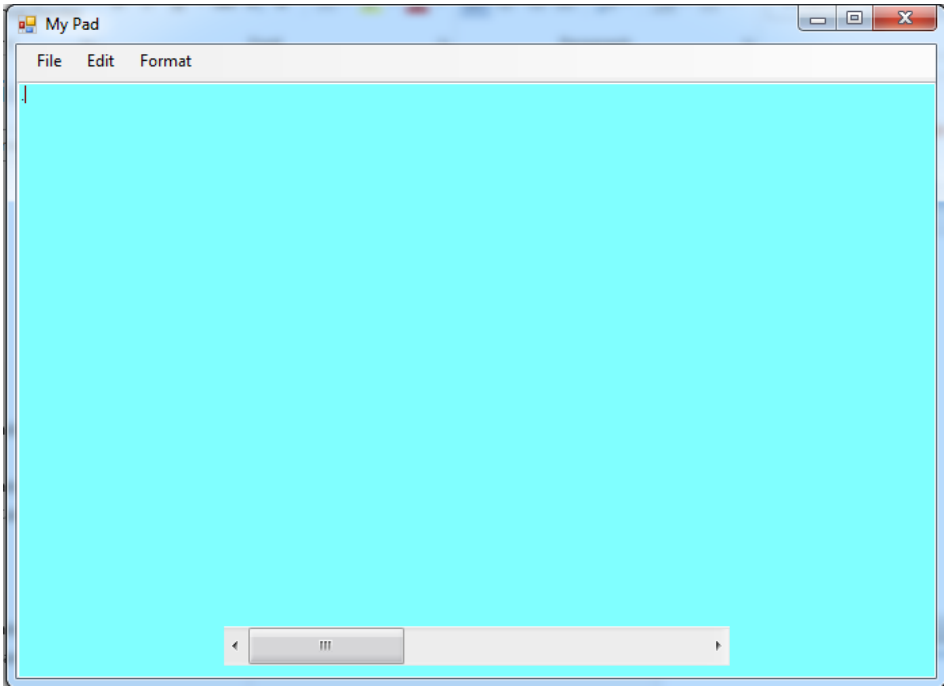
Week 5 Independent Study 5.6 My Pad [A simple version of notepad]

Here I am going to build a program that looks almost like a identical copy of the actual windows notepad editor program

Sample use of the myPad program

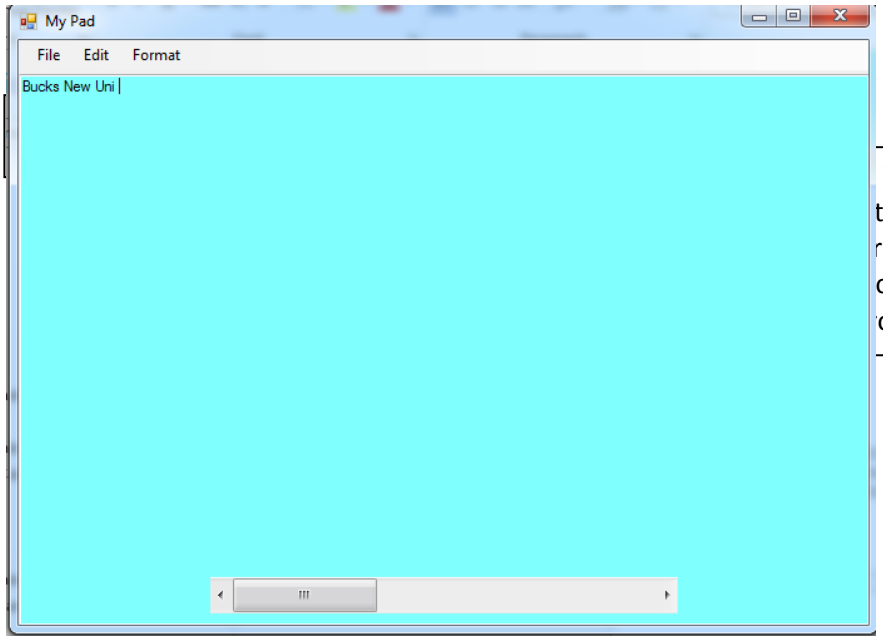


How my Pad starts up

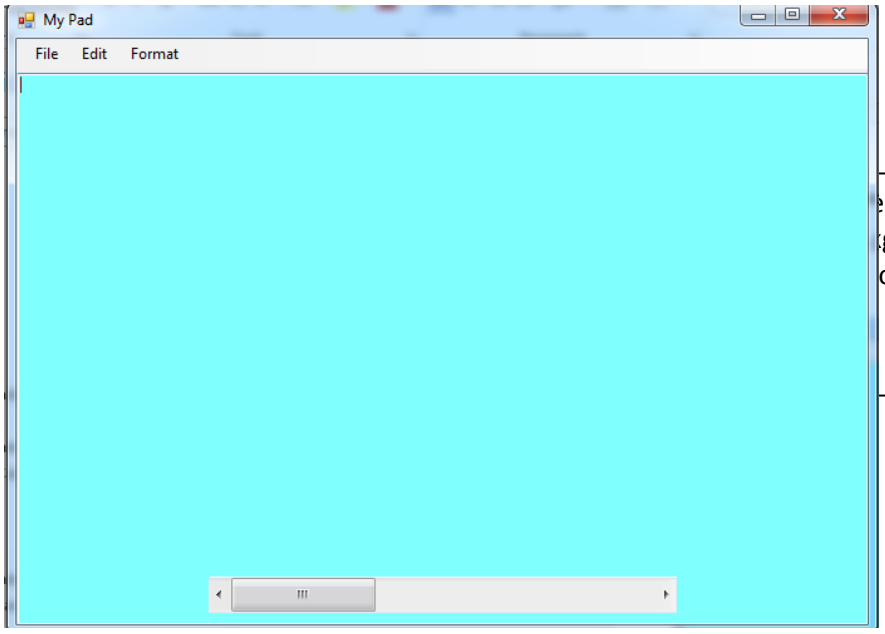


Sample myPad program with text

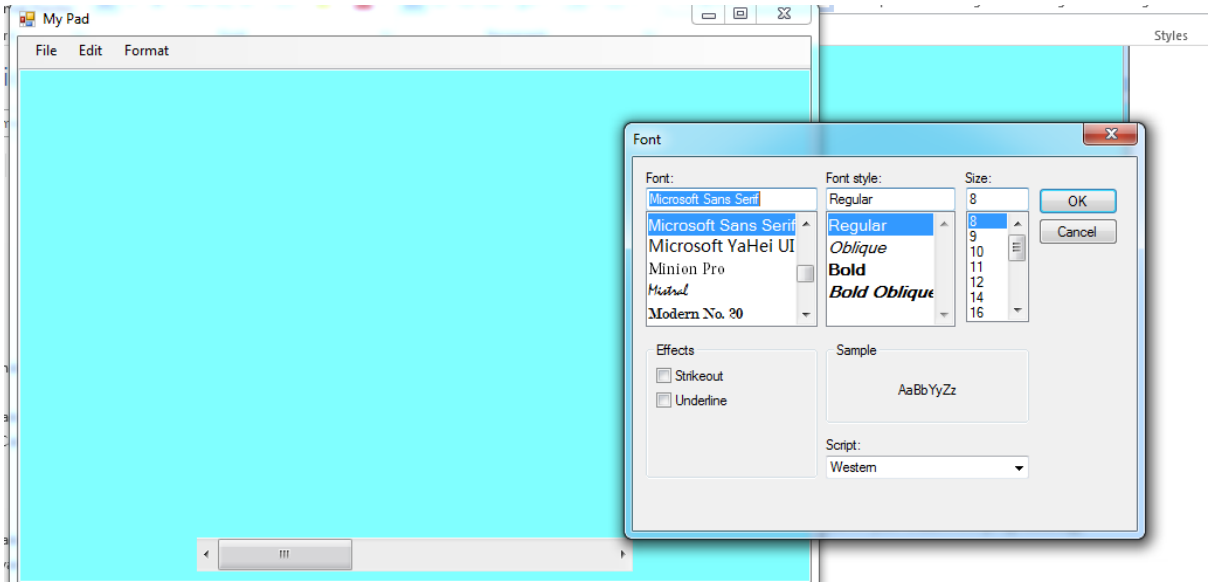
Here you can see I have put in some sample text in to demonstrate the myPad program can input text

