Stephen Wood

Notre Dame Catholic Sixth Form College | Computer Science Dept.

A-Level Computer Science

Year 1 – Programming Challenges

Coding Challenges – **Volume 3**

Coding focus: **Iteration**

Name:

Class:

**Coding Challenges – Volume THREE**

Table of Contents

[Program 1 – Sum of Natural Numbers. 2](#_Toc21948150)

[Program 2 – Display a range of Prime Numbers. 3](#_Toc21948151)

[Program 3 – Reverse the input. 4](#_Toc21948152)

[Program 4 – Display a right angle triangle of numbers. 5](#_Toc21948153)

[Program 5 – Display a full Triangle of consecutive numbers. 6](#_Toc21948154)

[Program 6 – Convert a binary number to denary. 7](#_Toc21948155)

[Program 7 – Access Denied 8](#_Toc21948156)

[Program 8 – Count the Capitals 9](#_Toc21948157)

[Program 9 – Strings in a String 10](#_Toc21948158)

[Program 10 – Most common letter 11](#_Toc21948159)

[**The Hard(er) Bit 12**](#_Toc21948160)

[Program 10 – Morse Code Encoder 13](#_Toc21948161)

[Program 12 – Twos Complement 14](#_Toc21948162)

[Program 13 – Hexadecimal Converter 15](#_Toc21948163)

# Program 1 – Sum of Natural Numbers.

Write a program in C# to display the n terms of even natural number and their sum.

E.g. Input number of terms: 10

The even numbers are: 2 4 6 8 10 12 14 16 18 20

The Sum of even Natural Number up to 10 terms: 110

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 2 – Display a range of Prime Numbers.

Write a program in C# to find the prime numbers within a range of numbers.

E.g. Input starting number of range: 1

Input ending number of range: 50

The prime numbers between 1 and 50 are:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47.

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 3 – Reverse the input.

Write a program in C# to display the number in reverse order.

E.g. Input a number: 12345

The number in reverse order is: 54321

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 4 – Display a right angle triangle of numbers.

Write a program in C# to display the pattern like right angle triangle with a number.

Sample Output:

1

12

123

1234

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 5 – Display a full Triangle of consecutive numbers.

Write a program in C# to make such a pattern like a pyramid with numbers increased by 1 each time. Utilise a FOR loop to do this.  
The pattern is as follows:

1

2 3

4 5 6

7 8 9 10

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 6 – Convert a binary number to denary.

Write a program in C# to convert a binary number into a decimal number without using an array, function and while loop.

**E.g.** Input a binary number:1000001

The Binary Number: 1000001

The equivalent Decimal Number: 65

Use the binary numbers 11001010 and 00110011 to test your program works.

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 7 – Access Denied

Write a program in C# that determines whether or not a user has entered a password correctly, the user will also only have a maximum of 5 attempts to do so:

* The password they will need to enter is: “C0mputer-Sc1ence”

You should test this program to prove it will deny access to any incorrect password numerous times, and that the user will only gain access when the correct password is entered.

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 8 – Count the Capitals

Write a program in C# that takes a single string and returns an ordered array containing the indexes of all capital letters in the string.

You should test this program to prove it will work for different strings, including; no capitals, 1 capital and many capital letters.

**E.g.** IndexOfCapitals("eQuINoX") ➞ [1, 3, 4, 6]

IndexOfCapitals("determine") ➞ []

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 9 – Strings in a String

Write a program in C# to find the number of times a substring appears in the given string.

**E.g.** Input the original string: **Welcome to** **Computer Science**

Input the string to be searched for: **put**

**The string ‘put’ occurs 1 times.**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 10 – Most common letter

Write a program in C# to find maximum occurring character in a string.

**E.g.** Input the string: Welcome to Computer Science

Highest frequency character: 'e'

It appears 5 times

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# The Hard(er) Bit

This is the bit where the spicy challenges live…

****

***Not officially endorsed by Nando’s…***

# Program 10 – Morse Code Encoder

Create a function that takes a string as an argument and return a non-encoded, encrypted string.

**Notes**

* Input value can be lower or upper case.
* Input string can have digits.
* Input string can have some special characters (e.g. comma, colon, apostrophe, period, question mark, exclamation mark).

**Examples:**

* **EncodeMorse("EDABBIT CHALLENGE")   
  outputs ➞ ". -.. .- -... -... .. - -.-. .... .- .-.. .-.. . -. --. ."**
* **EncodeMorse("HELP ME !") ➞ ".... . .-.. .--. -- . -.-.--"**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 12 – Twos Complement

Write a program in C# to convert a number that is entered into its negative (Twos Complement) version.

**E.g.** Input your number: **00110100**

Twos Complement Number is: **11001100**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 13 – Hexadecimal Converter

Write a program in C# that will convert Hexadecimal numbers into both denary and binary. The program should take in a Hex number (Max 2 Hex digits) and convert it into its Denary and Binary equivalent.

The program should repeat this process until the user asks the system to ‘STOP’. The program should be user friendly and robust.

**E.g.** Input your number: **A5**

Denary Equivalent is: **165**

Binary Equivalent is: **10100101**

Input your number: **3D**

Denary Equivalent is: **61**

Binary Equivalent is: **00111101**

Input your number: **STOP**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…