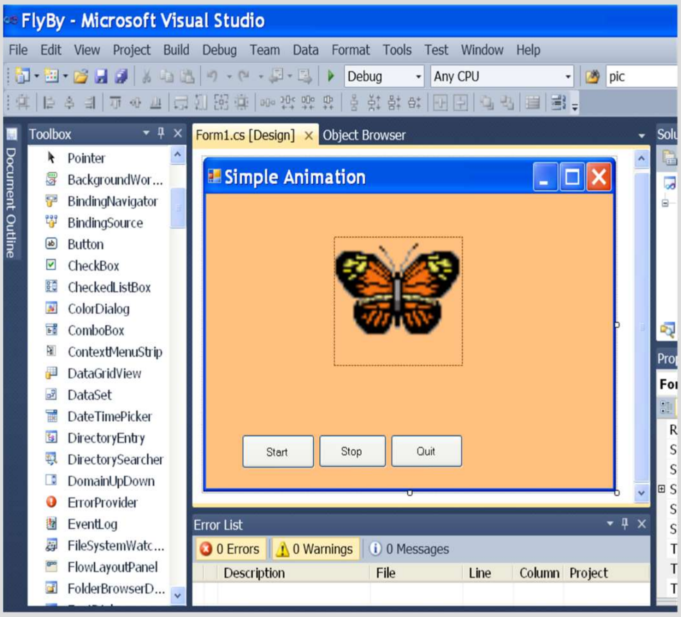
Windows Programming

bucks
new university

.NET Windows Programming Using C#



Application Programming (CO453)

Part C – Weeks 9-13

Optional unit

Web Programming ASP.NET

Introduction to Web Forms and ASP.NET

- So far we have been programming ‘stand-alone’ programs that run inside the Windows environment. But C# also allows us to program for the internet.
- This involves creating **Web Forms** that can be displayed in an internet browser such as Internet Explorer.
- The latest Microsoft Web programming technology is called **ASP.NET**. This gives us a set of libraries and controls that enable us to write programs to interact with the user, keep ‘state’ information and produce HTML
- When we create a Web Form we are using ASP.NET.
- Each Web Form has 2 separated parts:
  - the **HTML** code that displays the page (stored in an **.aspx** file)
  - the **C#** code (in an **.aspx.cs** file)
- The **HTML** code is produced automatically but it can be modified if you know html.
- The **C#** (or ‘code-behind’ file) contains the instructions that enable the form to respond to events

9.1 Creating a new Web Site

- In Visual Studio, choose **File … New … Web Site**
- Then Select **ASP.NET Empty Web Site** as the template and **Visual C#** as the language
- Click the **Browse** button to select a location for your WebSite and change the name to **MyBlog**. Then click **OK**.
- Then select **Window** … **Add New Item** … select **Web Form** (**Default.aspx** is the default name)
- Notice the 3 tabs (**Design**, **Split**, **Source**) at the bottom of the centre section .. make sure that the **Design** tab is selected
- You should be able to see the **Properties** window and the **ToolBox** window .. although they will look quite different from usual. If not, select them using **View** on the menubar.

9.2 Creating a User Interface

- Click on the blank page and press the **Enter** key 4 times
- Now type the words: **Enter your first name:**
- Then press the space bar a few times and add a **TextBox** to the page from the **ToolBox** (either double-click or drag one)
- Press the enter key a couple of times and add the words **Enter your surname:**
- Put another TextBox in position to the right of this as before
- Press the enter key a couple of times and add a **Label** to the page .. remove the **Text** from the label
- Now change the **ID** properties (IDs are used instead of names) of the text boxes and label to **txtFname**, **txtSname** and **lblMessage**
- Press the Enter key a couple of times, then the spacebar a dozen times
- Add a **Button** here and change its Text property to **Submit** and its **ID** to **btnSubmit**
- Double-click the button and enter code into its click method in the normal way so as to get lblMessage to display some text including the first name and surname entered e.g. **Welcome “Your Name” to my Blog**

9.3 Displaying your Web Page

- You can run and test your program using **Start without Debugging** (Ctrl + F5)
    - This should display your page using the Internet Explorer browser.
- Alternatively you can display a Page Preview by doing the following:
    - Right-click on the page (or the aspx file)
    - Select **View in Browser**

9.4 Changing the Document

- Go back to the **Design** tab for your page
- Look in the **Properties** window just under the word Properties to drop down a list of the objects in use … choose **DOCUMENT**
- Now change the **Title** property to **Brian's Fabulous Blog** (using your name)
- Use the **Text** property to change the colour of the text on the page
- To change the background colour of the page, go to the formatting toolbar(underneath **Help** in the menu bar) then find the icon with two squares(⧉). Then select a colour from the pallet.

