130089872

**Assignment 1 – Report on programming concepts**

Contents

[Introduction 3](#_Toc55056029)

[Characteristics of programming 3](#_Toc55056030)

[Instructions and algorithms 3](#_Toc55056031)

[Programs (IDEs) 4](#_Toc55056032)

[Language Generations 5](#_Toc55056033)

[1st gen 5](#_Toc55056034)

[2ndgen 5](#_Toc55056035)

[3rdgen 5](#_Toc55056036)

[4th gen 5](#_Toc55056037)

[Programming syntax 6](#_Toc55056038)

[Command rules 6](#_Toc55056039)

[Variable declaration 6](#_Toc55056040)

[Standards: use of comments 6](#_Toc55056041)

[Code layout and indentation 7](#_Toc55056042)

[Data types and Data structures 7](#_Toc55056043)

[Variables 7](#_Toc55056044)

[Iteration and Logic 8](#_Toc55056045)

[Types of language 8](#_Toc55056046)

[Procedural languages 8](#_Toc55056047)

[Object-oriented 9](#_Toc55056048)

[Event-driven 9](#_Toc55056049)

[Reasons for choice of language: 9](#_Toc55056050)

[Bibliography 10](#_Toc55056051)

[Glossary 11](#_Toc55056052)

Table of Figures

[Figure 1 Set of instructions. 3](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056516)

[Figure 2 Output of the instructions. 3](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056517)

[Figure 3 The features of an IDE. 4](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056518)

[Figure 4 different forms of commenting 6](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056519)

[Figure 5 demonstration of an Integer 7](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056520)

[Figure 6 Example of Count-controlled loop 8](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056521)

[Figure 7 Example of condition controlled 8](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056522)

[Figure 8 Example of procedural language 8](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056523)

[Figure 9 Diagram of an Object-oriented system 9](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056524)

[Figure 10 Example of an event driven program 9](file:///C:\Users\dales\Dropbox\My%20PC%20(LAPTOP-D0497OJ7)\Desktop\Assignment%201–Report%20on%20programming%20concepts%20%20Dale%20Smallwood%20130089872.docx#_Toc55056525)

# Introduction

This document includes information on programming concepts.

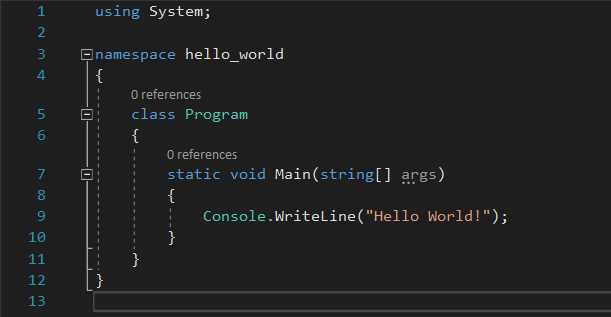
Included topics are,

* Characteristics of programming
* Programming syntax
* Data types and Data structures
* Types of language
* Reasons for choice of language

## Characteristics of programming

## Instructions and algorithms

Figure Set of instructions.

A program is a series of instructions that are given to a computer to perform a set task.

During the execution of the program, the data inputted is displayed into the desired output format that the user has inputted.

