MUHAMMAD ALI NASIK AWAN

023-20-0107

BSSE V

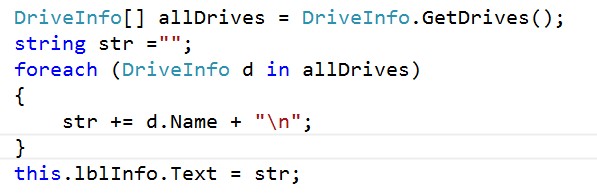
OBJECTIVES:

1. Understanding and working with file system.
2. Understanding and working with Progress bar control.
3. Understanding and working with Check boxes and Radio buttons.
4. Practice activities.

**Objective 1:**

# File System

* Accessing drives, directories and files on system.
* *System.IO.DriveInfo class* o This class models a drive and provides methods and properties to query for drive information.
  + Use DriveInfo to determine available drives
  + You can also query to determine the capacity and available free space on the drive.
  + The drive name must be either an uppercase or lowercase letter from 'a' to 'z'



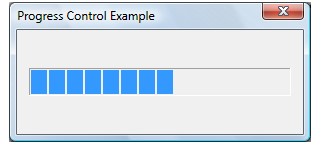
* ***System.IO.DirectoryInfo class*** o Exposes instance methods for creating, moving, and enumerating through directories and subdirectories. o This class cannot be inherited.

* ***System.IO.FileInfo class*** o Provides properties and instance methods for the creation, copying, deletion, moving, and opening of files. o This class cannot be inherited.

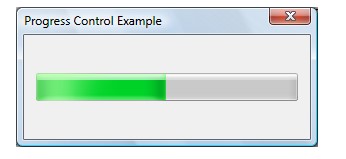
**OBJECTIVE 2:** Understanding and working with Progress bar control.

# Progress Bar control

* A Progress Bar control visually indicates the progress of a lengthy operation in one of three styles:
  + Segmented blocks that increase in steps from left to right.o A continuous bar that fills in from left to right.
  + A block that scrolls across a Progress Bar in a marquee fashion.
* Following illustration show progress bar without visual styles, segmented blocks.



* Following illustration show progress bar with visual styles, continuous progress bar.



* Range and current position of Progress Bar
* A progress bar's *range* represents the entire duration of the operation, and the *current position* represents the progress that the application has made toward completing the operation.
* If you do not set the range values, the system sets the minimum value to 0 and the maximum value to 100.
* **Minimum and Maximum** properties are used to define custom range values.
* **Value** property shows the progress that application has made towards completing the operation.
* **Style** property indicates the style of progress in progress bar with the help of values of **ProgressBarStyle** enumeration and its members are: o Blocks, Continuous and Marquee

**Objective 3:** Understanding and working with Check boxes and Radio buttons.

# Checkboxes

* Use a Check Box to give the user an option, such as true/false or yes/no.
* The Check Box control can display an image or text or both.
* Check Box allow the user to choose from a list of options.
* Check Box controls let the user pick a combination of options.
* The [Appearance](https://msdn.microsoft.com/en-us/library/system.windows.forms.checkbox.appearance(v=vs.110).aspx) property determines whether it appears as a typical Check Box or as a button.
* The [ThreeState](https://msdn.microsoft.com/en-us/library/system.windows.forms.checkbox.threestate(v=vs.110).aspx) property determines whether the control supports two or three states.
* Use the [Checked](https://msdn.microsoft.com/en-us/library/system.windows.forms.checkbox.checked(v=vs.110).aspx) property to get or set the value of a two state check box.
* Use the [CheckState](https://msdn.microsoft.com/en-us/library/system.windows.forms.checkbox.checkstate(v=vs.110).aspx) property to get or set the value of a threestate CheckBox control.

# Radio Buttons

* The RadioButton control can display text, an [Image,](https://msdn.microsoft.com/en-us/library/system.drawing.image(v=vs.110).aspx) or both.
* When the user selects one option button (also known as a radio button) within a group, the others clear automatically.