Task 9.1: My First Web Site

1. Follow the steps 9.1 to 9.4 above .. this creates the basic Web Page
2. Add a **Label** control near the top of the page
    - Change its **Text** property to: **Welcome to my Blog**
    - Change the **Font** to something exotic
    - Change the Font **Size** to **XX-large** and set it to **Italic**
3. Add another label that says: **Please enter your details below**
4. Find a suitable **image** and copy this into the MyBlog folder
    - Then in the **Solution Explorer** window click the **Refresh** button .. your image filename should now appear in this window
    - Now use the **Toolbox** to add an **Image** control to the page
    - Link this to your image by using its **ImageUrl** property
    - You may want to change the **Height** and **Width** of the image too
5. Add another label and textbox **txtAge** for entering the users age
6. Finally add a second **button** to the page
    - program it to Clear the contents of all the text boxes.

Task 9.2: Adding a ListBox (country of origin)

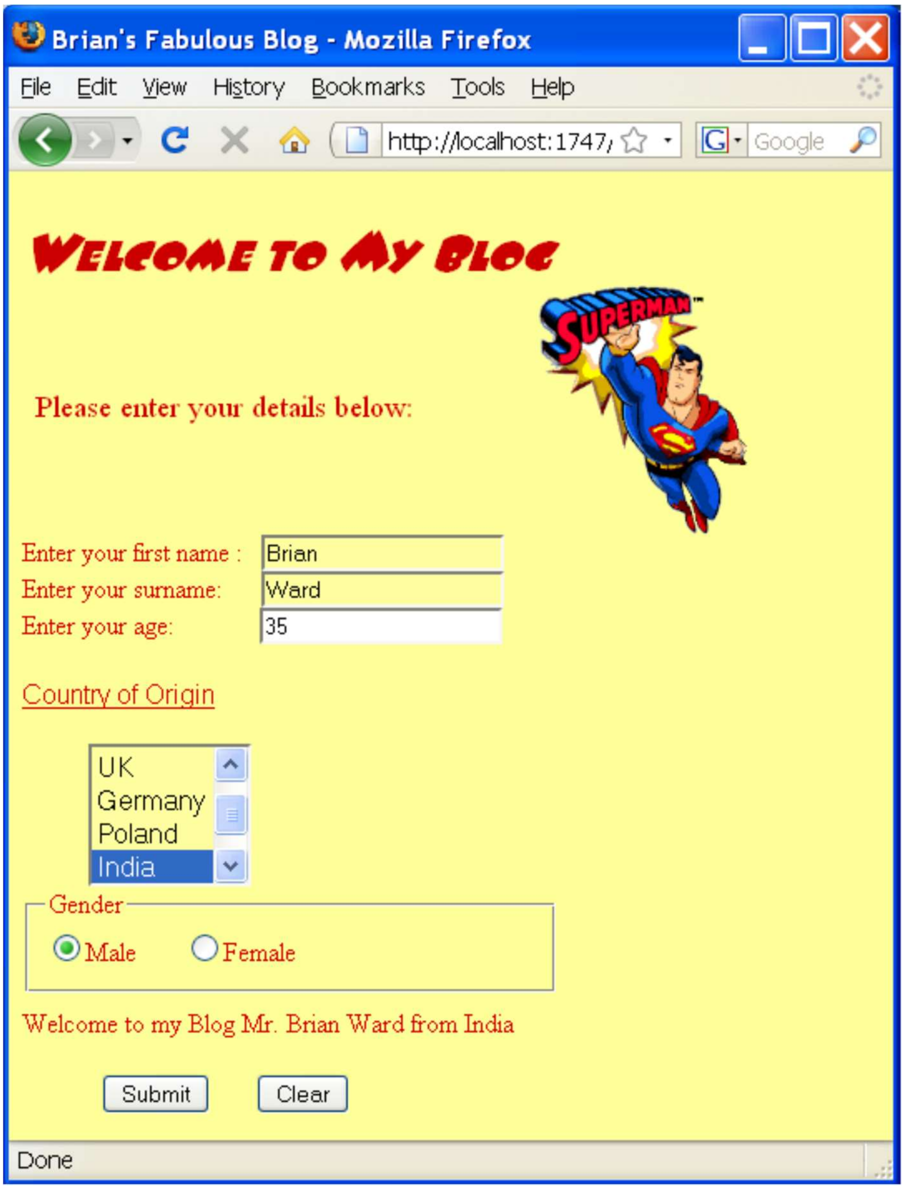
1. Now choose a suitable position for a listbox
2. Add a ListBox from the Toolbox and change its **ID** to **lstCountry**
3. Add a label above the listbox saying: **Country of Origin**
4. Select **Items** from the listbox Properties window
    - Add 6 items (the names of 6 different countries)
    - Select a medium size **font**
    - Add a **ToolTip** .. **Select your country from this list**
5. Now add more code to the Submit button so the message now displays the country selected and looks like this (for example):

   Welcome to my Blog "Your Name" from "XXXXX"

Task 9.3: Adding RadioButtons inside a Panel

1. Now choose a suitable position for a male/female gender choice
2. Add a **Panel** from the Toolbox
3. Change its **GroupingText** property to **Gender**
4. Change its **BorderStyle** property to something suitable
5. Now add 2 **RadioButtons** to the panel and change their **IDs** to **rbnMale** and **rbnFemale**
6. Also change their **Text** properties to display the words **Male** and **Female**
7. Add a **ToolTip** to display : **Choose your Sex here**
8. Now add code to do the following:
    - The message displays **Mr** in front of the name if **male** is selected
    - The message displays **Miss** in front of the name if **female** is selected

- Now you should have a page that displays something like the one below:


Brian's Fabulous Blog - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:1747/
Google
WELCOME TO MY BLOG
SUPERMAN

Please enter your details below:

Enter your first name : Brian
Enter your surname: Ward
Enter your age: 35

Country of Origin

UK
Germany
Poland
India

Gender

● Male ○ Female

Welcome to my Blog Mr. Brian Ward from India

9.5 Adding another Web Page

- Select **WebSite** from the menubar, followed by **Add New Item**
- Select the **Web Form** template and change the name to **myDetails.aspx**
  - Check that the language is set to **C#**
  - Check that **Place code in a separate file** is ticked
- Now click the **Add** button
- Click the **Design** tab at the bottom of the page to begin designing this new page

9.6 Preparing the new Page

- Place a large label near the top of the new page displaying: **All My Secret Information**
- Find suitable picture of yourself and put this into the **MyBlog** folder so it will be picked up automatically

9.7 Adding a Table

- Positioning items on a web page can be tricky and there are many different browsers, screen sizes, resolutions etc. that can mess up your layout.
- If you put items inside a **Table**, this helps to bind them together and preserve the layout of your Web Page.
- Table cells can hold pictures, labels, textboxes, buttons, etc.
- The table border does not have to be visible (e.g. by making it size zero)
- Add a **Table** to your new **myDetails** page as follows:
  - Go to Design view for the myDetails page
  - Select **Table** from the menubar, followed by **Insert Table**
  - Choose **4** rows and **3** columns
  - Change the **Layout … Specify Width** to **80 percent** and click **OK**
  - The table outline should appear on your page but this will not be visible to the user (unless you change the table's border property)

9.8 Changing the Table Style

- Select the whole table and look at the **Properties** window
- Enter **tblDetails** for the value
- Select **Align** then **Center** to centre the table in the page
- Now select **Style** and click on the '…' button here.
  - In the **Modify Style** dialog box you can change lots of things .. font, border, position, etc.
  - Center the table by selecting **Box** ... **Margin** (untick 'same for all') … **Right : Auto** and **Left : Auto**
  - Then select **Table** and then change the **table-layout** to **fixed** (Otherwise all the columns will alter as you type into them.)