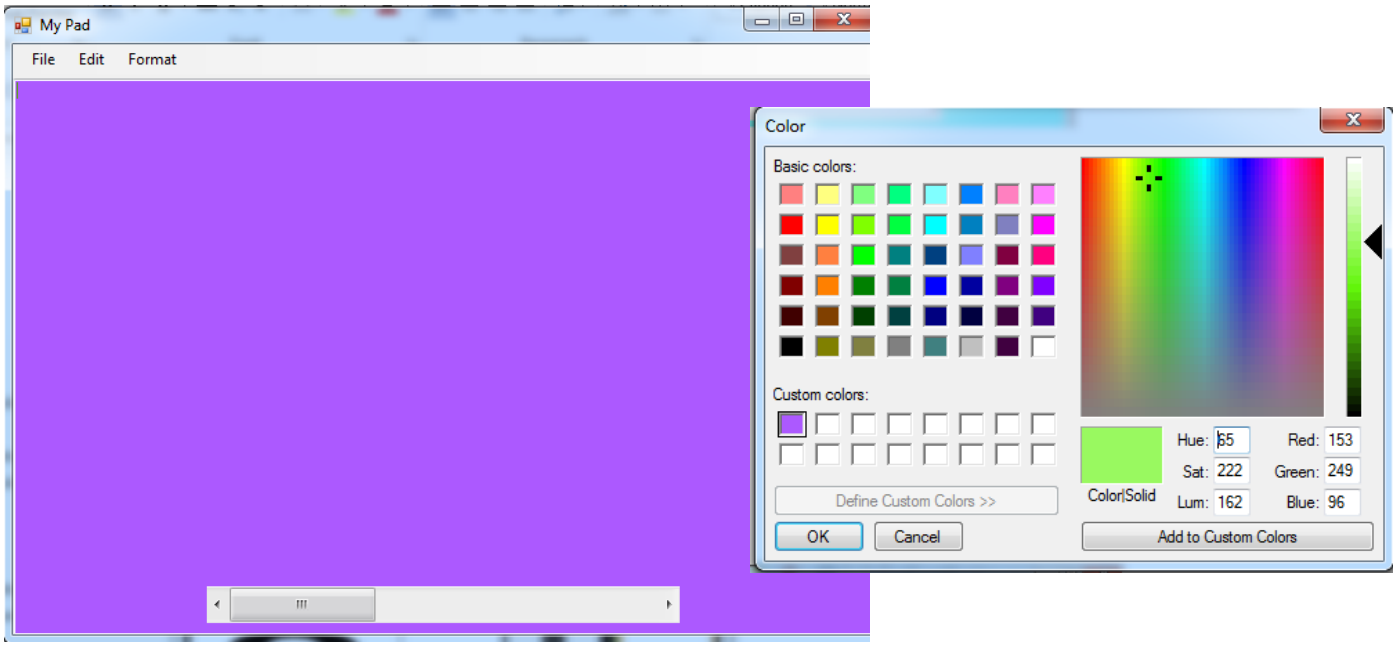


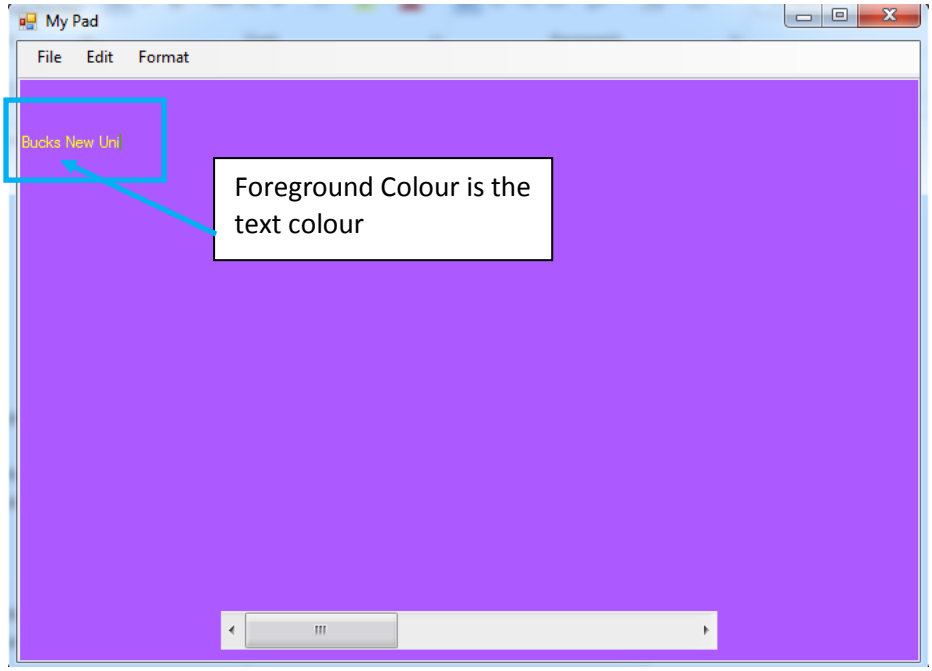
is the my pad program working
t looks like a standard notepad
ram that people would use in a
ows operating system
onment

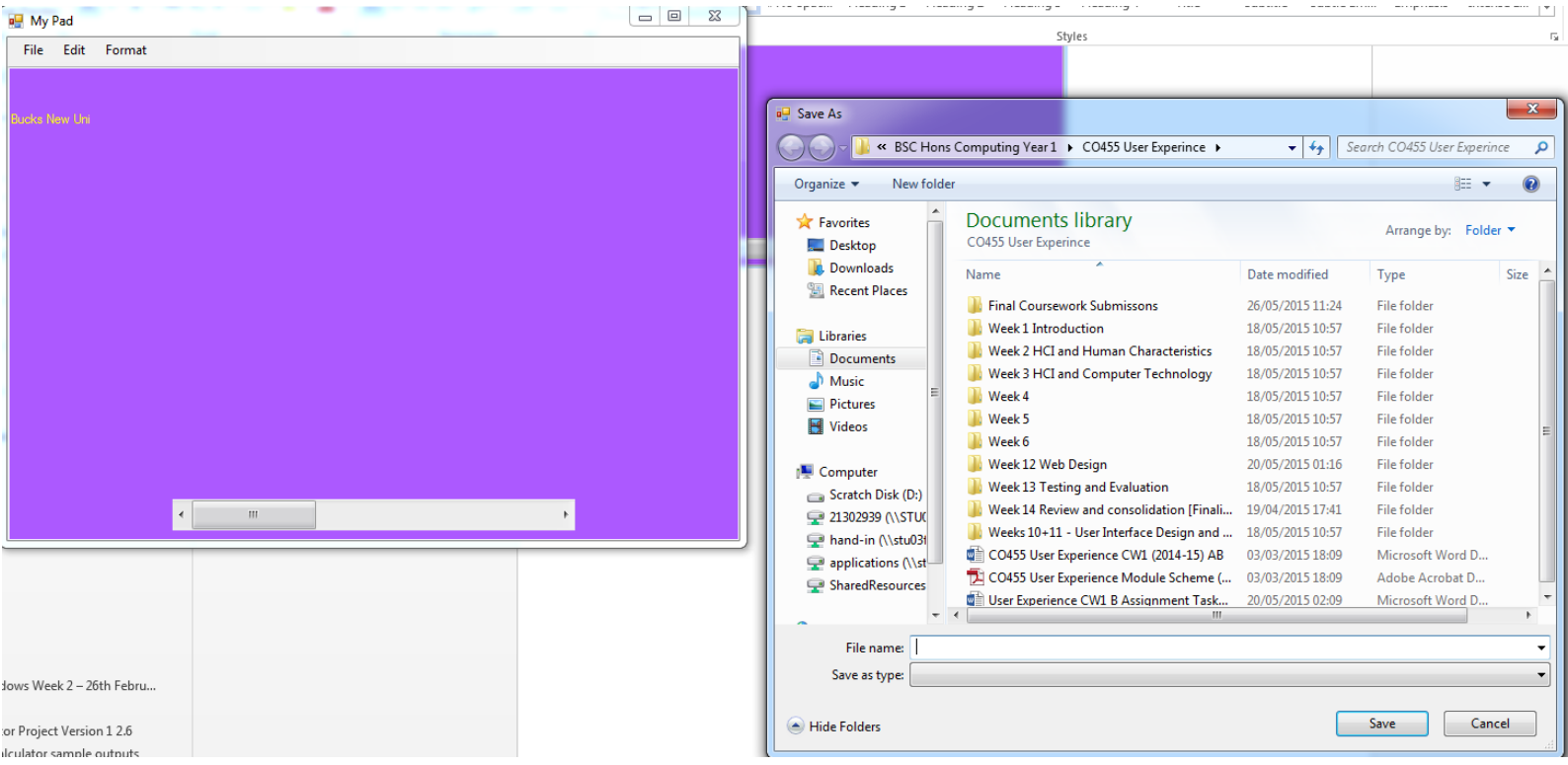


I have changed the
ground colour for the
on to write on

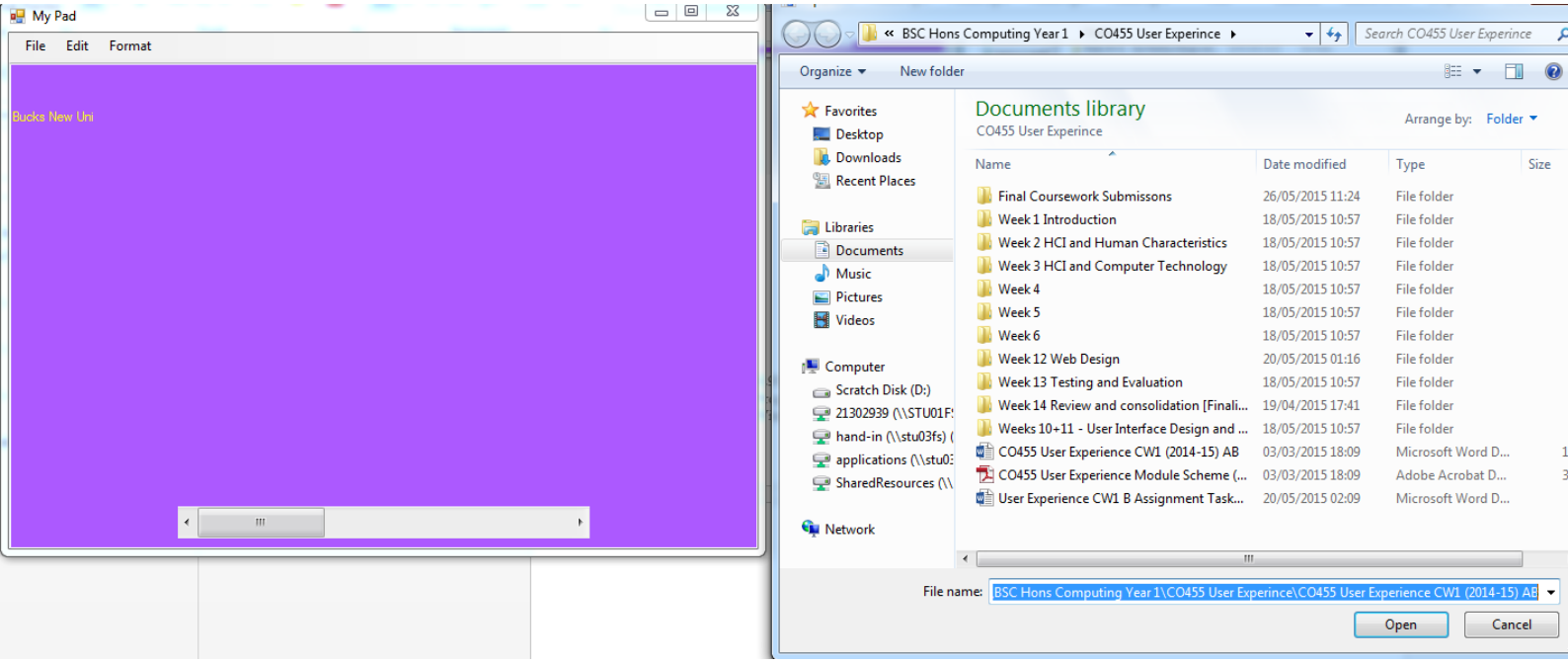

My Pad
File
Edit
Format
Styles
Font
Font:
Microsoft Sans Serif
Microsoft Sans Serif
Microsoft YaHei UI
Minion Pro
Mistral
Modern No. 20
Font style:
Regular
Regular
Oblique
Bold
Bold Oblique
Size:
8
8
9
10
11
12
14
16
OK
Cancel
Effects
Strikeout
Underline
Sample
AaBbYyZz
Script:
Western


My Pad
File
Edit
Format
Color
Basic colors:
Custom colors:
Define Custom Colors >>
OK
Cancel
Color|Solid
Hue: 85
Sat: 222
Lum: 162
Red: 153
Green: 249
Blue: 96
Add to Custom Colors