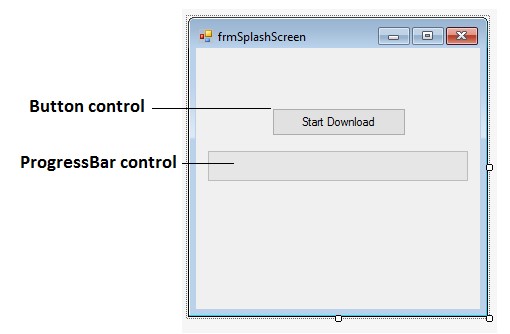
All RadioButton controls in a given container, such as a [Form,](https://msdn.microsoft.com/en-us/library/system.windows.forms.form(v=vs.110).aspx) constitute a group.

* To create multiple groups on one form, place each group in its own container, such as a [GroupBox](https://msdn.microsoft.com/en-us/library/system.windows.forms.groupbox(v=vs.110).aspx) or [Panel](https://msdn.microsoft.com/en-us/library/system.windows.forms.panel(v=vs.110).aspx) control.
* RadioButton and [CheckBox](https://msdn.microsoft.com/en-us/library/system.windows.forms.checkbox(v=vs.110).aspx) controls have a similar function: they offer choices a user can select or clear.
* The difference is that multiple [CheckBox](https://msdn.microsoft.com/en-us/library/system.windows.forms.checkbox(v=vs.110).aspx) controls can be selected at the same time, but option buttons are mutually exclusive.
* Use the [Checked](https://msdn.microsoft.com/en-us/library/system.windows.forms.radiobutton.checked(v=vs.110).aspx) property to get or set the state of a RadioButton.

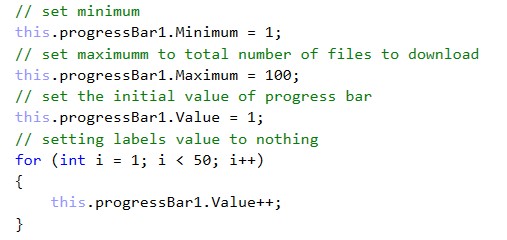
**ACTIVITIES SECTION**

# ACITVITY 1: STEPS

* Create a windows forms application named StudentManagement.
* Create a form in it named frmSplash.
* Place following controls on form.



* Place given code in button’s click event.



* Execute the code and observe the results.

**Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab06

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

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

{

progressBar1.Minimum = 1;

progressBar1.Maximum = 100;

progressBar1.Value = 1;

for (int i = 0; i < 50; i++)

{

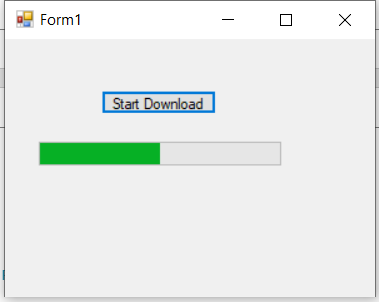
progressBar1.Value++;

}

}

}

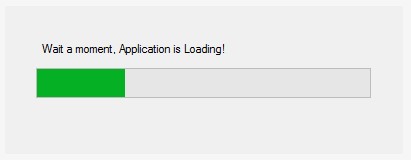
}

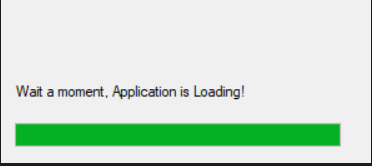


# ACITVITY 2: STEPS

* Create a windows forms application named SplashScreen.
* Create a form named frmSplashScreen.
* Place ProgressBar control on the form.
* Set StartPosition property of form to CenterScreen. Set FormBorderStyle property of form to none.

* Now write code on form\_load event.
* Wait for 10 seconds just to show that application is loading.
* In that 10 seconds progress in progress bar should be completed 100%.
* When progress is complete then make current window invisible and show the next window that you want.





**Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab06

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

progressBar1.Minimum = 1;

progressBar1.Maximum = 100;

progressBar1.Value = 1;

for (int i = 0; i < 99; i++)

{

progressBar1.Value++;

}

}

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

{

}

private void label1\_Click(object sender, EventArgs e)

{

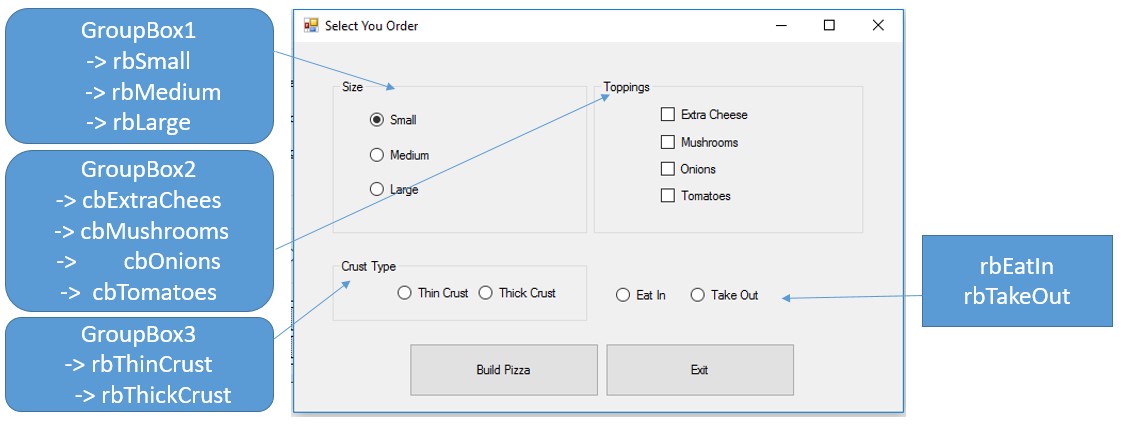
}

}

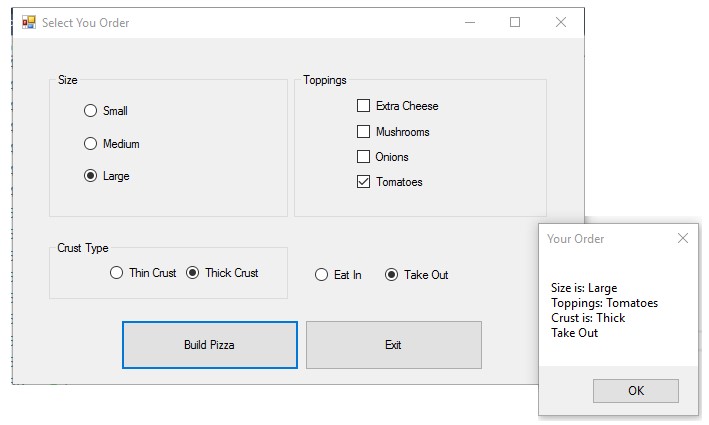
}

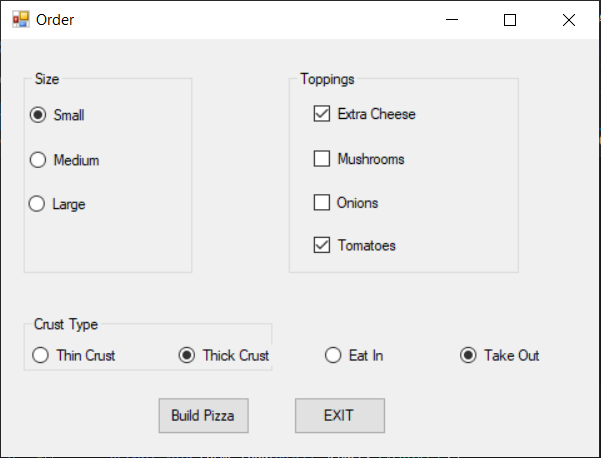
# ACTIVITY 3: STEPS

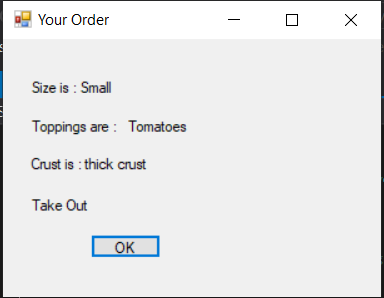
* Create a windows form named frmBuildPizza.
* Create interface on the form as given in diagram.
* You only have to work on event of btnBuildPizza to make an order.
* Exit button will close the application.