9.9 Putting things into the Table

- We shall now show how we can put things into the table cells
- See the task sheet on the next page

Task 9.4: The myDetails Table

1. Follow the instructions in 9.5 to 9.8 to add a new page to your blog and then put a table in it
2. Now select the entire first row then in the menubar choose **Table** followed by **Modify** followed by **Merge Cells**
3. Put your picture into this merged first row.
4. In the following rows add text to display certain things about yourself

e.g.	Name:	Brian Ward
	Where I live :	SnootsVille

5. Apply appropriate fonts, sizes and alignments to your text. You can use the **Style** property for this.

9.10 Adding a Hyperlink

- Hyperlinks are used to allow the user to move from one page on your website to another page or to link to another website altogether.
- Add a hyperlink to the first page to link to the new myDetails page as follows:
  - Find **HyperLink** in the Toolbox
  - Display the first page, Default.aspx, in Design view
  - Add a HyperLink to the bottom of this page
  - Set the HyperLink **Text** property to **See My Details**
  - Set the **NavigateUrl** property to **myDetails.aspx**

9.11 Setting the Start Page

- When a website has more than one page, we need to set the start page.
- To do this:
  - Find the file **Default.aspx** in the Solution Explorer window
  - Right-click on this and choose **Set as Start Page**
- Now run your application and check that the hyperlink works OK.

Task 9.5: Hyperlinks

1. Follow the instructions in 9.10 to 9.11 to add a **HyperLink** and set the **Start Page**. Check that all this works.
2. Now add another HyperLink at the bottom of the myDetails page to return back to the previous page.
3. Add another link to the myDetails page with appropriate text that links to the **myface** website (**http://www.myface.com**)
4. Add another link to another suitable website of your choice
5. Check that all this works OK.

9.12 Validation Controls

- **Validation** is the general name used when we want a program to check whether data that has been entered is valid or reasonable or within a certain range of values. For example:
  - an **age** could be checked to be over 18
  - a **name** textbox could be checked to make sure it wasn't left empty
- ASP.NET has several controls that are very useful for doing validation.
  - You can find them in the Toolbox .. in the **Validation** section
- NOTE: A problem was found (May 2014) where a server error is produced when attempting to validate the form. This was found to be caused by the updates to the ASP.NET framework. One work around solution would be to add a web.config key to revert to the validation mode used before version 4.5 of the ASP.NET framework.
  - In **Web.config**, add the following code inside the **<configuration>** (root) element, to allow our pre-4.5 controls to work:

```
<appSettings>
      <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
</appSettings>
```

  - Understanding this code isn't necessary but for those interested:
  - *The key is of type "UnobtrusiveValidationMode", which will use a style of JavaScript designed to be consistent across browsers.*
  - *The value is set to "None" and will therefore default to the validation behaviour used in framework versions before 4.5.*
- Now, go back to **Default.aspx** and add a **RequiredFieldValidator** control to the txtSName box in Default.aspx
  - Then set the **ControlToValidate** property to **txtSName**
  - Set the **ErrorMessage** property to **Error: You must enter your surname**
  - Set the **Display** property to **Dynamic**
- Test the validation to see if it works

Task 9.6: Validation

1. Follow the instructions in 9.12 to validate the **txtSName** textbox
2. Add similar validation to the **txtFName** textbox
3. Add a **RangeValidator** control to the **txtAge** textbox
4. Change the control properties so only ages between 18 and 30 are allowed
5. Also validate this so it must not be left blank (note: you can add more than one validation controls to the same object)

Your Logbook

Put code and screenshots into your logbook

Unit 9: Independent Study

Task 9.7: More Pages

1. Add 2 more pages to your website with appropriate links between all of them
2. Smarten up your pages with images and text so that they form an attractive site and add validation checks to any data entered

Some Extra Useful C# .NET Stuff

Some Display Instructions

```
________.BackColor = Color.Blue;            // set a background colour
________.ForeColor = Color.Yellow;          // set a foreground colour
________.Text = "Hello " + name ;     // set the Text property
```