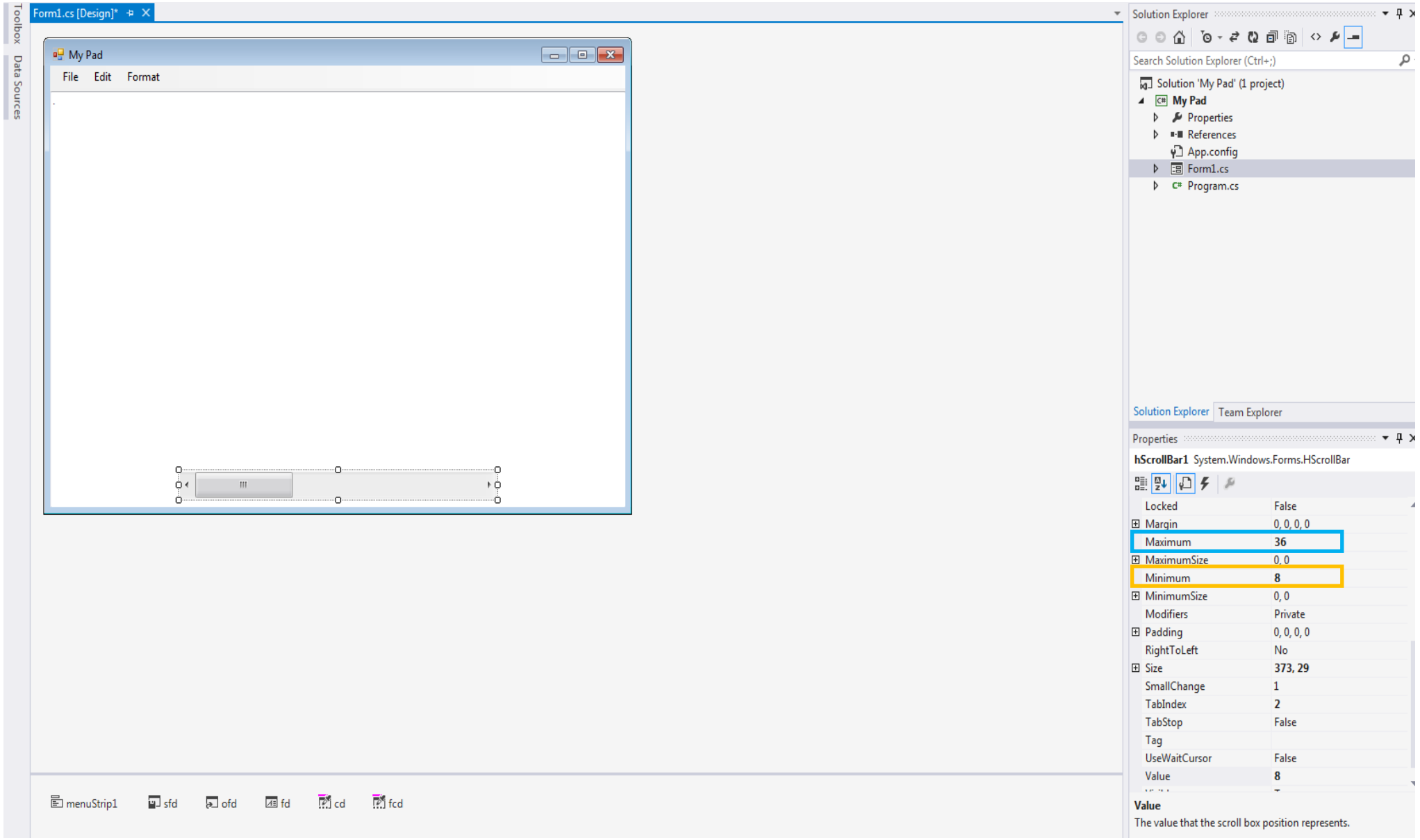




Problem here is the files could not be saved as there is no file extension in the code to save file tyoes I tried to get this to work but the IDE was not recognizing the file types



Here is a file being opened



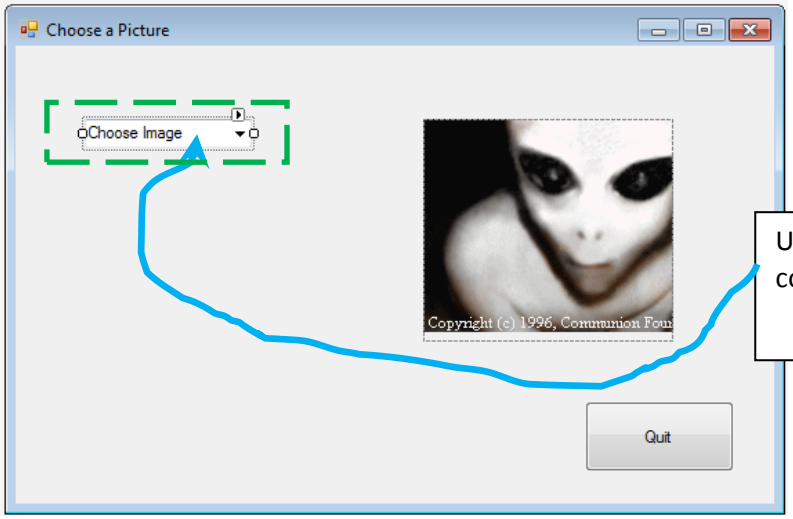
Note – I changed the h scroll bar properties from 8 to 36 this means the font size can only start at 8 and end at 36 as otherwise the text will not be visible to a user on the myPad program however the problem was here actually programming the scroll bar but the size has been set so that's something achived for the task

Source Code

```
namespace My_Pad
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

        }

        private void cutToolStripMenuItem_Click(object sender, EventArgs e) // get
input to use cut


        {
            txtMain.Cut();
        }

        private void copyToolStripMenuItem_Click(object sender, EventArgs e)
// get input to use copy

        {
            txtMain.Copy();
        }

        private void pasteToolStripMenuItem_Click(object sender, EventArgs e)
// get input to use paste

        {
            txtMain.Paste();
        }

        private void undoToolStripMenuItem_Click(object sender, EventArgs e)
// get input to use undo

        {
            txtMain.Undo();
        }

        private void saveToolStripMenuItem_Click(object sender, EventArgs e)
// Bring up the save dialog

        {
            sfd.ShowDialog();
        }

        private void openToolStripMenuItem_Click(object sender, EventArgs e)
// gBring up the open dialog

        {
            ofd.ShowDialog();
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
// exit the program

        {
            Application.Exit();
        }

        private void fontToolStripMenuItem_Click(object sender, EventArgs e)
        {
            fd.ShowDialog();
```

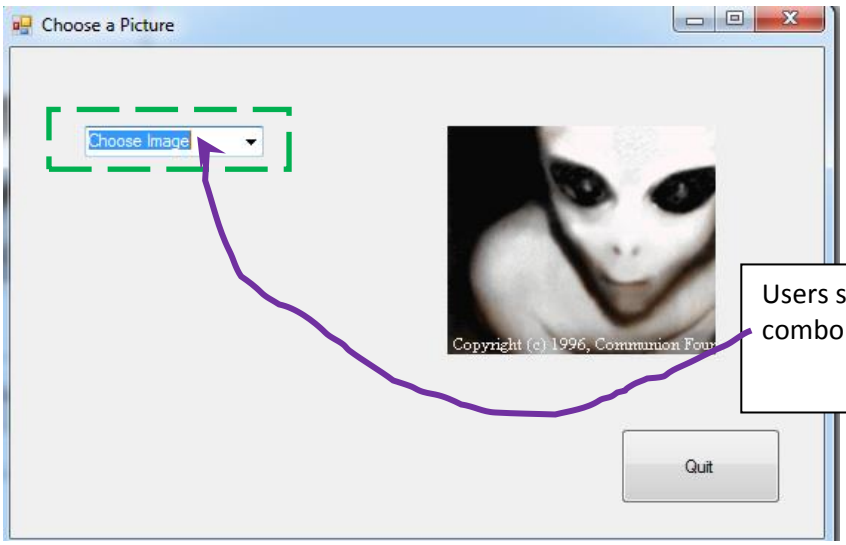
```
            txtMain.Font = fd.Font;
        }

        private void backgroundToolStripMenuItem_Click(object sender, EventArgs e)
        {
            cd.ShowDialog();
            txtMain.BackColor = cd.Color;
        }

        private void foregroungColourToolStripMenuItem_Click(object sender, EventArgs
e)
        {
            fcd.ShowDialog();
            txtMain.ForeColor = fcd.Color;
        }

        private void txtMain_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

5.12 Choose a Picture

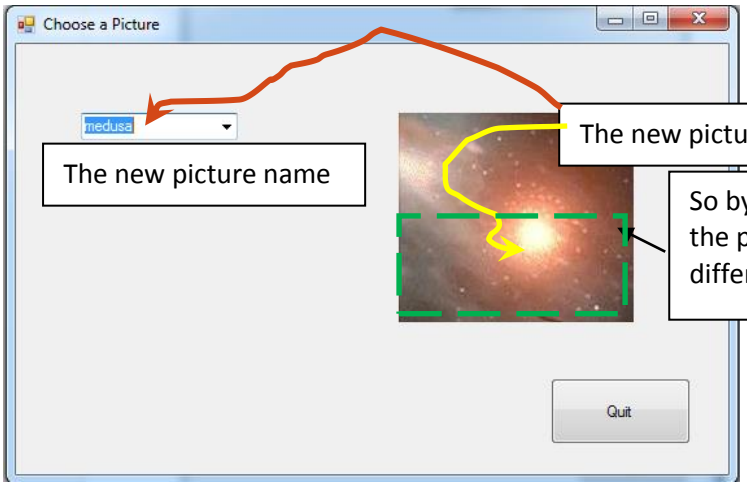


This exercise is a simple exercise to get a user to choose a picture

Users select the choose image combo box to choose an image



Users select the choose image combo box to choose an image



The new picture name

The new picture that was selected

So by selecting an different image the picture box should change to a different image

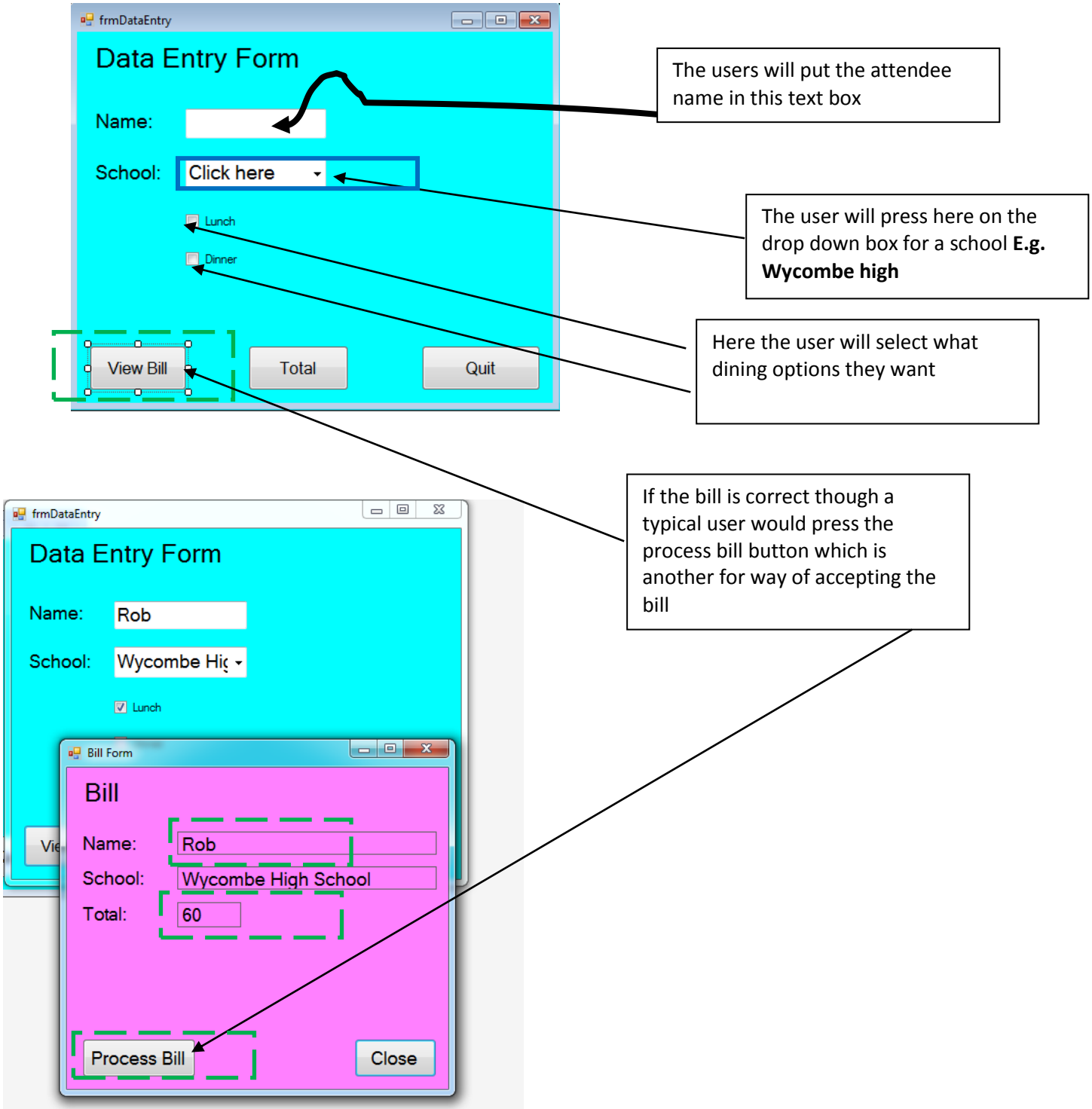
Choose a Picture Source Code

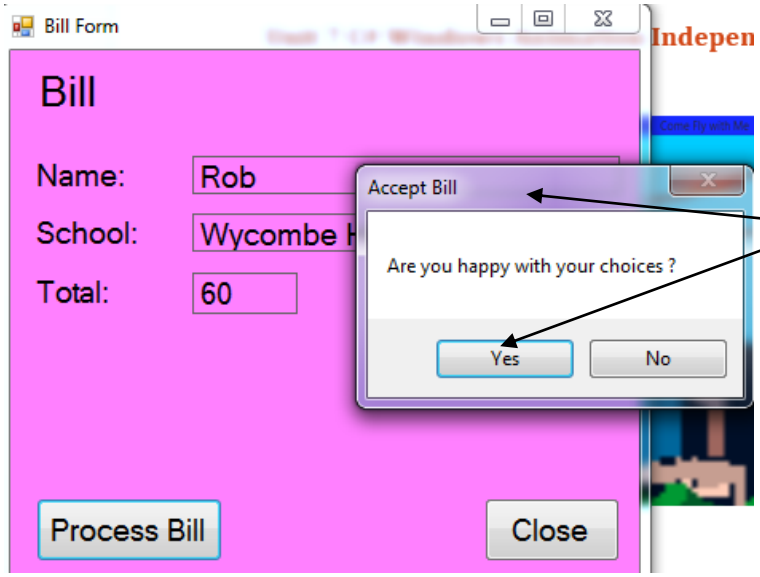
```
namespace _5._12_Choose_a_Picture
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)