#  Program Output







**Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace PizzaShop

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string size, toppings, crust, place;

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

{

}

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

{

Application.Exit();

}

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

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (radioButton6.Checked)

{

place = "Eat In";

}

if (radioButton7.Checked)

{

place = "Take Out";

}

if (radioButton4.Checked)

{

crust = "Thin Crust";

Console.WriteLine(crust);

}

if (radioButton5.Checked)

{

crust = "thick crust";

}

if (sender == checkBox1)

{

toppings += " " + " Extra Cheese";

}

if (checkBox2.Checked)

{

toppings += " " + " Mushrooms";

}

if (checkBox3.Checked)

{

toppings += " " + " Onions";

}

if (checkBox4.Checked)

{

toppings += " " + " Tomatoes";

}

if (radioButton1.Checked)

{

size = "Small";

}

if (radioButton2.Checked)

{

size = "Medium";

}

if (radioButton3.Checked)

{

size = "Large";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Form2 test = new Form2();

test.order(size,toppings,crust,place);

this.Hide();

test.Show();

toppings = "";

}

private void groupBox3\_Enter(object sender, EventArgs e)

{

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

}

}

**Form2:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace PizzaShop

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

String size, toppings, crust, place;

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

{

Application.Exit();

}

public void order(string a,string b ,string c , string d )

{

size = a;

toppings = b;

crust = c;

place = d;

}

private void Form2\_Load(object sender, EventArgs e)

{

label1.Text = "Size is : " + size;

label2.Text = "Toppings are : " + toppings;

label3.Text = "Crust is : " + crust;

label4.Text = place;

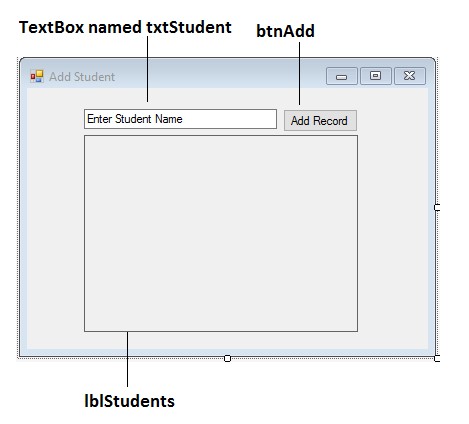
}

}

}

# ACTIVITY 4: STEPS

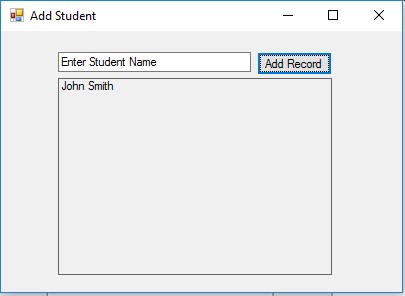
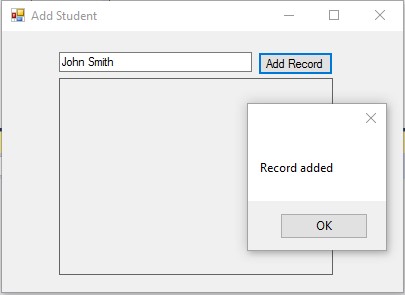
* Create a form named frmDynamicArray.
* Create given interface on form as given in image.



* Create a class named StudentClass.

o Place FirstName : Public field in StudentClass.

* Create an array in a form of StudentClass that will hold the records created by user.
* Make an array dynamic, don’t use built in classes like ArrayList and so on.
* Use Leave and Enter events of text box to set and clear the text. o **Leave** event triggers when text box is not active control. o **Enter** event triggers when text box becomes active control.
* Working mechanism of above task:



**Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

label1.Text +="\n"+ textBox1.Text;

textBox1.Text = "";

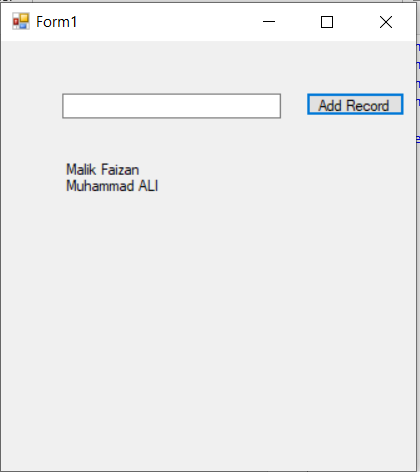
MessageBox.Show("Record Added");

