Stephen Wood

Notre Dame Catholic Sixth Form College | Computer Science Dept.

A-Level Computer Science

Year 1 – Programming Challenges

Coding Challenges – **Volume 4**

Coding focus: **Subroutines**

Name:

Class:

**Coding Challenges – Volume FOUR**

Table of Contents

[Program 1 – Count Spaces in any string. 2](#_Toc25917702)

[Program 2 – Sum of a user-defined Array. 3](#_Toc25917703)

[Program 3 – The Fibonacci Sequence. 4](#_Toc25917704)

[Program 4 – Prime Number Validation. 5](#_Toc25917705)

[Program 5 – Sum of the Individual Digits in a number. 6](#_Toc25917706)

[Program 6 – String Matching. 7](#_Toc25917707)

[Program 7 – Double Glazing Calculator (it’s back) 8](#_Toc25917708)

[Program 8 – To the Power of… 9](#_Toc25917709)

[Program 9 – Validating a string input. 10](#_Toc25917710)

[**The Hard Bit** 11](#_Toc25917711)

[Program 10 – Bubble Sort 12](#_Toc25917712)

[Program 11 – Recursive Numbers 13](#_Toc25917713)

[Program 12 – Recursive Array Reversal 14](#_Toc25917714)

# Program 1 – Count Spaces in any string.

Write a program in C# to create a function to input a string and count number of spaces are in the string.

E.g. Input String: Subroutines in a program

"Subroutines in a program" contains **3** spaces

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 2 – Sum of a user-defined Array.

Write a program using subroutines in C# to calculate the sum of elements in an array. The user will be able to enter a set amount of numbers into the array, that will be able to choose at the start of the program.

The total value should be returned from the function before being output.

E.g. Input amount of numbers: 5

Input numbers: 12, 14, 25.6, 63, 8

The total of your array is: 122.6

Program Code goes here…

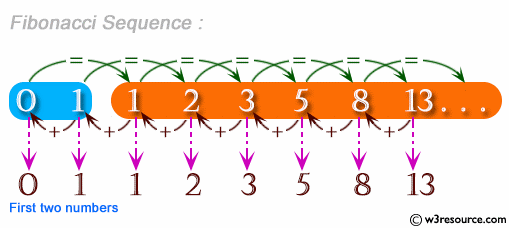
Test Evidence (screenshot of it working) goes here…

# Program 3 – The Fibonacci Sequence.

Write a program in C# to create a program using subroutines to display the ‘**n**’ (chosen by the user) number Fibonacci sequence.

**E.g.** Input a number: 10

The 10 Fibonacci numbers are: 0 1 1 2 3 5 8 13 21 34



Test the program by entering: **5, 10, 15 and 20**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 4 – Prime Number Validation.

Write a program in C# to make use of a function that will return a Boolean value which will test to see if a user input number is Prime or not.

The program should stop when no number is entered.

**Sample Output:**

Input a number: 23

23 is a prime number

Test the program using: **7, 15, 23, 47, 60, 99**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 5 – Sum of the Individual Digits in a number.

Write a program in C# to create a function to calculate the sum of the individual digits of a given number.

**Worked Example:**

Enter a number: 25

The sum of the digits of the number 25 is: 7

**or**

Enter a number: 747

The sum of the digits of the number 747 is: 18

Use the test evidence of: **45, 62, 118, and 4697**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 6 – String Matching.

Write a program in C# to take in two strings from the user, these will then be passed to another subroutine to determine if they are the same word.

The program will need to be case-sensitive.

**E.g.** Input word 1: Computing

Input word 2: Computing Output: True

Input word 1: Testing

Input word 2: TesTinG Output: False

You should use a variety of tests to prove this works.

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 7 – Double Glazing Calculator.

Write a program in C# that uses subroutines to calculate figures for a Double Glazing company, it will work in the following way:

* Subroutine 1 – will take in the user inputs, these will be the height and width measurements of the window.
* Subroutine 2 will be passed these values and it will validate them to be within the range of minimum 50cm, and a maximum of 300cm.
* Only when valid values have been supplied will it calculate the result.
* Subroutine 3 will calculate the result based on the validated input values it will output the total area of glass required and the length of wood to be used for the framework.

You should test this program to prove it will work for the following:

* **W – 45 H – 120**
* **W – 80 H – 150**
* **W – 120 H – 300**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 8 – To the Power of…

Write a program in C# that takes two whole number inputs from the user and uses these to calculate the first input to the power of the second.

You should use a **function** to collect and **validate** the number inputs so that they are correct and whole numbers. The calculated value should be returned to the Main subroutine to be output.

Test with the following:

Input (in this order; number followed by power):

**2, 10 then 10, 2 then 5, 7**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 9 – Validating a string input.

Write a program in C# to take in an input string from the user, the program should then pass this string to another subroutine to be validated.

The string entered must consist of only upper-case and lower-case letters, numbers and any of the following symbols: **#** **\_ - . @**

Secondly the string must be of a minimum 8 and a maximum of 20 characters.

The program should then finally be able to confirm to the user if the string they entered is valid or not.

**E.g.** Input String: **Sc1ence** Output: Invalid

Input String: **Comp$c1ence**  Output: Invalid

Input String: **Comp\_Sc1ence2019** Output: Valid

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# 

# The Hard Bit

This is the bit where the spicy challenges live…

****

# Program 10 – Bubble Sort

Write a C# Sharp program to take in an assortment of numbers from the user, this will stop when the user enters 0 (zero).

This List/array of numbers will then be passed to a function that will perform a Bubble sort and return the sorted list of elements to be output.

**Notes**

* Input value can be whole number or not.
* Input values must be valid digits, no characters or symbols allowed.
* Bubble sort must be written by yourself, **no cheeky use of a .Sort() method**, as this doesn’t necessarily use a Bubble Sort (they’re not really that efficient).

**Examples:**

**Original array : 3 0 2 5 -1 4 1**

**Sorted array : -1 0 1 2 3 4 5**

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 11 – Recursive Numbers

Before you start this spicy challenge, you will need to have a brief investigation into **Recursion** and how that works.

Write a program in C# to print numbers from n to 1 using **recursion**.

**E.g.** How many numbers to print : **20**

20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Program Code goes here…

Test Evidence (screenshot of it working) goes here…

# Program 12 – Recursive Array Reversal

Before you start this spicy challenge, you will need to have a brief investigation into **Recursion** and how that works.

Write a program in C# to take in 8 input strings from the user, it will store these in a List, the program will use another **recursive subroutine** to then output these in the reverse order to when they were entered.

**Tip.** Recursion makes this easier than you think…

Program Code goes here…

Test Evidence (screenshot of it working) goes here…