AST21105 Object-Oriented Programming & Design

Lab 10 – More onGUI Programming using C#

A. Submission Details

No submission is required for this lab. However, you are highly encouraged to practice all the techniques mentioned in this note for your assignment.

B. Objective

As mentioned, programs with Graphical User Interface (GUI) are undoubtedly the most useful and important applications that we need nowadays. Programs with no Graphical User Interface still serve the needs of users, however, these programs are generally less user-friendly and therefore not easy to use compared to those with good GUIs. In this lab, you will once again be introduced to building of GUIs so as to help you out for the assignment. The objective of this lab is to help you get familiar with the procedure on creating Graphical User Interface using C# for your assignment.

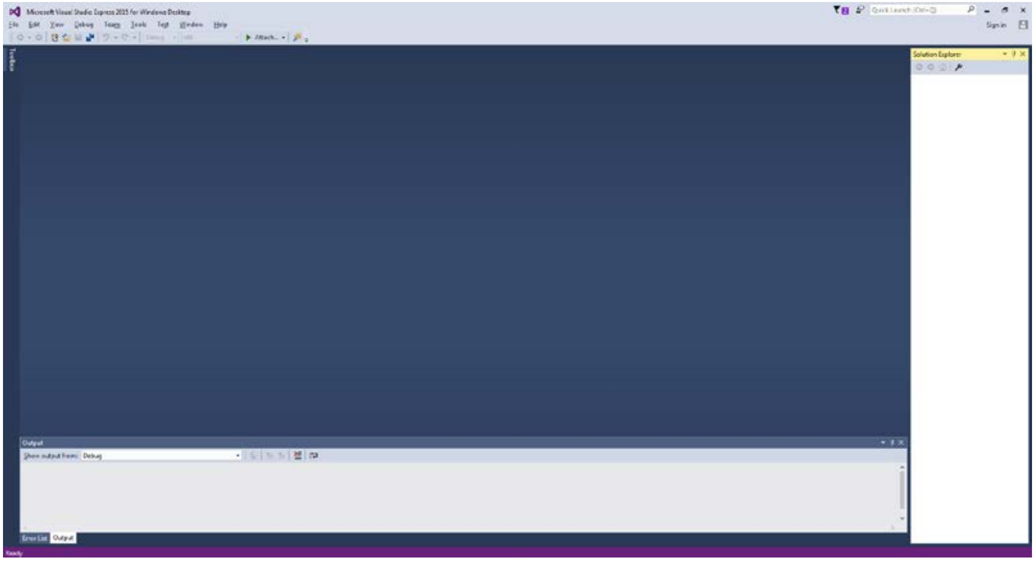
The purpose of this part of lab is to demonstrate you how to use Visual Studio development environment to write simple GUI applications for your assignment. This lab focuses on giving you the idea of an event-driven architecture and helps you to understand how to create simple user interface using Visual Studio Form Designer and to get the concept of an event as a user action that triggers execution of a named code segment.

C. Demonstration

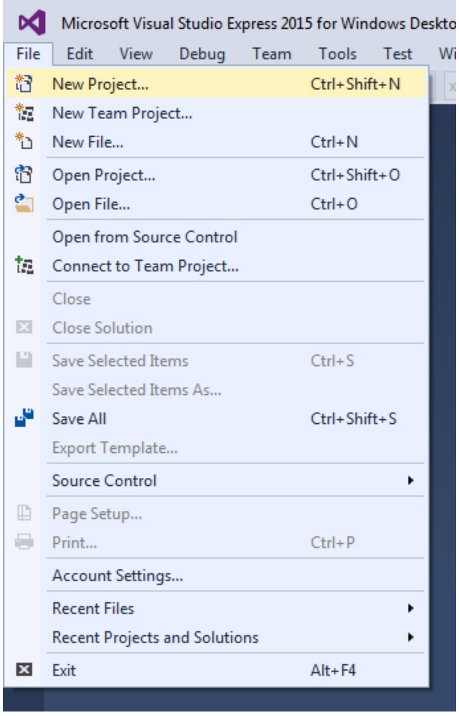
The purpose of this part of lab is to demonstrate you how to use Visual Studio development environment to write simple GUI applications. This lab focuses on giving you the idea of an event-driven architecture and helps you to understand how to create simple user interface using Visual Studio Form Designer and to get the concept of an event as a user action that triggers execution of a named code segment.

