VISUAL C#
PROGRAMMING
Multiple Forms
Often, when creating a C# application, you will want or need multiple pages or forms for your application. This is an easy task to accomplish in C# and allows you to design a program that looks better, is easier to use and is better organized.
Step One

- Your first step is to create the initial form of the application
  - It is best to make this the control page
  - Create buttons that will cause the other pages to open up
Main Page

This is my master page

Goto Form2
Goto Form3
Goto Form4

Goto Form5
Goto Form6
Goto Form7
Step 2 – Create new forms

• The next step is to add the window forms you will need for the project
  • This is found under project – add windows form
Type: Visual C# Items
A blank Windows Form
Directory Content
Step 3

- Each button on the main form requires code
  - Part of the code shows the new form
  - Part of the code hides the main form

```csharp
Private void btn1_Click(object sender, EventArgs e)
{
    form2 myNewForm = new form2();
    myNewForm.Show();
    this.Hide();
}
```
Step 4

- You need to build each of the other forms
  - Build them as you would, having whatever functions are required for the form to carry out its task
  - Each form requires a button coded to take you back to the main form
  - This should be a reverse of the code that took you to the new form

Private void btn2_Click(object sender, EventArgs e)
{
    Form1 myNewForm = new Form1();
    myNewForm.Show();
    this.Hide();
}
How to implement check boxes

- Your forms may utilize tools such as check boxes
- You can test whether check boxes have been checked
  - If(cb1.Checked == true)
- You can also test if a check box has not been checked
  - If(cb1.Checked == false)
Simplifying Some Confusion

• If you require a conditional statement based on the state of multiple check boxes, the possible combinations become staggering.

• You can shortcut the problem by only checking for one state by using variables for storage of string values and some concatenation for an output statement.
How to implement Radio Buttons

• Radio buttons work similar to check boxes
• The main difference is that you can only select one radio button in a group.
• We use group boxes to group related radio buttons
My Form2 Layout

Select your pizza options

Size  ○ Small  ○ Medium  ○ Large
Crust  ○ Thick  ○ Thin

Toppings:  □ Extra Cheese  □ Sausage  □ Pepperoni  □ Mushroom  □ Onion

□ Delivery

Calculate Order

Order Summary

Total Cost:

Return to Main
The code behind the form

```csharp
private void btn1_Click(object sender, EventArgs e)
{
    string st1 = "";
    string st2 = "";
    string st3 = "";
    string st4 = "";
    string st5 = "";
    string st6 = "";
    string st7 = "";
    string st8 = "";
    string st9 = "";
    string st10 = "";
    string st11 = "";
    string st12 = "";
    double total = 0;

    if (cb1.Checked == true)
    {
        st1 = "extra cheese, ";
        total = total + .50;
    }
    if (cb2.Checked == true)
    {
        st2 = "sausage, ";
        total = total + 1.00;
    }
    if (cb3.Checked == true)
    {
        st3 = "pepperoni, ";
        total = total + 1.00;
    }
    if (cb4.Checked == true)
    {
        st4 = "mushroom, ";
        total = total + 1.00;
    }

    tb1.Text = "One " + st6 + st7 + st8 + st9 + st10 + "pizza with " + st1 + st2
    tb2.Text = total.ToString("#.00");
```
The Final Output

Select your pizza options

Size
- Small
- Medium
- Large

Crust
- Thick
- Thin

Toppings:
- Extra Cheese
- Sausage
- Pepperoni
- Mushroom
- Onion

- Delivery

Calculate Order

Order Summary
One medium thin crust pizza with sausage, pepperoni, mushroom, for delivery.

Total Cost: 12.00

Return to Main