}

}

}

OUTPUT:

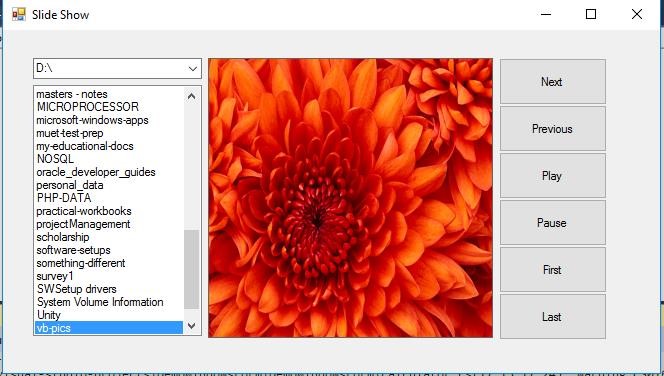


# ACITVITY 5: STEPS

* Create application named SlideShowApp.
* Create form in it named frmSlideShow.
* Create interface of form according to given user interface.

#  Conditions For Program

* On loading of form display current drives of system in combo box.
* When user selects any drive display all its directories in list box.
* When user selects any directory if it has pictures then first picture should load in picture box and display rest of pictures if user want to do that by previous or next buttons or by playing slide show.



**CODE:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace pictureShow

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string pic = "D:\\vp\\LABS\\Lab-04\\pictureShow\\wallpapers\\";

int count = 1;

int prev;

int no = 0;

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

{

comboBox1.DataSource = System.IO.DriveInfo.GetDrives();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Dispose();

}

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

{

pictureBox1.ImageLocation = pic + "\\" + count + ".jpg";

count++;

prev = count - 1;

if (count > 5)

{

//MessageBox.Show("No More Pics !");

count = 1;

}

}

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

{

pictureBox1.ImageLocation = pic + "\\" + prev + ".jpg";

if (prev < 1)

{

prev = 5;

count = prev;

pictureBox1.ImageLocation = pic + "\\" + prev + ".jpg";

}

prev--;

}

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

{

pictureBox1.ImageLocation = pic +"\\"+ count + ".jpg";

count++;

if(count == 5)

{

count = 1;

}

}

private void button4\_Click(object sender, EventArgs e)

{

timer1.Enabled = true;

}

private void button5\_Click(object sender, EventArgs e)

{

timer1.Enabled = false;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

String[] dirs = System.IO.Directory.GetDirectories("C:\\");

int i;

String[] files = System.IO.Directory.GetFiles("C:\\");

if (comboBox1.SelectedIndex == 0){

for (i = 0; i < dirs.Length; i++)

{

listBox1.Items.Add(dirs[i]);

}

for (i = 0; i < files.Length; i++)

{

listBox1.Items.Add(files[i]);

}

}

else if (comboBox1.SelectedIndex == 1)

{

dirs = System.IO.Directory.GetDirectories("D:\\");

files = System.IO.Directory.GetFiles("D:\\");

listBox1.Items.Clear();

for (i = 0; i < dirs.Length; i++)

{

listBox1.Items.Add(dirs[i]);

}

for (i = 0; i < files.Length; i++)

{

listBox1.Items.Add(files[i]);

}

}

else if (comboBox1.SelectedIndex == 2)

{

dirs = System.IO.Directory.GetDirectories("E:\\");

files = System.IO.Directory.GetFiles("E:\\");

listBox1.Items.Clear();

for (i = 0; i < dirs.Length; i++)

{

listBox1.Items.Add(dirs[i]);

}

for (i = 0; i < files.Length; i++)

{

listBox1.Items.Add(files[i]);

}

}

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

pic = listBox1.SelectedItem.ToString();

no = System.IO.Directory.GetFiles(pic).Length;

Console.WriteLine(no);

pictureBox1.ImageLocation = pic+"\\" + count+ ".jpg";

}

private void button7\_Click(object sender, EventArgs e)

{

pictureBox1.ImageLocation = pic + "\\" + no + ".jpg";

}

private void button6\_Click(object sender, EventArgs e)

{

pictureBox1.ImageLocation = pic + "\\" + "1" + ".jpg";

}

}

}

OUTPUT:

