**U11.3 – File reading**

**LO1 – Read data from a text file in C# and manipulate its contents**

**LO2 – Read data from a csv file in C# and manipulate its contents**

**LO3 – combine the use of reading with multidimensional arrays to create cohesive solutions**

**Settler task**

Answer the following questions:

What is the memory heap and how does C# interact with it? [2 marks]

**The memory heap is a section of memory reserved for a program, the amount is variable and not known until runtime. C# automatically allocates objects to the heap, without the need for the programmer to manually allocate it.**

What is the difference between a primitive and a compound data type? [3 marks]

**Primitive data types are types of data that form the building blocks of structures. For example integers and strings, which themselves may be used to form a larger class. A class is therefore an example of a compound data type.**

A programmer wishes to use a dynamic data structure to store variables. State what a static data structure is and the advantages / disadvantages of using one. [5 marks]

**A static data structure has fixed amount of data that it can contain, whereas a dynamic data structure is capable of storing more or less data when it’s needed. Static data structures are more advantageous where the exact amount of date needed is known, as it can be used to create extremely efficient programs. Conversely, if the programmer does not know how much data is needed, a dynamic data type capable of storing more data than expected would prevent errors. Dynamic data structures also prevent inefficiencies where the amount of data falls short of the limit of a static structure, and space would otherwise be wasted. However, dynamic data structures could overflow if they reach the limit of the memory heap. Elements of a dynamic structure may be also difficult to access.**

Why would a static data structure be more useful to the programmer? Why not? [5 marks]

**It would be more efficient for the programmer if they know how much data would be used, so no space is wasted. It would also be easier for them to access a specific element they need, as its location will be known and won’t move unpredictably. However, a static data structure may not be useful to them if they do not know how much data will be used beforehand, a dynamic structure would be beneficial as the memory allocated can change accordingly.**

**Task 1**

Predict the output of the following code. Once predicted, get this code working on your own PC.

namespace FileReader

{

internal class Program

{

static void Main(string[] args)

{

string path = @"H:\C#\FileReader\sentence.txt";

string text = File.ReadAllText(path);

Console.WriteLine(text + " that I read from a file" );

}

}

}

The text file contains the words “This is a sentence”

Predict the output of the following code. Once predicted, get this code working on your own PC.

namespace FileReader

{

internal class Program

{

static void Main(string[] args)

{

string linePath = @"H:\C#\FileReader\lines.txt";

string[] lineText = File.ReadAllLines(linePath);

for(int i = 0; i < lineText.Length; i++)

{

Console.WriteLine(lineText[i]);

}

}

}

}

The text file contains:

These

Are

Individual

Lines

**Task 2**

Create the following txt file:

Dog

Cat

Map

Book

Flyer

Rhythmic

Read the file into a C# program line by line. From the words, create a guessing game. Each word is a different “round”.

Users should be able to choose what level they wish to play.

Users should be given a hint as to what the word is. If they enter correctly, “Well done!” is outputted and the game ends. If they guess incorrectly, “Try again” is displayed and the game is played again

**Extension:**

Use a 2D array to store your words by difficulty.

**Paste screenshots of your code here:**

**Task 3**

Change your text file from the previous task to look like the following:

Dog,Cat,Map

Book,Flyer,Rhythmic

Change your code from task 1

**Paste screenshots of your code here:**