// Look up the images stored on file and let the picture box display them

        {
            pbxChoice.Image = Image.FromFile("alien.gif");
            pbxChoice.Image = Image.FromFile("Dragon.gif");
            pbxChoice.Image = Image.FromFile("Godzilla.jpg");
            pbxChoice.Image = Image.FromFile("Hydra.gif");
            pbxChoice.Image = Image.FromFile("kong.gif");
            pbxChoice.Image = Image.FromFile("medusa.jpg");
            pbxChoice.Image = Image.FromFile("munsters.jpg");
            pbxChoice.Image = Image.FromFile("Nessie.jpg");
            pbxChoice.Image = Image.FromFile("piratespider.gif");
            pbxChoice.Image = Image.FromFile("RedHarpy.gif");
            pbxChoice.Image = Image.FromFile("scorpius.jpg");
            pbxChoice.Image = Image.FromFile("Shadow.jpg");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

    }
}
```

C# Windows Week 6 – Muti Form projects

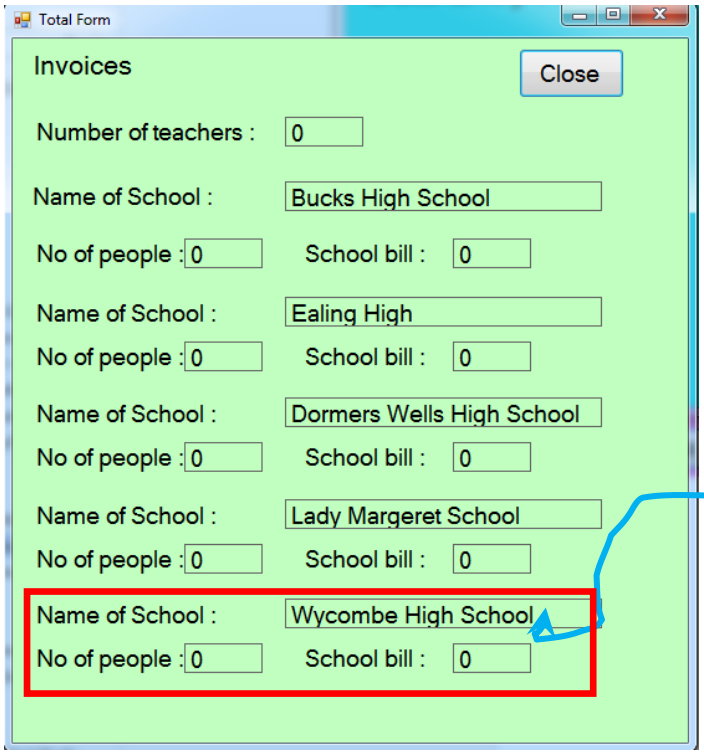
This module of the C# Windows programming will require forms to work together the project is

Program Name -- Education of the 21st Century





This is a message box that asks the user are they happy with the choices they have made they can always say no to change their mind



Now what should have happened here is the bill number should have been updated to 60 pounds for Wycombe high school

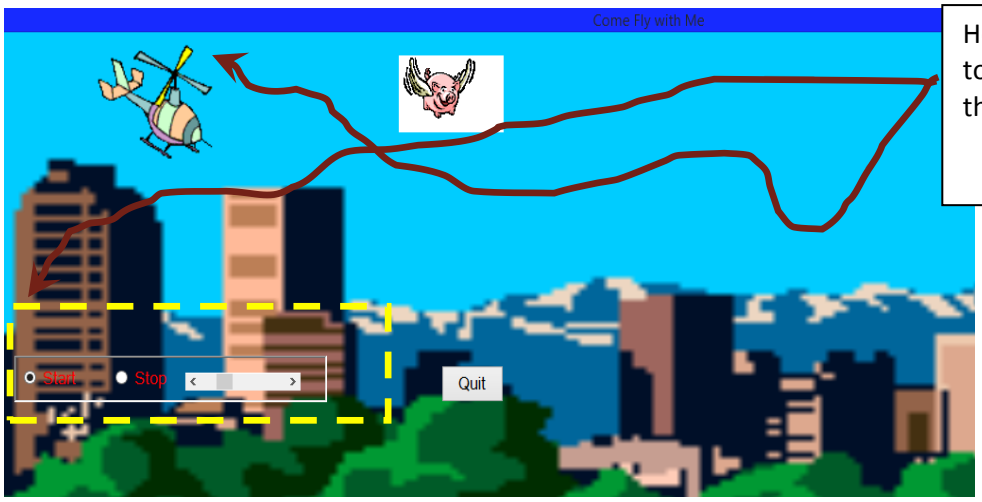
As much as I tried to get this to work I could not

C# Windows Week 7 – Animation

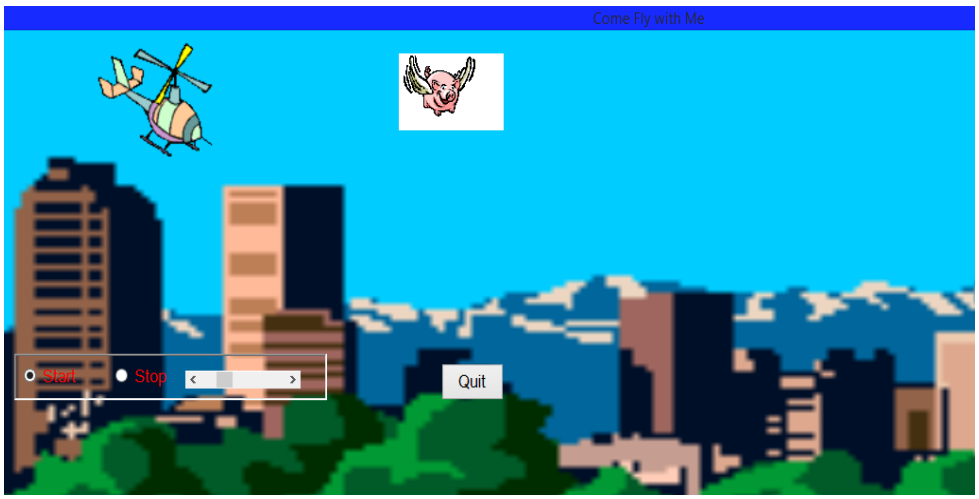
This week I am producing programs that have animation in.

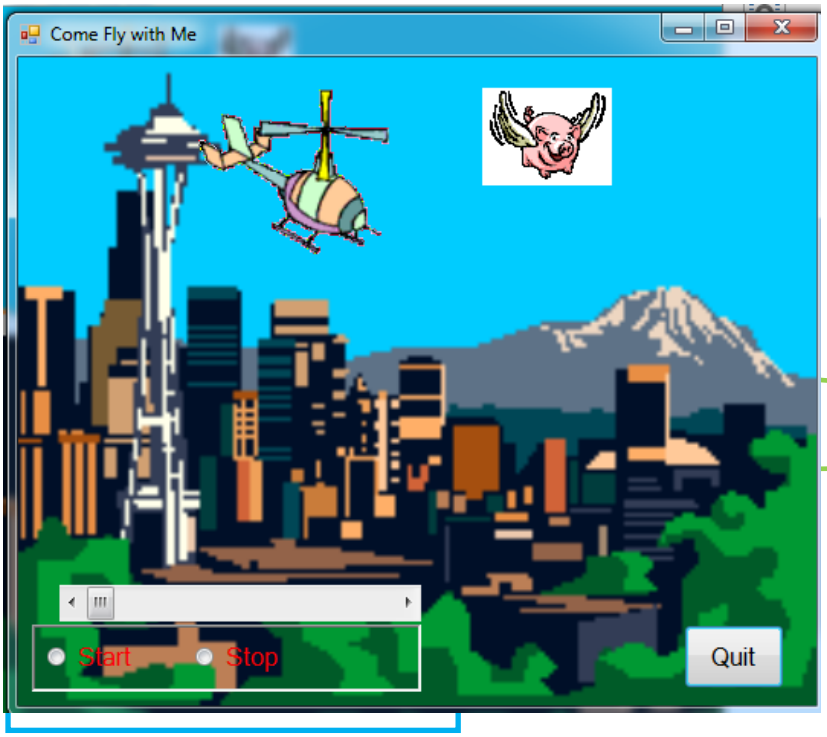
Animation is very useful as it aids users in programs where to go and what they are about to press on such as file to save a document

Unit 7 C# Windows Animation Independent Study



Here I wanted the helicopter to move along the screen with the pigs

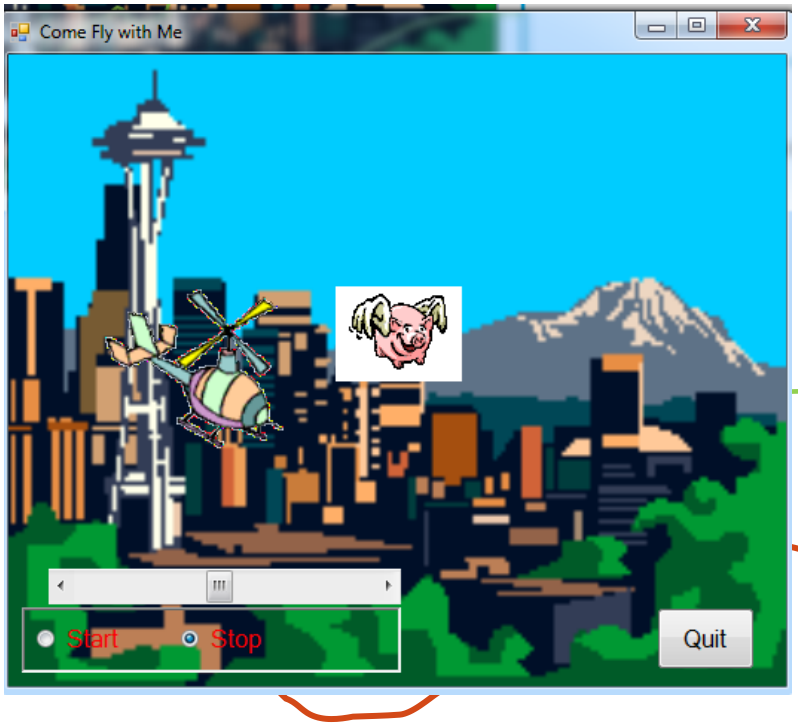




Here the helicopter animation starts and the use of start stop buttons starts and stops it

The scroll bar is there to control the sped

Quit button quits the animation boo



Here the animation is moving and the speed has been adjusted using the scroll bar so users can watch it more easily

Scroll Bar to adjust the speed of the copter and the pig

Source Code