Procedure for creating Graphical User Interface (GUI) using Visual C#:

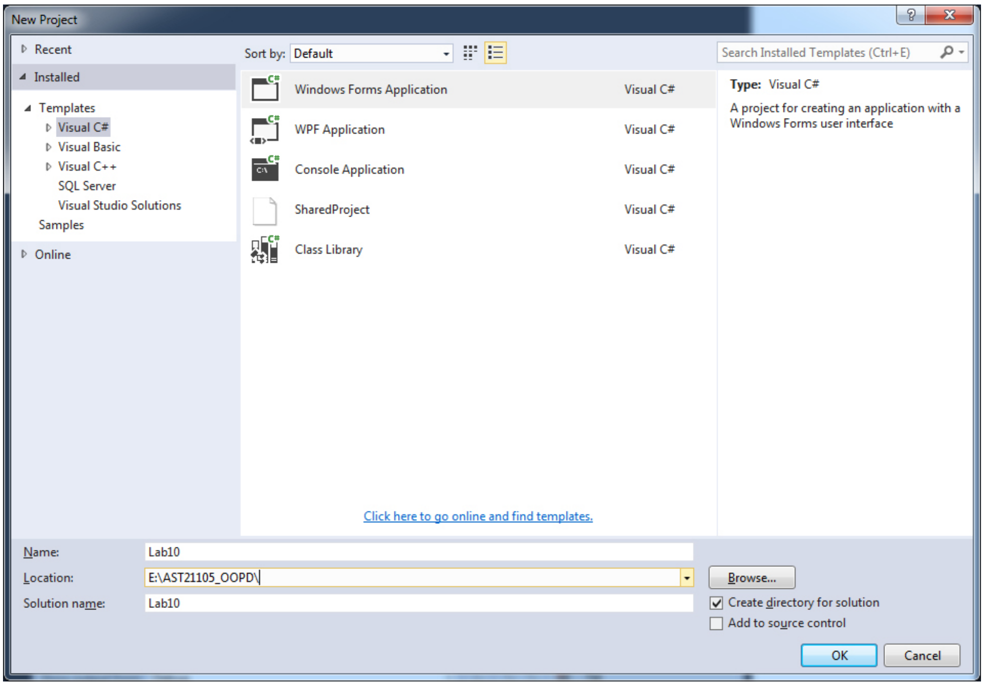
1. Start up Visual Studio and you should see something as follows:



2. Create a new project by clicking on the "File" > "New Project".



3. Create a Visual C# “Window Forms Application” and name it as Lab9. Then click “OK”.



Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: Lab10

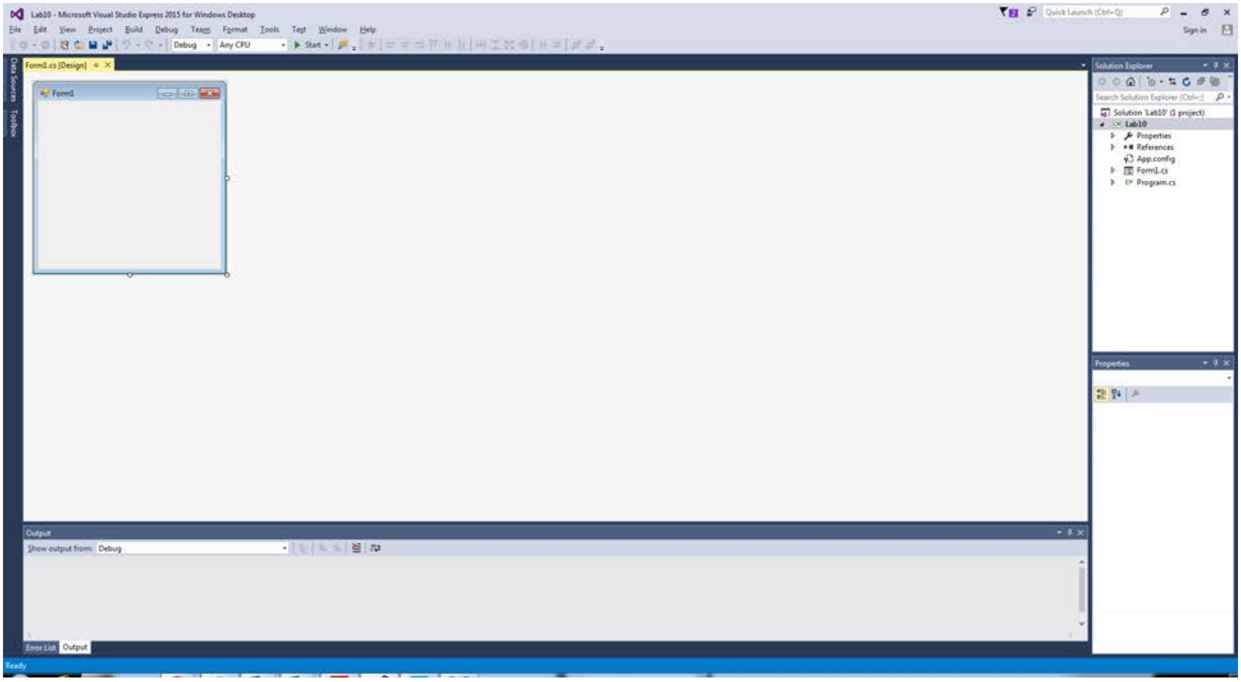
Location: E:\AST21105_OOPD\

Solution name: Lab10

☑ Create directory for solution

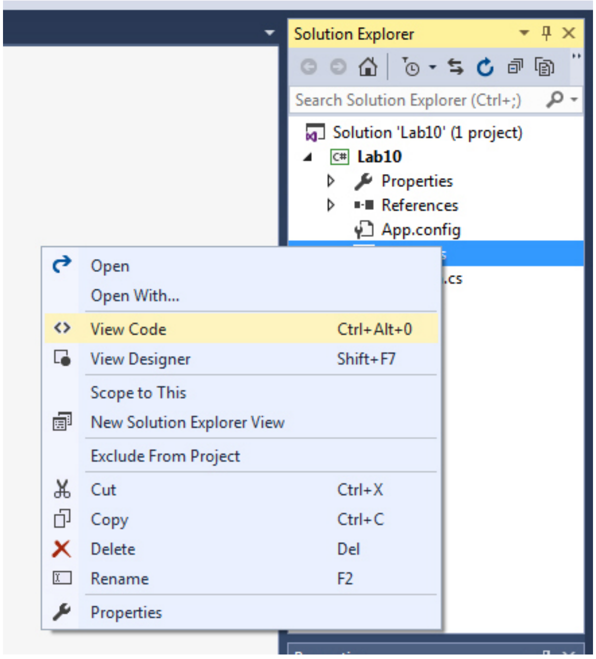
☐ Add to source control

After the application project is created, the following should be shown in screen.



There may be some differences in configuration (such as placement of the tool bars and boxes). A new window application will start with an ***empty form called Form1***. This form represents the user interface that will be shown when the application is first run. You can interact with it to change its appearance.

4. View the code file “Form1.cs” by right-click on the Form1.cs ☒ “View Code”.



5. In “Form1.cs”, define a new class “MyPictureBox” that inherits PictureBox.

```
public class MyPictureBox : PictureBox
{
    private bool selected = false; public bool isSelected()
       {
          return selected;
       }
       public void setSelected(bool s)
       {
          selected = s;
       }
}
```

MyPictureBox is a derived class of a class called PictureBox that has been defined in C#. As it is a derived class of PictureBox, it consists of all properties in PictureBox. In this new class, one data member (we call it instance variable in C#) “selected” is added. It denotes whether the picture box is selected or not. Its initial value is set to false.

To provide access to this data member, two member functions (we call it methods in C#) are defined. isSelected() returns the current value of selected, and setSelected() assign new value to selected.

6. Replace the original code in the class “Form1” by the following.

```
private MyPictureBox[] pb = new MyPictureBox[13];
private TextBox tb = new TextBox();
private int noCard = 0;

public Form1()
{
   InitializeComponent();
   this.SuspendLayout();
   this.Text = "Lab10 - More on GUI Programming using C#";
   this.ClientSize = new Size(395, 100);
   for (int i = 0; i < 13; i++)
   {
      pb[i] = new MyPictureBox();
      pb[i].Image = Image.FromFile("image1" + (i + 1).ToString() + ".jpg");
      pb[i].Location = new Point(10 + i * 26, 35);
      pb[i].Size = new Size(26, 36);
      pb[i].Click += new System.EventHandler(this.pictureBox_MouseDown);
      this.Controls.Add(pb[i]);
   }
   tb.Location = new Point(360, 39); tb.Width = 20;
   tb.TextAlign = HorizontalAlignment.Center;
   tb.Text = noCard.ToString();
   this.Controls.Add(tb);
   this.ResumeLayout();
}

private void pictureBox_MouseDown(object sender, EventArgs e)
{
    if (!((MyPictureBox)sender).isSelected())
    {
        ((MyPictureBox)sender).Location = new
        Point(((PictureBox)sender).Location.X,
        ((PictureBox)sender).Location.Y - 10);
        ((MyPictureBox)sender).setSelected(true);
        noCard++;
    }
    else
    {
        ((MyPictureBox)sender).Location = new
        Point(((PictureBox)sender).Location.X,
        ((PictureBox)sender).Location.Y + 10);
        ((MyPictureBox)sender).setSelected(false);
        noCard--;
    }
    tb.Text = noCard.ToString();
}
```