Delay for a time

```
System.Threading.Thread.Sleep (1000);
 // gives a delay of 1 second (1000 milliseconds)
```

The Math.Pow() function

Use the Math.Pow() function to return the power of a number: e.g.

```
cube = Math.Pow(number, 3);
```

The Math.Abs() function

This function returns the value of a number, ignoring any + or - sign e.g.

```
value = Math.Abs(number);
```

Converting Text into upper case

```
lblName.Text = lblName.Text.ToUpper() ;
// converts a label's Text into upper case (e.g. yes becomes YES)
```

Random Number Generation

- First we must create a new object from the Random class .. e.g.

```
Random  r = new  Random();         // creates a new object called r
```

- Then you can use **r.Next()** to pick the next number e.g.

```
n = r.Next(6) + 1;          // puts either 1,2,3,4,5 or 6 into n
```

Output of Decimal Places

If you want to display a double type of variable (num) to 2 decimal places:in a label:

```
lblAnswer.Text = "The answer is " + num.ToString("0.00") ;
```

Number Conversion

e.g. Convert text from a textbox to the right type using Convert e.g.

```
num = Convert.ToDouble(txtAmount.Text);    or
num = Convert.ToInt32(txtAmount.Text);
```

Closing Down

```
Application.Exit();
```

Graphics Summary (after first getting a Graphics object g)

Clearing the Screen

```
g.Clear (BackColor);              // clear using the current backcolor
```

Rectangles and Squares

```
g.DrawRectangle (myPen, x, y, w, h);      or    g.FillRectangle(myBrush, x, y, w, h);
```

Ellipses and Circles

```
g.DrawEllipse (myPen, x, y, w, h);        or    g.FillEllipse(myBrush, x, y, w, h);
```

Polygons

```
g.DrawPolygon (myPen, shape);             or    g.FillRectangle(myBrush, shape);
```

Lines

```
g.DrawLine (myPen, x, y, w, h);
```

Text

```
g.DrawString (s, myFont, myBrush, x, y);
```

Pens and Brushes (for drawing and filling graphics)

```
Pen  myPen = new Pen(Color.Green, 5);              // create a new green pen, size 5
Brush  myBrush = new SolidBrush(Color.Red);        // create a new red brush
```

RGB Colours (up to 256 values each for Red Green and Blue combinations)

```
this.BackColor  = Color.FromArgb(100, 50, 10);         // set a RGB background for this form
```

Appendix A: The Basics

1. Console Input and Output

name = Console.ReadLine(); .. store input in a name variable (defined as string)
Console.WriteLine("I am " + name); .. output a message with text joined to a name variable
num1 = Convert.ToDouble (Console.ReadLine()); .. enter string and convert to a double
num2 = Convert.ToInt32 (Console.ReadLine()); .. enter string and convert to an integer

2. Variables

int count; .. define a variable called count to store an integer number
double num; .. define a variable called num to store a double (decimal) number
string name; .. define a variable called name to store a string (text or words)

3. Assignments to Variables (must be defined first)

count = 0; .. put 0 into the count variable (previously defined as int)
num = 5.67; .. put 5.67 into the num variable (previously defined as double)
name = "Fred Bloggs"; .. put these 11 characters in the name variable (defined as string)

4. Calculations

count ++; .. add 1 to the value of the count variable
count --; .. subtract 1 from the value of the count variable
count = count + 3; .. add 3 to value of the count variable (or use **count += 3;**)
count = count - 6; .. subtract 6 from the count variable (or use **count -= 6;**)
av = (num1 + num2 + num3 + num4) / 4; .. work our average of 4 numbers
tax = bill * 17.5 / 100; .. work out 17.5 percent tax on your bill