```
//Robert Collcott
//ID:21302939
// Computing
// Helicopter animation
// 2014-2015 Academic Year


namespace Animation
{
    public partial class Form1 : Form
    {
        const int MAX = 4;              // number of images in the array
        Image[] pics = new Image[MAX];  // create array called pics
        Image[] pigs = new Image[MAX];  // create array called pics



        int count = 0;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // load image files into the pics and pigs arrays
            for (int i = 0; i < MAX; i++)
            {
                pics[i] = Image.FromFile("copter" + i + ".gif");
                pigs[i] = Image.FromFile("pig" + i + ".gif");
            }
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            pbxMoving.Image = pics[count];  // show an image from the array
            pbxMoving2.Image = pigs[count];

            count++;                        // move on to next pic
            if (count == MAX) count = 0;    // reset to zero

            if(pbxMoving.Left>=this.Width)
            {
                pbxMoving.Left = 0;
            }
            else if(pbxMoving.Top>=this.Height)
            {
                pbxMoving.Top = 0;
            }
            else
            {
                pbxMoving.Left = pbxMoving.Left + 3; // to start motion towards left
                pbxMoving.Top = pbxMoving.Top + 1; // to start motion downwards
            }



            pbxMoving2.Image = pigs[count];  // show an image from the array

            count++;                        // move on to next pic
```

```
            if (count == MAX) count = 0;    // reset to zero

            if (pbxMoving.Left >= this.Width)
            {
                pbxMoving2.Left = 0;
            }
            else if (pbxMoving2.Top >= this.Height)
            {
                pbxMoving2.Top = 0;
            }
            else
            {
                pbxMoving2.Left = pbxMoving2.Left + 10; // to start motion towards
left
                pbxMoving2.Top = pbxMoving2.Top + 1; // to start motion downwards
            }




        }

        private void rbStart_CheckedChanged(object sender, EventArgs e)
        {
            timer1.Interval = 10;
            timer1.Enabled = true;                                              //
Start the timer
        }

        private void rbStop_CheckedChanged(object sender, EventArgs e)          //
stop the timer
        {
            timer1.Enabled = false;
        }

        private void btnQuit_Click(object sender, EventArgs e)          // This
function exits the program
        {
            Application.Exit();
        }

        private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)       //
This function controls the speed of the flight of the helipector and the pig
        {
            int amount = hsbSpeed.Value + 1;        // The amount of speed of the
copter and pig
            timer1.Interval = 100/ amount;             // This is the speed of the
helicopter
        }

    }
}
```

Unit 8 Graphics Independent Study

Here you will see the my independent study tasks for Unit 8 windows C# programming which is based on Graphics

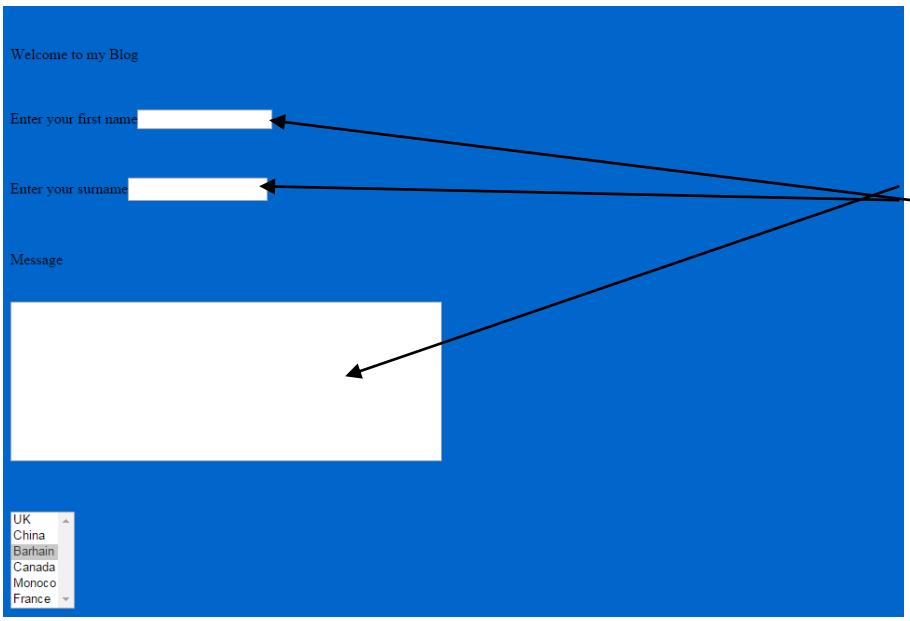
8.9 Pick a Graphic

Had a go of trying to create the program but could not get it to work

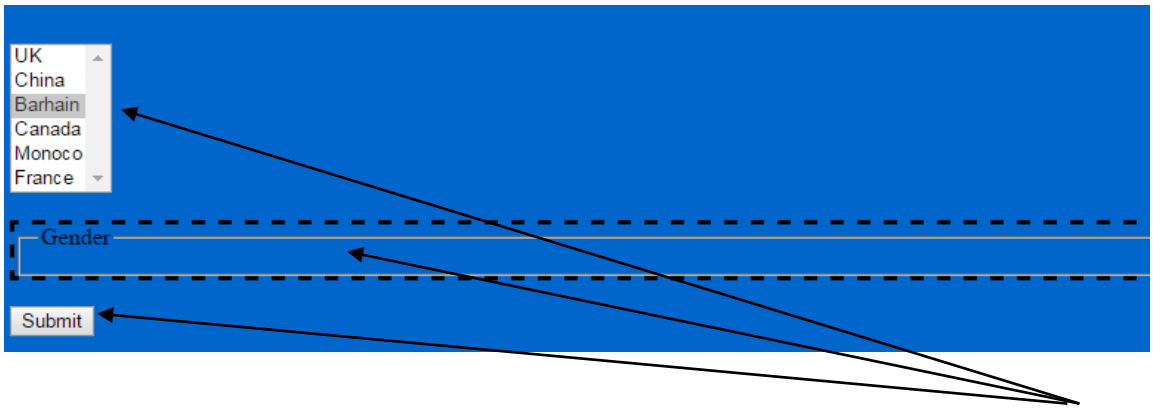
8.10 Concentric Circles

Had a go of trying to create the program but could not get it to work

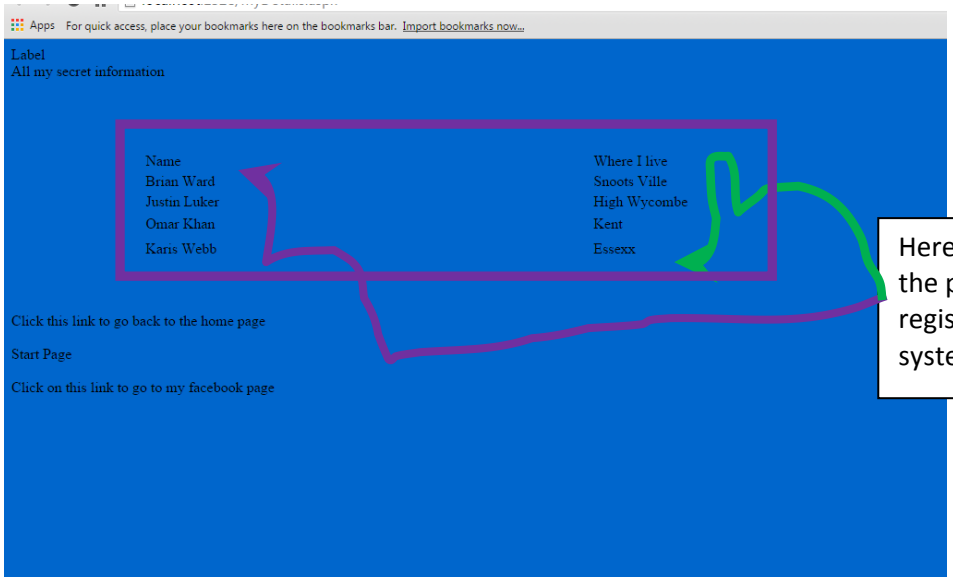
This week I am going to use Visual Studio web site tools of asp.net



Here is the home web page I created named index this is where a typical user registers for the blog



Here a user will select what country of origin that they are from and the gender as well and submit



Here is all the details of all the paper that are registered within my blog system

Home Page Source Code

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Robert&#39;s Blog</title>
</head>
<body style="background-color: #0066CC">
    <p>
        <br />
    </p>
    <form id="form1" runat="server">
    <p>
        <asp:Label ID="Label3" runat="server" Text="Welcome to my Blog"></asp:Label>
        </p>
        <p>
            <asp:HyperLink ID="HyperLink3" runat="server"
NavigateUrl="~/myDetails.aspx">Personal details</asp:HyperLink>
 
            <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/History.aspx">Website History</asp:HyperLink>
  
            <asp:HyperLink ID="HyperLink4" runat="server"
NavigateUrl="~/Quiz.aspx">Quiz</asp:HyperLink>
        </p>
    <p>
        Enter your first name<asp:TextBox ID="TextBox1" runat="server"
Width="141px"></asp:TextBox>
        </p>
    <div aria-dropeffect="none">

        <br />
        <br />
        Enter your surname<asp:TextBox ID="TextBox2" runat="server" Height="19px"
Width="146px"></asp:TextBox>
        <br />
        <br />
        <br />
        <br />
        <asp:Label ID="Label2" runat="server" Text="Message"></asp:Label>
        <br />
        <br />
        <br />
        <asp:TextBox ID="TextBox3" runat="server" Height="165px"
Width="459px"></asp:TextBox>
        <br />
        <br />
        <br />
        <br />
        <asp:ListBox ID="ListBox1" runat="server" Rows="6">
            <asp:ListItem>UK</asp:ListItem>
            <asp:ListItem>China</asp:ListItem>
            <asp:ListItem>Barhain</asp:ListItem>
            <asp:ListItem>Canada</asp:ListItem>
            <asp:ListItem>Monoco</asp:ListItem>
            <asp:ListItem>France</asp:ListItem>
        </asp:ListBox>
        <br />
        <br />
        <asp:RadioButton ID="Male" runat="server" />
        <asp:RadioButton ID="Female" runat="server" />
```

```
        <asp:Panel ID="Panel1" runat="server" BorderStyle="Dashed"
GroupingText="Gender" ToolTip="Choose your Sex here">
        </asp:Panel>
        <asp:Label ID="lblText" runat="server" Text="Welcome"></asp:Label>
        <br />
        <br />
        <asp:Button ID="btnSubmit" runat="server" OnClick="btnSubmit_Click"
Text="Submit" />
        <br />
        <br />

    </div>
    </form>
</body>
</html>
```