The following explains the meaning of each statement in the last page.

```
private MyPictureBox[] pb = new MyPictureBox[13];
private TextBox tb = new TextBox();
private int noCard = 0;
```

13 instances of MyPictureBox and a TextBox are instantiated. In addition, an integer named noCard is defined and initialized to 0, which denotes the no of card currently being selected is 0.

```
this.Text = "Lab10 - More on GUI Programming using C#";
```

This sets the title text of the form to “Lab10 – More on GUI Programming using C#”.

```
this.ClientSize = new Size(395, 100);
```

The above statement sets the size of the form to 395 in length and 100 in height.

```
for (int i = 0; i < 13; i++)
{
    pb[i] = new MyPictureBox();
    pb[i].Image = Image.FromFile("image1" + (i + 1).ToString() + ".jpg");
    pb[i].Location = new Point(10 + i * 26, 35);
    pb[i].Size = new Size(26, 36);
    pb[i].Click += new System.EventHandler(this.pictureBox_MouseDown);
    this.Controls.Add(pb[i]);
}
```

The code segment above adds 13 instances of MyPictureBox to the form. Each instance is linked with an image loaded from disk, location and size are set and an event handler is associated with it. Finally, all the instances are added to the form.

```
tb.Location = new Point(360, 39);
tb.Width = 20;
tb.TextAlign = HorizontalAlignment.Center;
tb.Text = noCard.ToString();
this.Controls.Add(tb);
```

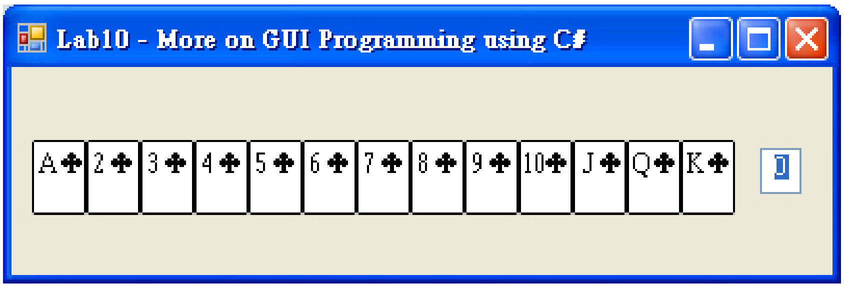
Properties of the created textbox are set. Properties set include Location, Width, TextAlign and Text. Lastly, the textbox is added to the form.

```
private void pictureBox_MouseDown(object sender, EventArgs e)
{
  if (!((MyPictureBox)sender).isSelected())
  {
    ((MyPictureBox)sender).Location = new
    Point(((PictureBox)sender).Location.X,
    ((PictureBox)sender).Location.Y - 10);
    ((MyPictureBox)sender).setSelected(true);
    noCard++;
  }
  else
  {
    ((MyPictureBox)sender).Location = new
    Point(((PictureBox)sender).Location.X,
    ((PictureBox)sender).Location.Y + 10);
    ((MyPictureBox)sender).setSelected(false);
    noCard--;
  }
  tb.Text = noCard.ToString();
}
```

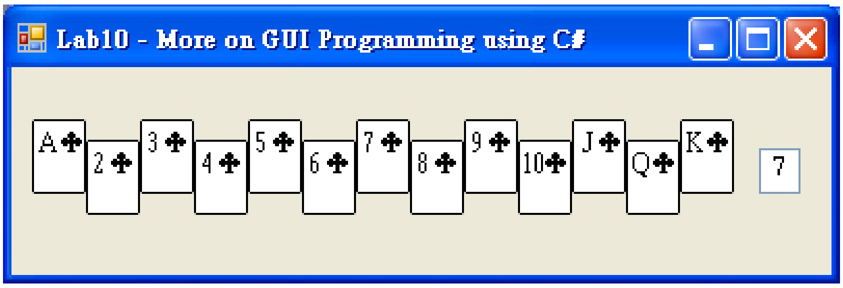
This is an event handler (a method), which moves the selected picture box up and down and count the number of cards selected.

7. Unzip the given ZIP file, images.zip, and copy all the extracted JPEG to Lab10\Lab10\bin\Debug.

8. Run the program by clicking the "green triangular button" and the following will be shown on screen.



9. Select some of the playing cards by clicking on them. After that, you should be able to see the number of cards selected is shown in the textbox.



-End-