5. Loops (iteration)

a. The while loop

```
int count = 0;           // initialise a loop counter to zero

while (count < 10)       // continue while loop counter is less than 10
{
        Console.WriteLIne (“The count is " + count);   // repeated
        count ++;       // keep loop going by adding 1 to counter
}
```

an infinite loop

```
while (true)     // continue the while loop forever
{
        Console.WriteLIne ("Yippeeee!!");   // repeated forever
}
```

b. The for loop

```
// initialise loop counter; continue while count less than 10 ;  add 1 at end of loop

for (int count = 0; count < 10; count ++)
{
        Console.WriteLine (“The count is " + count);   // repeated 10 times
}
```

c. The do while loop

```
int  count = 0;            // initialise a loop counter to zero

do
{
        count ++;       // keep loop going by adding 1 to loop counter
        Console.WriteLine (“The count is " + count);   // repeated message
}
while (count < 10);     // continue while loop counter is less than 10
```

6. Selection

a. The if statement

```
        if (count == 4)        // if count is equal to 4
        {
                Console.WriteLine (“We are half way" );
        }
```

b. The if else statement

```
        if (count >= 4)        // if count is greater or equal to 4
        {
                Console.WriteLine (“We have reached half way" );
        }
        else
        {
                Console.WriteLine ("We are NOT half way yet");
        }
```

c. The switch statement

```
        switch(count)        // use count value to switch to various cases below:
        {
                case 1:                                    // i.e. if  count value = 1
                        Console.WriteLine (“We are just starting" );            break;
                case 2:  case 3: case 4:
                        Console.WriteLine (“We are on our way" );               break;
                case 4:
                        Console.WriteLine (“We are half way" );                 break;
                default:
                        // do nothing for any other values                break;
        }
```

7. Conditions

(a == b)	.. a is equal to b ?
(a > b)	.. a is greater than b ?
(a < b)	.. a is less than b ?
(a >= b)	.. a is greater or equal to b ?
(a <= b)	.. a is less than or equal to b ?
(a != b)	.. a is NOT equal to b ?

8. Multiple Conditions

(a == b **||** a == c**)** .. a is equal to b **OR** a is equal to c ?
(a == b **||** a == c **||** a == d**)** .. a is equal to b **OR** a is equal to c **OR** a is equal to d ?
(a == b **&&** a == c**)** .. a is equal to b **AND** a is equal to c ?
(a <= 100 **&&** a >= 0**)** .. a is less than or equal to 100 **AND** a is greater or equal to 0 ?

9. Classes, Objects and Methods

```
class Meal                    // define a class called Meal
{
        private string food;  // the class has one class variable (attribute or field)

         public static void Main()          // program starts executing here
         {
               Meal myMeal = new Meal();         // create a new myMeal object
              myMeal.getFood();                  // call the object's getFood() method
         }

        public Meal()                       // this is the Meal class constructor
        {
                food = "Fish and Chips";    // this sets the default food
        }

        public void getFood()            // define a method getFood()which returns nothing
(void)
        {
```

*// this defines a simple class **Meal** which has one variable, one method, one constructor*

10. Methods with parameters

- *this defines a method **setTax()** which has 1 parameter (amount) and <u>returns</u> a <u>double</u> value*
- *this method is defined inside a class e.g the Meal class above*
- *to use it, you can 'call' it like this:*
  vat = *myMeal*.**setTax(Bill);** *// assume myMeal is the object created from Meal*

```
public double setTax(double  amount)
{
      double  taxAmount;          // local variable
       taxAmount = amount * 17.5/100;
       return  taxAmount;
}
```

this passes the value of <u>Bill</u> to the method and picks up a returned tax value from it.

11. try/catch *(simple version : to trap errors or exceptions)*

```
try
{
        // enter instructions to be checked here
}
catch
{
        // error message display here
}
```

Appendix B:
Assessment of CO453 Application Programming

1. The module is assessed by coursework that consists of a series of directed study exercises and programming projects that must be recorded in a logbook.

2. The logbook must be an e-book and should contain your designs, algorithms, test plans, source code and results of your work.

3. The directed study includes **independent study tasks** and **programming projects**.

4. The **classwork** component of the directed study is assessed each week in your practical sessions. You must be observed doing the classwork in the computer laboratories during these timetabled sessions. You must record your classwork in a logbook and this will be presented for inspection at designated times. Your attendance and achievement will be recorded weekly.

5. The **independent** component of the directed study is your own unaided work. This work must be recorded in your logbook. The **independent** directed study is assessed when logbooks are submitted

6. The **programming projects** are your own unaided work and must be recorded in your logbook. They are assessed at various times during the timetabled practical session.