Blog Members details page source code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="myDetails.aspx.cs"
Inherits="myDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            width: 80%;
        }
        .auto-style2 {
            height: 23px;
        }
    </style>
</head>
<body>
    <body style="background-color: #0066CC">
    <form id="form1" runat="server">
    <div>

        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

    </div>
        <asp:Label ID="Label2" runat="server" Text="All my secret
information"></asp:Label>
        <br />
        <br />
        <br />
        <br />
        <table align="center" class="auto-style1">
            <tr>
                <td colspan="3"> </td>
            </tr>
            <tr>
                <td>Name</td>
                <td>Where I live</td>
                <td> </td>
            </tr>
            <tr>
                <td>Brian Ward</td>
                <td>Snoots Ville</td>
                <td> </td>
            </tr>
            <tr>
                <td>Justin Luker</td>
                <td>High Wycombe</td>
                <td> </td>
            </tr>
            <tr>
                <td class="auto-style2">Omar Khan</td>
                <td class="auto-style2">Kent</td>
                <td class="auto-style2"></td>
            </tr>
            <tr>
                <td class="auto-style2">Karis Webb</td>
                <td class="auto-style2">Essexx</td>
                <td class="auto-style2"></td>
            </tr>
        </table>
        <br />
        <br />
```

```
        <br />
        Click this link to go back to the home page<br />
        <br />
        <asp:HyperLink ID="HyperLink1" runat="server">Start Page</asp:HyperLink>
        <br />
        <br />
        Click on this link to go to my facebook page<br />
    </form>
</body>
</html>
```

For this C# Windows unit I had to create 2 simple web pages

The first one I did of these was a quiz page

Quiz Page source code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Quiz.aspx.cs" Inherits="Quiz"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>

        Here is a short quiz to test your knowledge about blog systems
        <br />
        <br />

    </div>
        <p>
            1. What is a blog system?
        </p>
        <p>
            <asp:RadioButton ID="RadioButton1" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton2" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton3" runat="server" />
        </p>
        <p>
             </p>
        <p>
            2. How long have blogs been around? </p>
        <asp:RadioButton ID="RadioButton4" runat="server" />
        <br />
        <br />
        <asp:RadioButton ID="RadioButton5" runat="server" />
        <br />
        <br />
        <asp:RadioButton ID="RadioButton6" runat="server"
OnCheckedChanged="RadioButton6_CheckedChanged" />
        <br />
        <p>
            3. Can you setup your blog to have rules?</p>
        <p>
            <asp:RadioButton ID="RadioButton7" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton8" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton9" runat="server"
OnCheckedChanged="RadioButton6_CheckedChanged" />
        </p>
        <p>
```

```
             </p>
        <p>
            4. How many people in millions use blogs?</p>
        <p>
            <asp:RadioButton ID="RadioButton10" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton11" runat="server"
OnCheckedChanged="RadioButton11_CheckedChanged" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton12" runat="server"
OnCheckedChanged="RadioButton6_CheckedChanged" />
        </p>
        <p>
             </p>
        <p>
            5. Can blogs be also search engines?</p>
        <p>
            <asp:RadioButton ID="RadioButton13" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton14" runat="server" />
        </p>
        <p>
            <asp:RadioButton ID="RadioButton15" runat="server" />
        </p>
    </form>
    <p>
 
    </p>
    <p>
         </p>
</body>
</html>
```

Note – Radio buttons are used on this page

Here is a short quiz to test your knowledge about blog systems

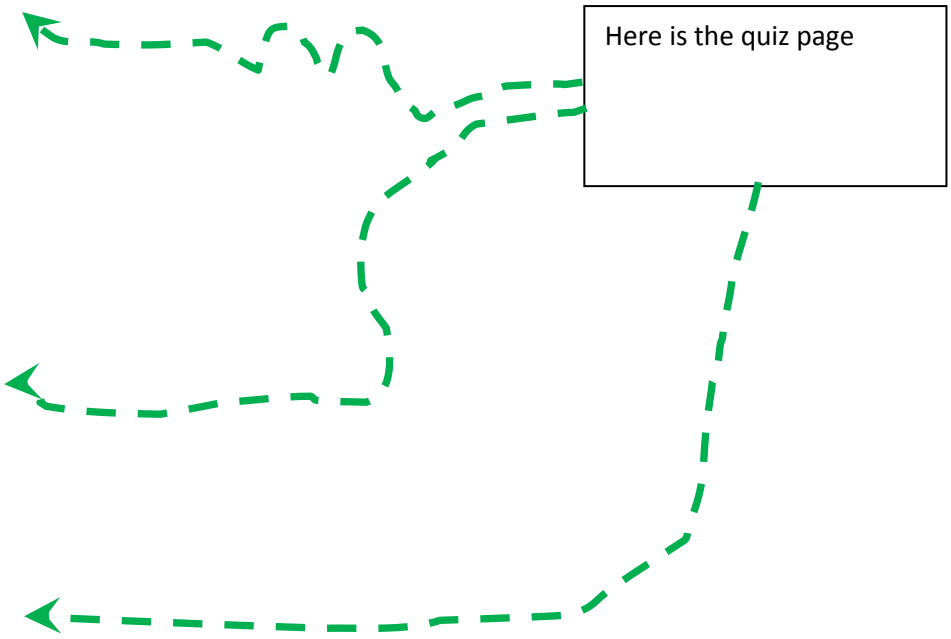
1. What is a blog system?

2. How long have blogs been around?

3. Can you setup your blog to have rules?

4. How many people in millions use blogs?

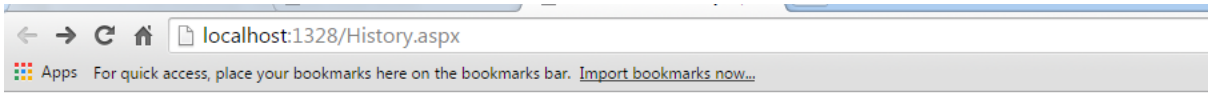
5. Can blogs be also search engines?



History page source code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="History.aspx.cs"
Inherits="History" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body bg.color="295BE4">


    <p>

        <br />
        History of  us
    </p>
    <form id="form1" runat="server">
    <div>

    </div>
    <p>
        We were founded in 2013 and have 3 milion vistors overall </p>
        <p>
             </p>
        <p>
            New for 2014 is going to be a place where you can upload special photo
memories and make videos with them</p>
        <p>
             </p>
        <p>
            <asp:HyperLink runat="server" NavigateUrl="~/Default.aspx">Home
Page</asp:HyperLink>
            <asp:HyperLink runat="server" NavigateUrl="~/myDetails.aspx">Home
Page</asp:HyperLink>
            <asp:HyperLink runat="server" NavigateUrl="~/Quiz.aspx">Home
Page</asp:HyperLink>