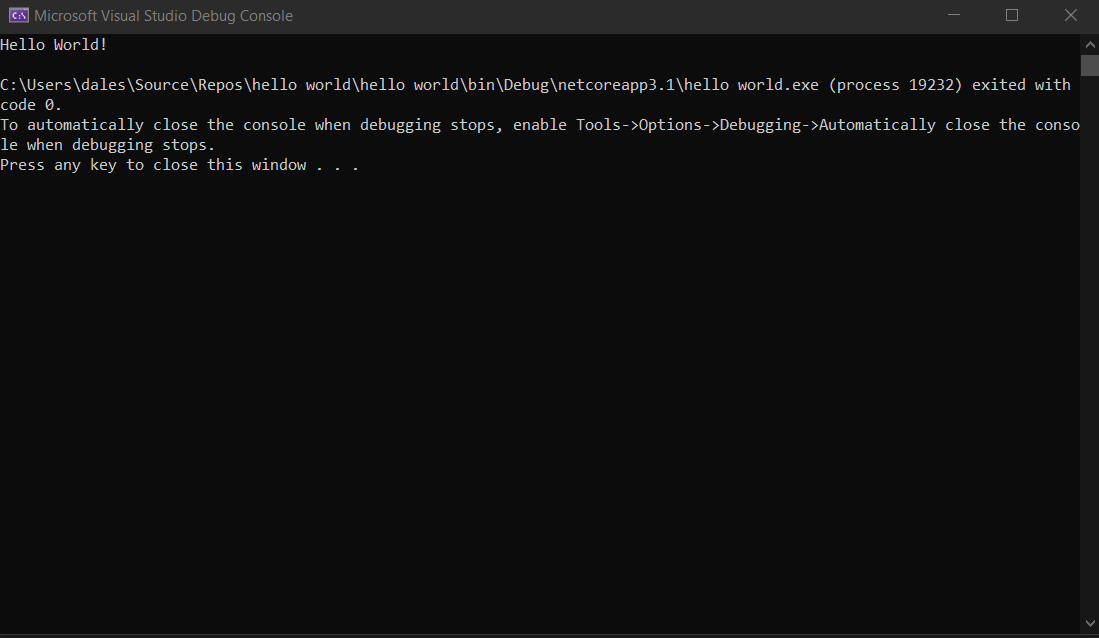
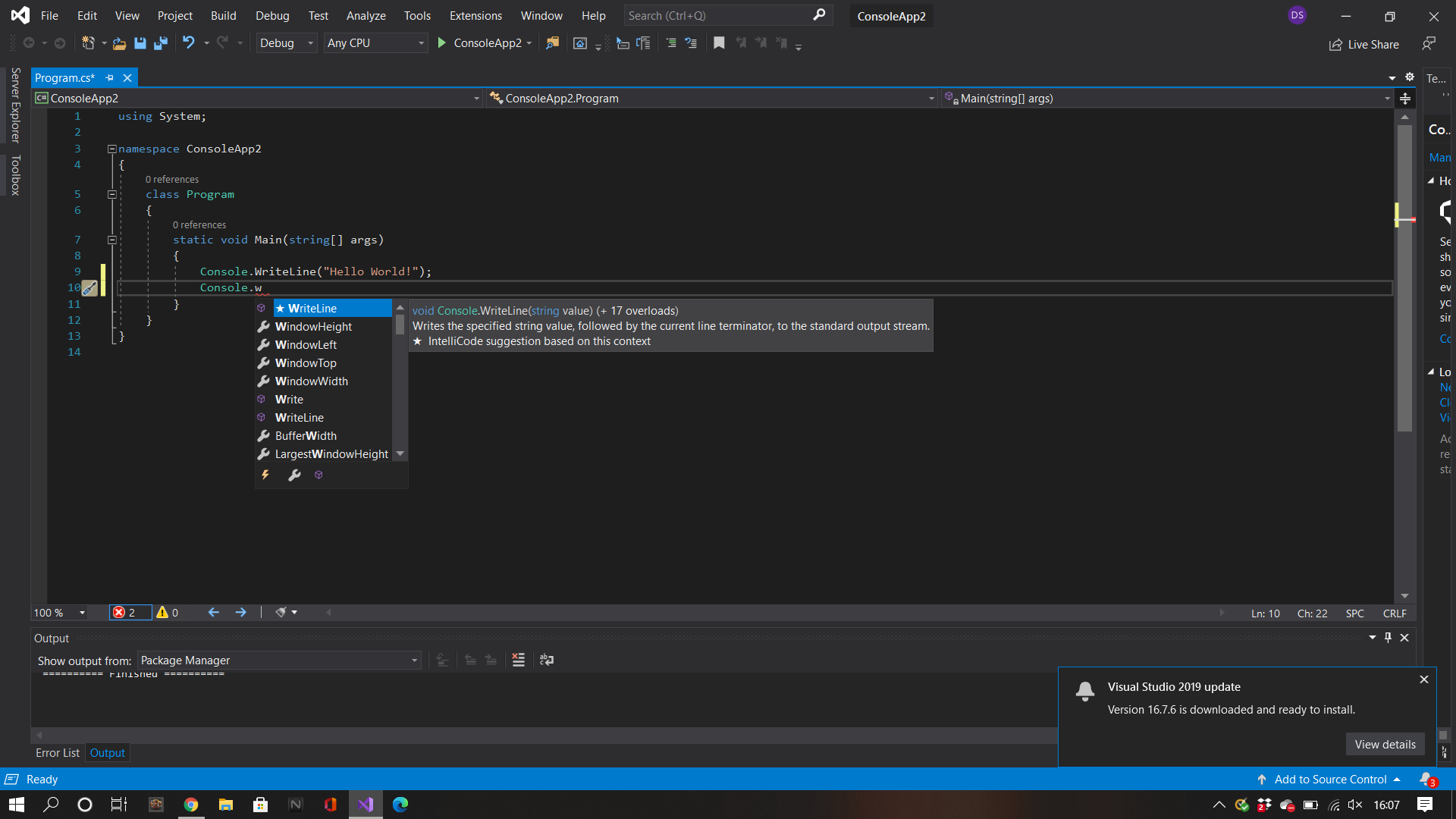