7. The assessed directed study is contained in several directed study packs.
   The weighting of the assessment to the final grade will be in the order of the following:

Independent Study 60%; and C# .NET Windows Project 40%

Log Books

For the module you should submit (unless otherwise instructed):

- Clearly numbered Task headings
- Simple description of task(s)
- Sample Screen Shots
- Fully commented Source code of relevant sections using highlighter pen to show added code where necessary
- Comments on problems encountered etc.

Grade related criteria for Programming - CO453

A	Where the student has demonstrated clear evidence of an excellent understanding of the theories and principles together with a high degree of analytical accuracy, good design skills, implementing fully tested solutions that show reliability, maintainability, readability and minimal complexity and correct form of presentation skills. *To acquire the knowledge and skills to demonstrate the above the student will normally be expected to attend the lecture and practical sessions and attempt at least 85% of directed study for each week.*
B	Where the student has demonstrated clear evidence of a good understanding of the theories and principles together with a good analytical ability, good design skills, implementing solutions that show reliability, maintainability, readability and minimal complexity and correct form of presentation skills. *To acquire the knowledge and skills to demonstrate the above the student will normally be expected to attend the lecture and practical sessions and attempt at least 75% the directed study for each week.*
C	Where the student has demonstrated a reasonable understanding of the theories and principles together with a reasonable analytical ability, design skills, implementing solutions that appreciate the need for reliability, maintainability, readability and minimal complexity and reasonable presentation skills. *To acquire the knowledge and skills to demonstrate the above the student will normally be expected to attend the lecture and practical sessions and attempt at least 66% of the directed study for each week.*
D	Where the student has demonstrated an understanding of the theories and principles of analysis, design, implementation and presentation skills. *To acquire the knowledge and skills to demonstrate the above the student will normally be expected to attend the lecture and practical sessions and attempt at least 50% of the directed study for each week.*
E	Where the student has made a genuine attempt to acquire the knowledge and skills but requires further application and study to demonstrate an understanding of the theories and principles of analysis, design, implementation and presentation skills. *In order to demonstrate a genuine attempt the student will normally be expected to attend the lecture and practical sessions and attempt at least 40% of the directed study.*
F	Where the student has clearly not acquired sufficient knowledge and skills and not attempted or coped with the directed study with any degree of competence regarding theories, principles, analysis, design, and implementation and presentation skills or where the student has NOT attended for assessment or where the student has copied work from an alternative source.

Module Name and code	Application Programming CO453

Staff: Richard Jones, Richard Mather, Carlo Lusuardi & Nick Day

Learning Outcomes:

- Analyse a simple requirement in a structured manner
- Design, document, implement and test reliable, maintainable programs as solutions to simple problems
- Use structured techniques of design and implementation and good documentation practice.
- Use software development tools.

Teach WK	Uni WK	LECTURE/TUTORIAL	PRACTICAL
1	18	**C# (Console) 4 Methods and Parameters**	C# Directed Study Unit 4
2	19	**C# (Console) 5 Arrays**	C# Directed Study Unit 5
3	20	**C# (Console) The Project**	C# Directed Study Project Unit
4	21	**Workshop week**	
5	22	**Windows C#1 Introduction & Splash Screen**	C# Directed Study Unit 1
6	23	**Windows C#2 SPS Game**	C# Directed Study Unit 2
7	24	**Windows C#3 Other .NET Controls**	C# Directed Study Unit 3
8	25	**Windows C#4 Multiform projects**	C# Directed Study Unit 4
	26-28	**Spring Recess – (Easter)**	
9	29	**Workshop**	
10	30	**Windows C#5 Animation**	C# Directed Study Unit 5
11	31	**Windows C#6 Graphics**	C# Directed Study Unit 6
12	32	**Windows C#7 The .Net Project**	C# Directed Study Unit 7
13	33	**Workshop**	

Course Texts:

Comprehensive Course Notes are provided

Bradley & Millspaugh, ***Programming in C#***, 2010, pub: McGraw Hill

Deitel & Deitel, ***Visual C# 2010 How to Program***, 2011, pub: Pearson