        </p>
    </form>
    </body>
</html>
```

Sample 2nd Page Created Brief History of my blog



History of us

We were founded in 2013 and have 3 milion vistors overall

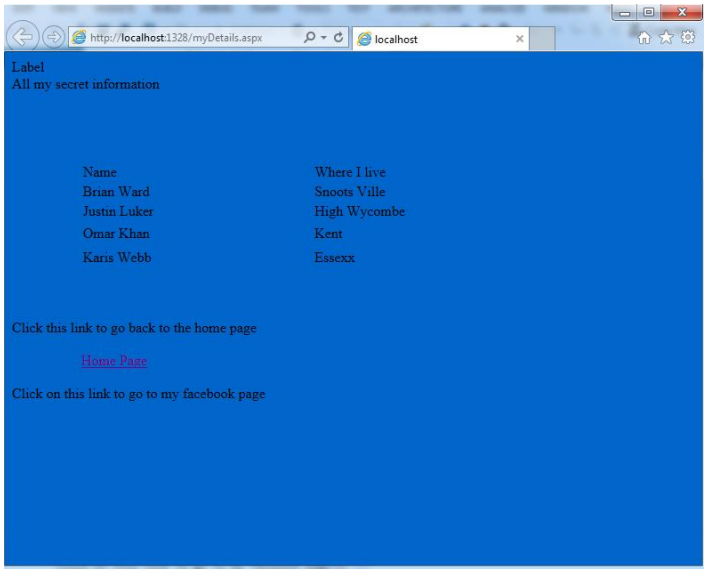
New for 2014 is going to be a place where you can upload special photo memories and make videos with them

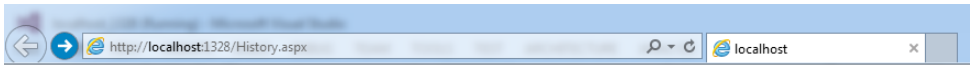
Home Page Home Page Home Page

I was unable to produce a full site with pictures and backgrounds but I got the majority of the functionality in such as links and text

If I had more time I could have built the whole website by putting validation in and selecting one answer per question not 3 as 1 has to be right and the other 2 wrong

Showing links between the different pages



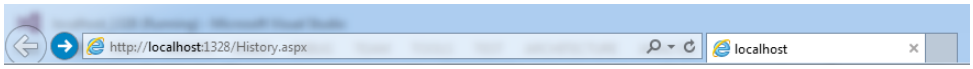


History of us

We were founded in 2013 and have 3 milion vistors overall

New for 2014 is going to be a place where you can upload special photo memories and make videos with them

Home Page Home Page Home Page



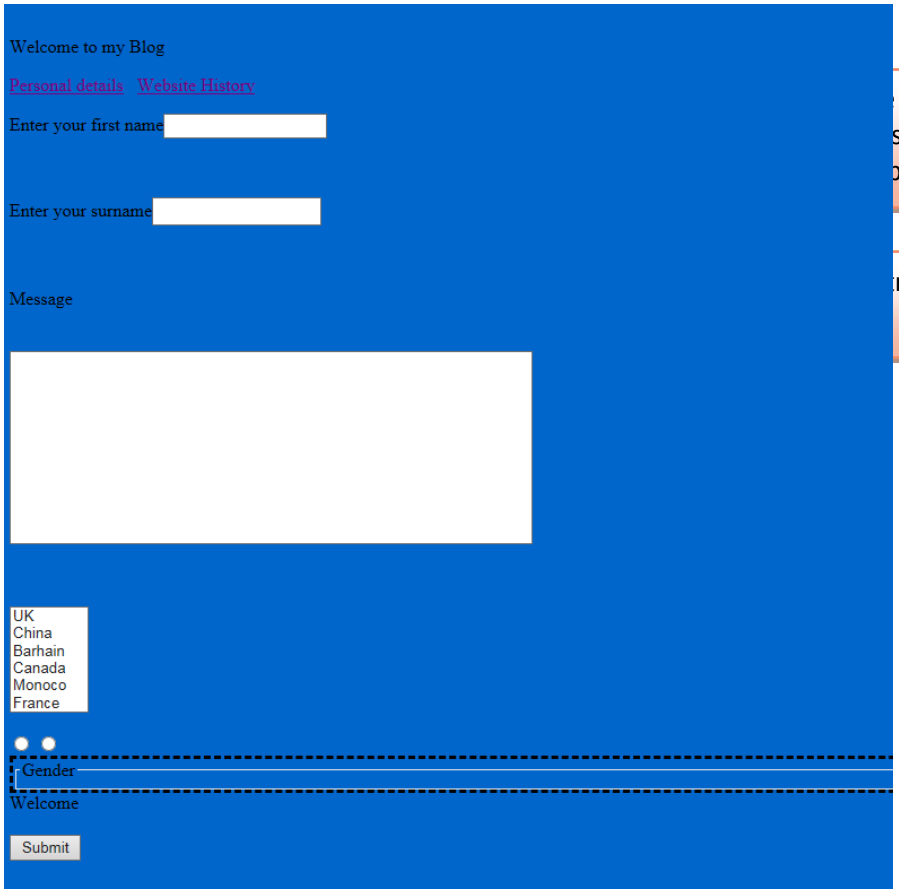
History of us

We were founded in 2013 and have 3 milion vistors overall

New for 2014 is going to be a place where you can upload special photo memories and make videos with them

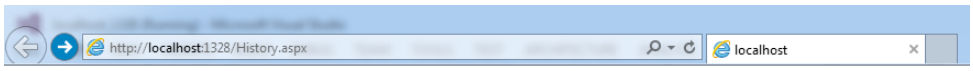
Home Page Home Page Home Page

The history page can link to the registration screen



registration screen can also link to
sonal details of blog members and
bsite history page

ration screen



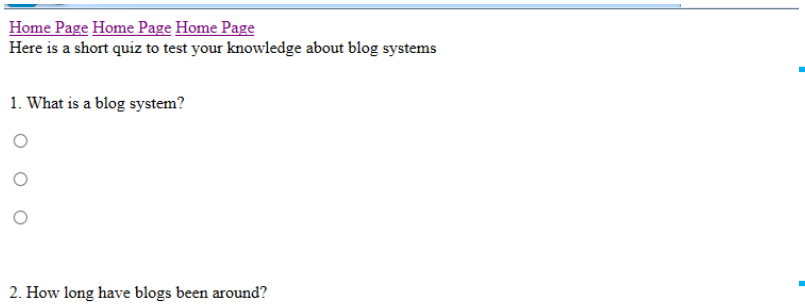
History of us

We were founded in 2013 and have 3 milion vistors overall

New for 2014 is going to be a place where you can upload special photo memories and make videos with them

Home Page Home Page Home Page

history page goes to
uiz page



Algorithm Logic

Declare variables

Declare brush colours

Display program screen

Get user input for drawing lines in all directions

Draw line

User is finished

Press Escape to bring up the message box

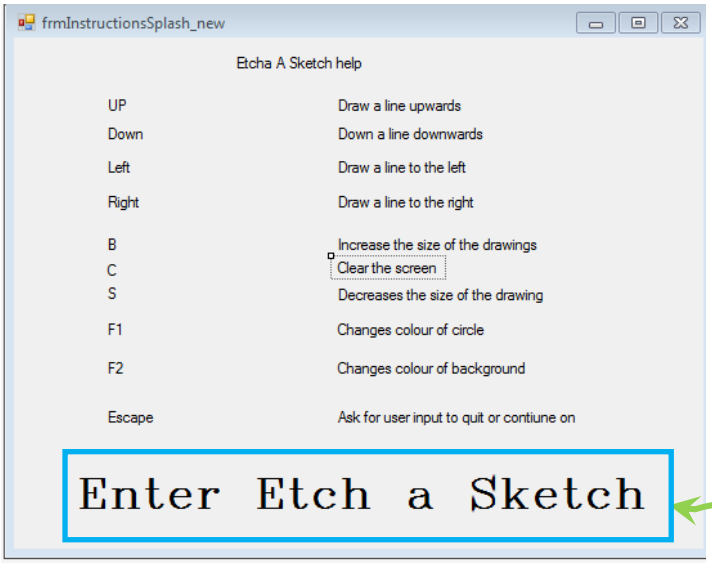
Display message box

Get user input from the user to quit the program or stay playing

If user input is yes then quit the program

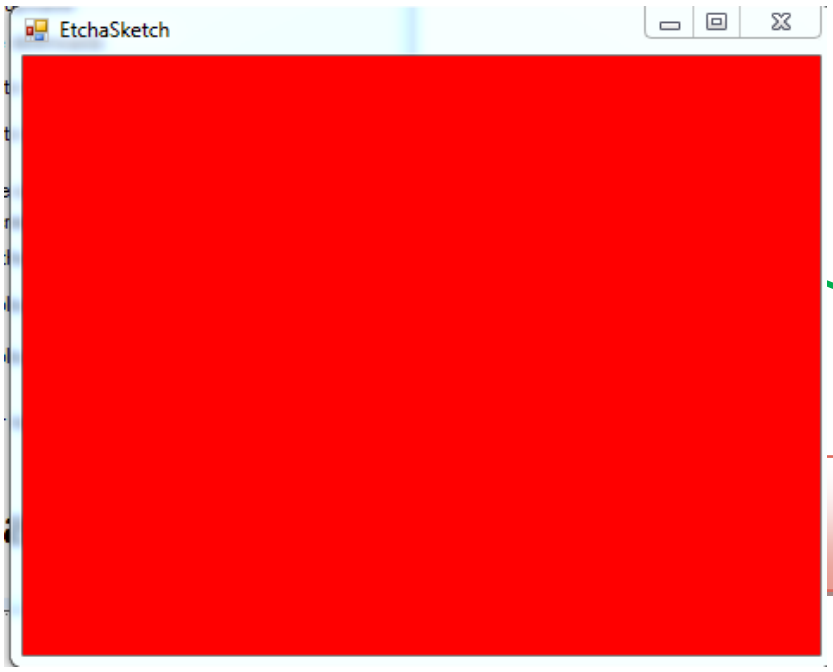
If user input is no carry on playing the Etch a Sketch game

C# Windows Etch a Sketch Sample program outputs

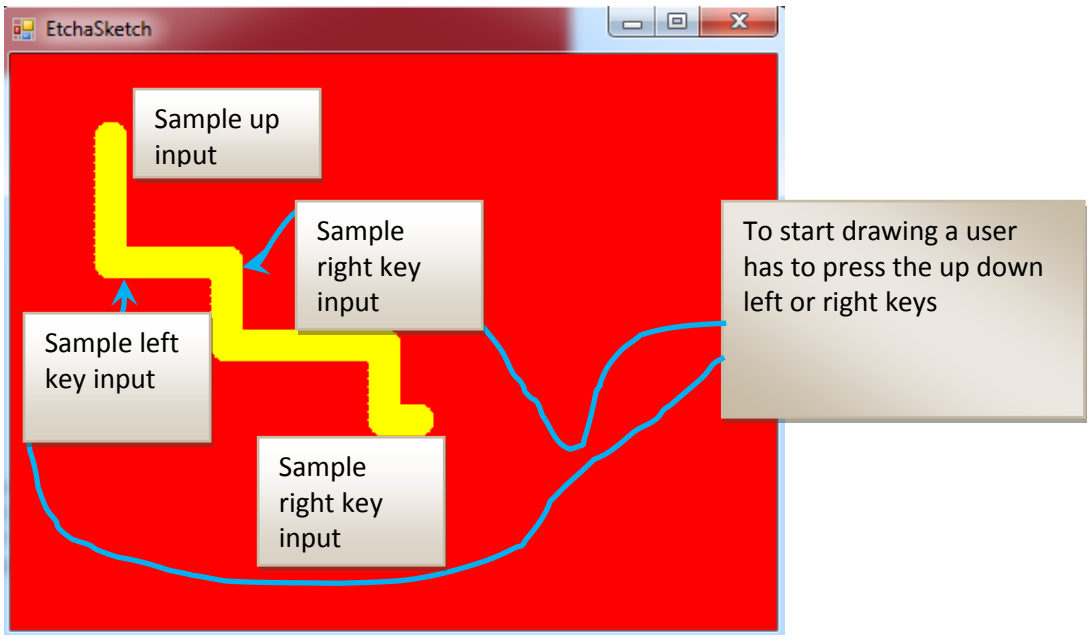


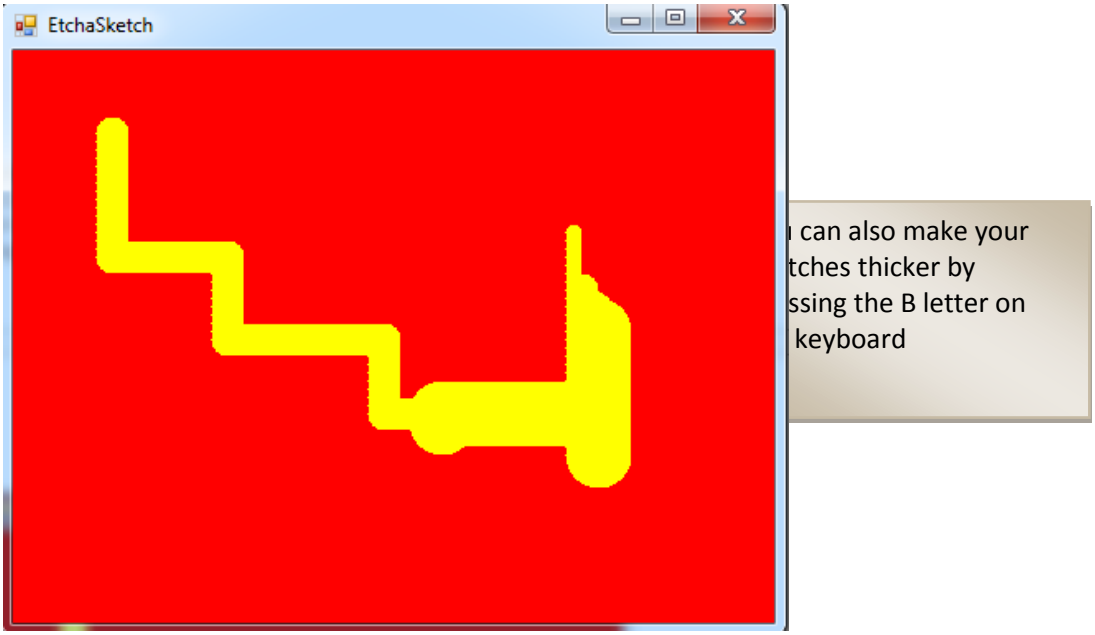
Here are the instructions for users to use the Etch a Sketch program

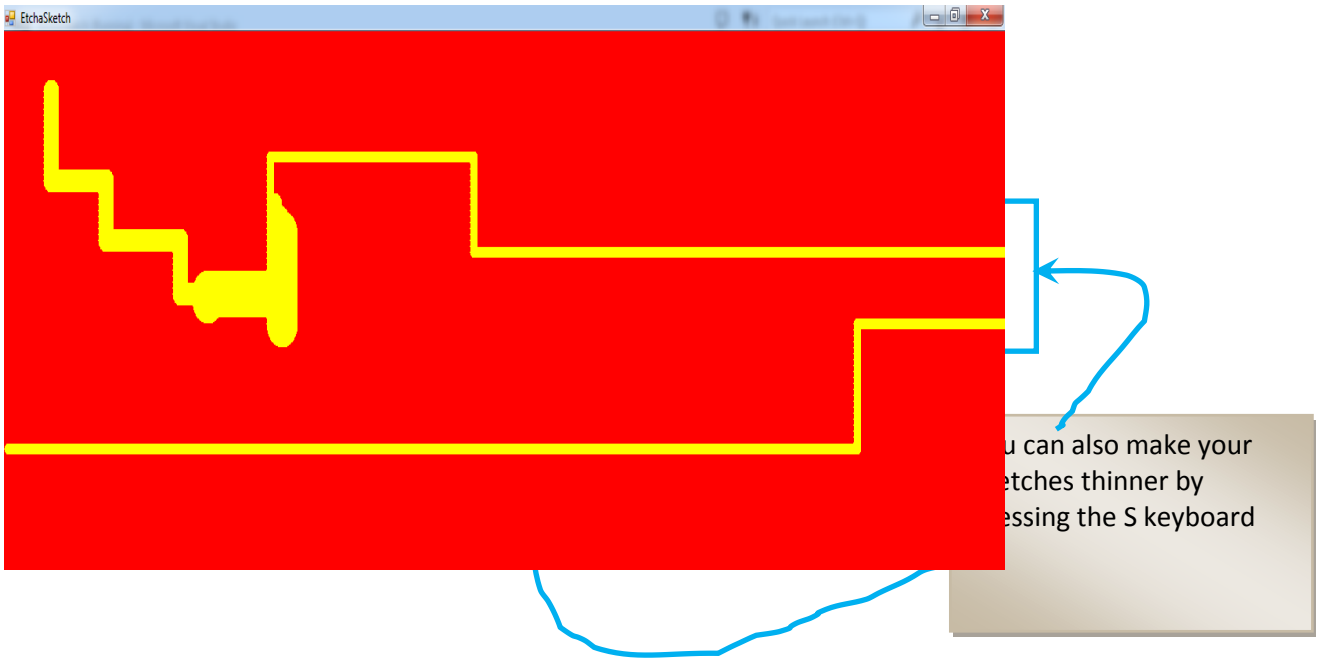
Here are the instructions for users to use the Etch a Sketch program

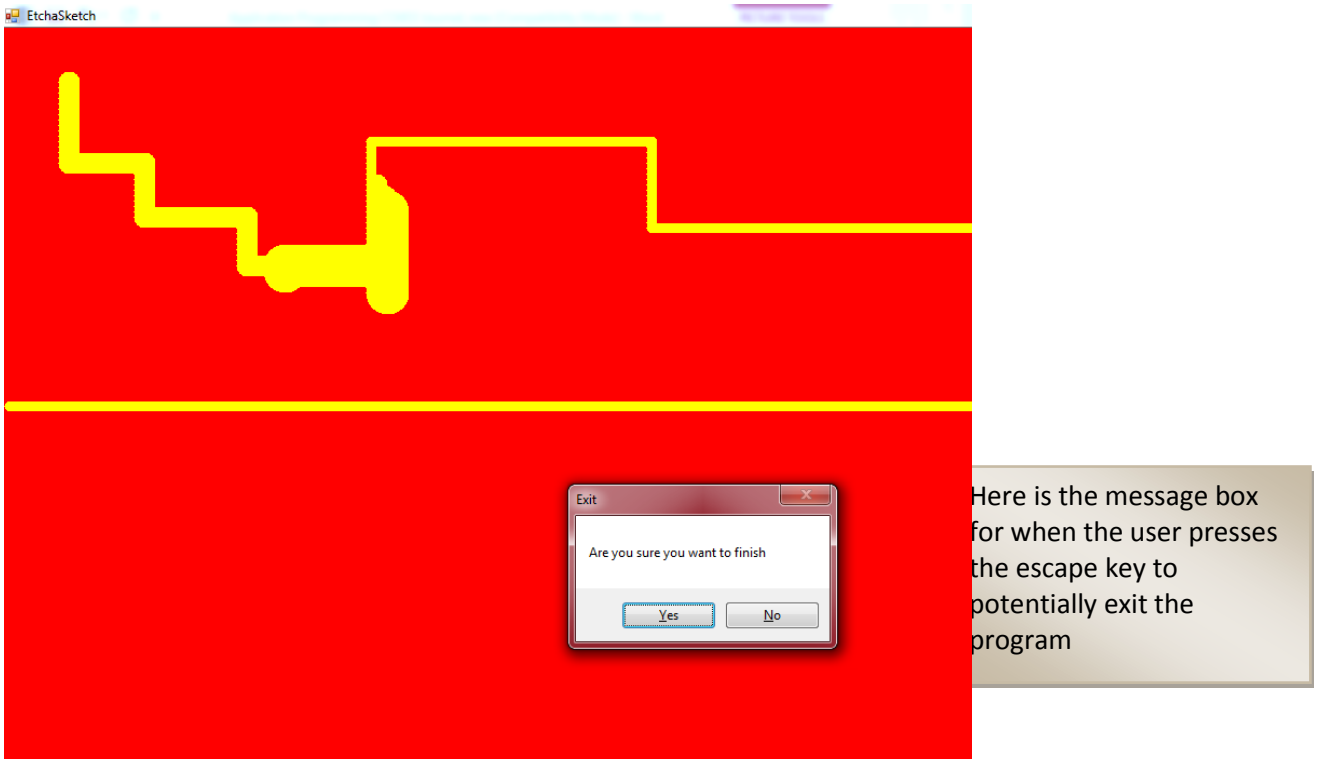


When the user clicks on Enter Etch a sketch on the splash screen the main program will then start with a red background


EtchaSketch
Sample up input
Sample right key input
Sample left key input
Sample right key input
To start drawing a user has to press the up down left or right keys


EtchaSketch
can also make your
tches thicker by
ssing the B letter on
keyboard


EtchaSketch
u can also make your
tches thinner by
ssing the S keyboard


EtchaSketch
Exit
Are you sure you want to finish
Yes
No
Here is the message box
for when the user presses
the escape key to
potentially exit the
program

Escape key brings up the message box

S makes the sketch thinner

B makes the sketch thicker

Main program Source Code

```
namespace EtchaSketch
{
    // Robert Collcott
    // ID 21302939
    // Computing
    // 28th May 2015
    // Etch A Sketch C# Windows Programming Project



    public partial class Form1 : Form
    {
        int x = 50, y = 50, size = 20;     // This is the vairable that delcares the
size
        int maximum = 100, minimum = 10;
        int red;                                                    // red colour
        int yellow;                                                 // yellow colour
        int green;                                                  // green colour
        Random rand = new Random();                                 // random

        Bitmap bm;

        public Form1()
        {
            frmInstructionsSplash_new SScreen = new frmInstructionsSplash_new();
// Get form details of the new splash screen
            SScreen.ShowDialog();
// Shows the splash screen first
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            bm = new Bitmap(this.Width, this.Height);   // create a form-size bitmap
            Graphics g = Graphics.FromImage(bm);        // get a graphic object for
the bitmap
            this.BackgroundImage = bm;                  // use the bitmap as the form
background

        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Brush myBrush = new SolidBrush(Color.FromArgb(red, green, yellow));
            Graphics g = Graphics.FromImage(bm);        // get a graphic object for
the bitmap
            g.FillEllipse(Brushes.Yellow, x, y, size, size);                // put a
circle in the bitmap and draw it yellow.
        }
        protected override bool ProcessCmdKey(ref Message msg, Keys keyData)
        {
            string input;
            input = keyData.ToString();
            Graphics g = Graphics.FromImage(bm);
            if (input == "Up")                          // get input for an up key
press
            {
                y = y - 5;
                Refresh();

            }
            else if (input == "Down")               // get input for a down key press
            {
```

```
                y = y + 5;
                Refresh();
            }
            else if (input == "Left")               // get input for a left key press
            {
                x = x - 5;
                if (x < 0)
                {
                    x = 1670;

                }

                Refresh();
            }
            else if (input == "Right")             // get input for a right  key press
            {
                x = x + 5;
                Refresh();
            }
            else if (input == "UP" && input == "Left")
            {
                y = y - 5;
                x = x - 5;
                Refresh();
            }
            else if (input == "UP" && input == "Right")
            {
                y = y - 5;
                x = x + 5;
                Refresh();
            }
            else if (input == "Down" && input == "Left")
            {
                y = y + 5;
                x = x - 5;
                Refresh();
            }
            else if (input == "Down" && input == "Right")
            {
                y = y + 5;
                x = x + 5;
                Refresh();
            }
            else if (input == "C")
            {
                g.Clear(this.BackColor);
                Refresh();
            }
            else if (input == "Escape")
            {
                DialogResult responce;
                responce = MessageBox.Show("Are you sure you want to finish", "Exit",
MessageBoxButtons.YesNo);             // get input to exit or stay on in the program
                if(responce == DialogResult.Yes)            // Response by user is
user = Yes // Exit the program
                {
                    Application.Exit();
                }
            }
            else if (input == "B")              // Increases the size of the drawing
            {
                if (size < maximum)
                {
                    size = size + 10;
                }
```

```
                Refresh();                      // Referesh the screen after this key
press
            }
            else if (input == "S")
            {
                if (size > minimum)
                {
                    size = size - 10;
                }
                Refresh();                      // Referesh the screen after this key
press
            }
            else if (input == "F1")                 //myBrush.Color =
Color.FromArgb(red, green, blue);
            {
                red = rand.Next(0, 255);
                green = rand.Next(0, 255);
                yellow = rand.Next(0, 255);
                g.FillEllipse(Brushes.Blue, x, y, size, size);
                g.FillEllipse(Brushes.Yellow, x, y, size, size);

                Refresh();
                return true;
            }
            return false;    // return true if key processed, otherwise false
        }

    }
}//Form2.cs
```

Splash Screen Source Code

This code bellows to the splash screen which comes up first so users have an idea of how to work the program

```
namespace EtchaSketch
{
    public partial class frmInstructionsSplash_new : Form
    {
        public frmInstructionsSplash_new()
        {
            InitializeComponent();
        }

        private void label20_Click(object sender, EventArgs e)
        {
            this.Close();          // this closes the splash screen down and enters
the main program

        }
    }
}
```

C# Windows Project Etch and Sketch Evaluation

This was one of the hardest projects I have ever had to tackle since coming and doing programming at Buckinghamshire New University

There were a lot of key presses that I had to program and this I had massive problems with as I had to program lots of different functions in order for the keys to work and refreshes each time so the input and the program did not get confused by what key I was pressing

Another problem I had was trying to get the correct syntax to get the lines drawn correctly this I eventually solved by looking the correct syntax on the MSDN **Microsoft Developer network site** in the C# format as this gave me the correct way if typing the code and told me extensively what goes were in the code

One of the other problems I had was trying to get the background colour changed this was meant to happen by a user pressing F2 I looked al forums such as MSDN but when trying the code to fit in line with mine It did not work

One of the successes I had though was drawing the lines in all directions and getting the background colour changed as changing the background colour was with a click of a button

I When I was drawing the lines using the up down left and right keys the program recognised my key presses this was good because it was the most fundamental part of the program to be able to draw as that was the main purpose of the program

C# Windows Project Summary

Overall this was a project that I would for sure not want to repeat as it is a lot of key presses to program and it was long to do and had to program graphics such as brush colours which sounds easy to program

C# Windows Rating of Project tackled

6/10

Lots of problems and key presses and graphic objects to program hard syntax to finalise and get working correctly

C# Windows Etch a Sketch Project Test Plan

C# Windows Etch a Sketch Project Test Plan				
Test Number	**Keys Pressed or Description of test**	**Expected Result**	**Actual Result**	**Further test comments**
1	B	Makes the sketch line thicker	The Sketch line was made thicker	The circle got bigger each time I kept pressing B to make the drawing more thicker in size
2	S	Makes the sketch line thinner	The Sketch line was made thinner	The circle got smaller each time I kept pressing S to make the drawing more thicker in size
3	F1	Should change colour of the line	The colour of the line stayed the same	Due to an error in the code this was not happening
4	Up Key	Draw a line upwards	The line drew upwards	The line drew straight up
5	Down Key	Draw a line downwards	The line drew downwards	The line drew downwards
6	Left Key	Draw a line to the left	The line drew upwards	The line can go diagonal left as well
7	Right Key	Draw a line to the right	The line drew upwards	The line can go diagonal right as well
8	Does the message box come up when the user wants to quit	Message box should come up	The message box came up	The message box is clear to users of the program who ill use it
9	Does the line hit the side edges	The lines do hit the edges	The lines hit the side of the edges	The lines by hitting the sides could form a nice rectangle