Figure Output of the instructions.

## Programs (IDEs)

An IDE is piece of software, that is downloadable from the internet. You can input your

high-level code onto the software to create a functioning program, that will also translate your code to a lower-level so your computer can interpret and execute the code.

Figure The features of an IDE.

It also has many other features such as:

* Code editor
* Auto-completion
* Bracket matching
* Syntax checks

## Language Generations

### 1st gen

Machine-level. This type of language was used to program the first computer. When creating the first-generation language, there was no translator used. The front panel switches on the computer system were used to enter the instructions. The instructions are made up of 1s and 0s. This is called binary code. The code is fast and efficient as the instructions are progressed by the CPU. But if an error does occur, the code will be difficult to fix. Machine level code is still used today and is now mainly used for hardware devices, drivers and interfaces with firmware.

### 2ndgen

(Gerard, 5 Nov. 2010)“Assembly languages are unlike first-generation programming languages in that the assembly code can be read and written more easily by a human. They require considerably more programming effort than high-level programming languages.”

This programming language is rarely used now though due to its many problems it was only really used to make game consoles such as the NES or the Commodore 64.

### 3rdgen

Programming languages such as C, C++, C# are all third-generation languages.

(Gerard, 5 Nov. 2010)

“Designed to be easier for a human to understand and include features such as named variables, conditional statements, iterative statements, assignment statements, and data structures.”

## 4th gen

(Gerard, 5 Nov. 2010)

“4GLs, are languages that consist of statements similar to a human language. Most fourth-generation languages are not procedural, and are often used in database programming.”

# Programming syntax

## Command rules

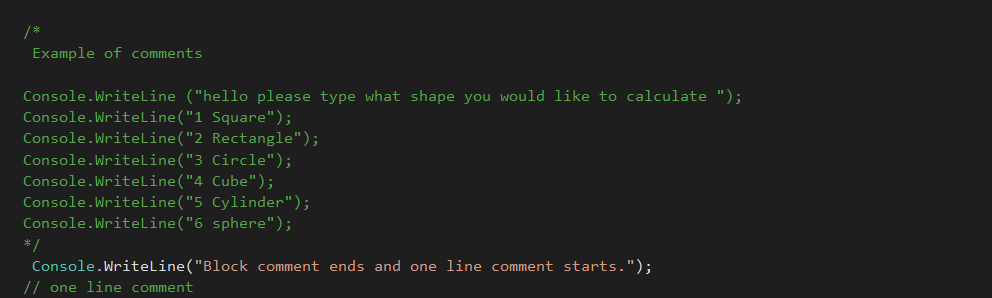
Also known as a statement, you can input into your code with codes such as print, write line.

## Variable declaration

It is extremely important to use appropriate variable names. Without this you could become confused with what the string does. For example, naming a variable button1 would be extremely confusing but using the variable name next page would be more useful, as you can easily distinguish what the variable will do.

## Standards: use of comments

Figure different forms of commenting

It is important to put comments within your code for multiple reasons.

Most projects are worked on in groups and are normally allocated on different sectors so you will need to explain what your lines of code do. You can do this wherever needed within the code and it will not cause complications. If done correctly you can do this by typing “//” for single line comments or “ /\* \*/”for a block of text or code that needs amending . (Figure 4)

## Code layout and indentation

(Martin, 01 Aug 2008)

“we indent the lines of source code in proportion to their position in the hierarchy. Statements at the level of the file, such as most class declarations, are not indented at all. Methods within a class are indented one level to the right of the class.”

