

Appendix B: The Basics of C# (Console)

1. 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)
    {
        Console.WriteLine("What would you like to eat?");
        food = Console.ReadLine(); // input into the class variable food
    }
}
```

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

10. Methods with parameters

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

*// this defines the method **setTax()** which has 1 parameter (amount) and returns a double value*

// this method will be 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
```

// this passes the value of Bill into the method and picks up the returned tax value from it.

Assessment of CO452 Programming Concepts

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:

100% Ceebot & C#, Independent Study and Projects - (Ceebot IS, 55%; C# Console IS, 30%; Ceebot Project, 15%; contribution.)

Grade related criteria for Programming - CO452

A	<p>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.</p> <p><i>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.</i></p>
B	<p>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.</p> <p><i>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.</i></p>
C	<p>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.</p> <p><i>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.</i></p>
D	<p>Where the student has demonstrated an understanding of the theories and principles of analysis, design, implementation and presentation skills.</p> <p><i>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.</i></p>
E	<p>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.</p> <p><i>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.</i></p>
F	<p>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, implementation and presentation skills or where the student has NOT attended for assessment or where the student has copied work from an alternative source.</p>

Module Name and code	Programming Concepts CO452
Staff:	Kevin Maher, Carlo Lusuardi, John Sadler, Nick Day, Based on original work by Brian Ward

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.

WK	LECTURE/TUTORIAL	PRACTICAL
1	INTRO to Ceebot, VARIABLES, INPUT and Output	Ceebot Chapters 1-6
2	ITERATION	Ceebot Chapters 7-8; 12-15
3	SELECTION	Ceebot Chapters 9-10
4	WORKSHOP for CW 1 Part 1 submission next week	
5	FUNCTIONS	Ceebot Chapters 18-19
6	PARAMETERS	Ceebot Chapters 20-21
7	ARRAYS	Ceebot Chapters 22-23
8	Ceebot PROJECT	Ceebot Chapter 24
9	WORKSHOP for CW 1 Part 2 submission next week	
10	C# 1 Input and Output	C# Intro Directed Study Pack: Unit 1
11	C# 2 Sequence, Selection, Iteration	C# Intro Directed Study Pack: Unit 2
12	C# 3 Classes, Objects and Methods	C# Intro Directed Study Pack: Unit 3
	Christmas Break	
13	WORKSHOP for CW 1 Part 3 submission next week	
14		

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