The layout of the code is important as some formats of code will not function if not done correctly, for example python. Therefore, you need to indent your code correctly as well. It is also useful to do a technique called camel casing. This is where you would have the start of the first word as a lower case and the second to start with a capital. This helps when using a variable multiple time as it prevents errors, such as typing the incorrect variable name as its key specific and without doing this you cannot determine when a ‘wordEnds’

# Data types and Data structures

Figure demonstration of an Integer

## Variables

There are 5 different data types that can be used.

An integer / int, which is used for whole numbers (Figure5), String / Str, which is used for text.

Character / char, this only contains one letter such as A = 1, a float if for numbers that contain decimals, and a Boolean / bool is for true or false statements.

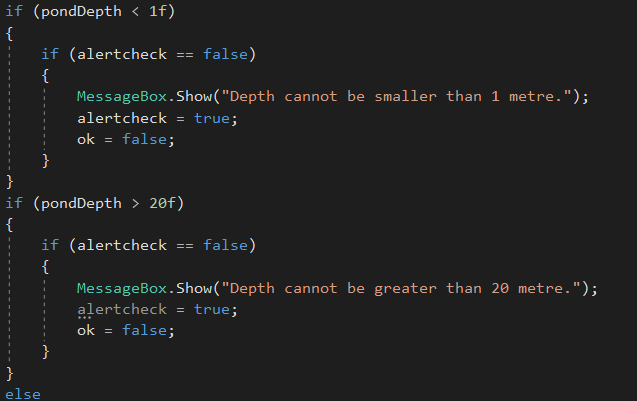
(Frieder Ophir, 2013) “A variable is a piece of data attached to a name. In most programming languages, variables are defined just as in algebra and can be assigned different values at different times. In a computer, they refer to a location in memory. “

## Iteration and Logic

Figure Example of Count-controlled loop

Also known as a loop, this is the process of a program that checks if a goal has been achieved. There are two types, **Count-controlled (Figure 6)** & Condition-Controlled (Figure 7).

Figure Example of condition controlled



# Types of language

Figure Example of procedural language

## Procedural languages

Using certain languages like pascal you need to structure your code differently for it to (Figure 8) function this is a procedure.

## Object-oriented

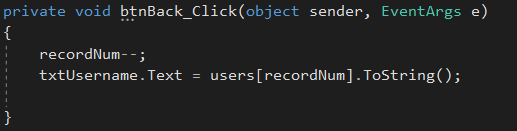
Figure Diagram of an Object-oriented system

This is a class system that uses methods and attributes (figure 9). This shows that a human must get food to be healthy.

## Event-driven

Figure Example of an event driven program

Figure 10 Example of an event driven program

Almost all devices are event driven. This is where an action or task such as a button click, or a key input is required for the action to proceed (Figure 10).

# Reasons for choice of language:

Programs mainly run on C# and C++ so it would be more beneficial for an organization to work alongside this language because, C# is a lot easier to learn than other languages. It’s both object and component oriented, as this language is so easy to learn it has a lot of popularity. Meaning there is a lot of potential staff interested in this field. C# is always evolving as it is developed by Microsoft and they use it on there programs so this language will not be left in the dust.

# Bibliography

Frieder Ophir, F. G. (2013). *Computer Science Programming Basics in Ruby.* Georgetown: O'Reilly Media. Retrieved October 26th, 2020

Gerard, O. ( 5 Nov. 2010). *A Brief History of Computing.* London: Springer; Softcover reprint of hardcover 1st ed. 2008 . doi:1849967253

Martin, R. C. (01 Aug 2008). *Clean Code: A Handbook of Agile Software Craftsmanship.* Cork: Prentice Hall . doi:ISBN-10 : 9780132350884

# Glossary

Code editor - where users input code also called the shell this allows you to edit and write

text and save when needed, also features that assist’s with the development of your code.

Auto-completion - works as a predictive text but predicts available code to save time and

unnecessary key inputs. it also displays a small explanation of what that line of code means.

Bracket matching – An auto-completion function but completes an opened bracket this allows ease of access aiding the user in minimalizing errors.

Syntax checks – Recognises errors and highlighting them so the user can notice them.