**Computation Thinking:** is the mental skill to break down a complex problem into segmented parts in order to develop a solution regarding finding a simpler way to tackle a issue or barrier that is preventing or reducing productivity or even the ability for something function.

**Decomposition:** Decomposition is a tool that consists of getting goals completed as soon as possible by looking at the most important tasks at hand when dealing with a to-do list and highlighting key details of the plan by using brainstorming as well. An example of this in the real world is the true minimalism of the tube map

One advantage of decomposition can be the fact that it helps you break down problems and further deconstruct any problems that are part of the issue. Another advantage of decomposition can be the fact that it saves a lot of time, this can be because of the simplification of any barrier that is put in place by the program or simply by the program making process. On the other hand, one disadvantage of decomposition can be that in order to decompose a problem you need to fully understand everything that is wrong from start to finish fully. Another disadvantage is that when a problem is broken to too many different parts it may be difficult for you to recombine everything back together with relevance.

This can have a positive impact on the program because it can reduce complications and further increase productivity when creating a program this can also have a positive impact on the future because the less time it takes to develop a solution for a problem there will be less troubles and unorganised complications. It also improves the overall quality of the product since it makes sure every part is digested by the team and taken care of individually or collaboratively since you could break it down and assign certain people to specific tasks further helping your team to complete the task as a group since there’s more focus on a single part rather than the project which also reduces stress from everyone’s point of view.

**Pattern Recognition:** Pattern recognition is the approach of identifying patterns by using automated process such as scripts that collect data over time which combines all the data together to give an overall clarification or statistical information based on the data received. An example of this in the real world is a Survey, this can be when the survey is finished all the data is combined in order to come up with a conclusion based on the results.

One advantage of pattern recognition is verifying data as it’s processed to reduce errors and recognise sequences or patterns that are used or that are attempted to use in order to reduce complications. Another advantage can be that you can look at the problem from a different angle further increasing number of solutions or ways to tackle a problem. On the other hand, a disadvantage of pattern recognition can be the fact that it is very time consuming and difficult to execute because it requires time to sort everything out. Another disadvantage can be the lack of clarification when a database isn’t large enough and won’t give a reasonable reason to why a pattern is the way it is if the number of patterns is dense.

This can have a positive impact on the program because it assists companies or individuals develop conclusions to said problems based on feedback, data entries or just everyday access from the public. For example, when a computer crashes the error codes will be sent to the company for them to analyse to cause of error. This can be useful because in the future there are likely certain adaptations in technology that will be new to people or machines that will be likely to have errors and when errors and cause or errors are combined this can lead to the root of the problem further decreasing the reoccurring problems. But on the other hand, the disadvantages of this can be the fact it is time consuming to wait for data to be compiled or even collected since not everyone will have the same errors or even same problems. But overall, this doesn’t improve the overall quality of the application because all it does it collect data from patterns or sequences.

**Abstraction:** Abstraction consists of getting goals completed as soon as possible by looking at the most important tasks while removing and irrelevant data so only data that is crucial can be store and processed, almost treating the problem a to-do list and only and highlighting key details of the plan by using brainstorming as well. An example of this in the real world is true minimalism of the tube map which reduces any confusion and shows relevant info to whoever wants to get from Point A to Point B.

One advantage of Abstraction is creating a rough idea of what a problem and how to tackle it which helps with brainstorming ideas in order to tackle it and just form an idea of how you’re going to solve the problem. Another advantage can be the ability to work with limited information but still be able to develop a solution independently due to brainstorming that can eventually lead to the solution. On the other hand, a disadvantage of abstraction can be that in some scenarios it may slow down the overall process and increase processing speed if you’re writing more lines of code in order to tackle a problem in software. Another disadvantage can be that it can make things harder to follow if the problem is too compressed because it sometimes may lack information and the solution may be completely dependant on the way you look at the problem and how you want to tackle it.

This may not always have an impact on the design of the program or quality of the final program since if you do not develop a good enough solution to a problem while using Abstraction often you might be working backwards and make yourself take longer to come up with a solution but if you do head in the right direction, it can make a vast difference.

**Generalisation:** Generalisation is a way of attempting to solve a problem based on previous solutions and comparing the one and another to each other and if there’s any relevance there could a chance that the same solution to an old problem may work on a current unsolved one. An example of this in the real world can be when you take a faulty device to repair shop, they will assess the situation and based on the problem they will put their finger down on one reason why the device is faulty based on their previous experiences with repairs on other devices.

One advantage of this can be decreasing the cost of development when dealing with a problem and instead of reaching for help from other professionals which may end up being quite costly you can compare the current problem to previous problems that you have already dealt with at no cost or lower cost. Another advantage can be reducing development effort since less critical thinking is done which just ends up being completed quicker. On the other hand, a disadvantage of this may be that if you haven’t dealt with a problem like the current one or if it’s completely new generalisation will be completely useless when dealing with type of problem. Another disadvantage of this may be that not everything can be generalised due to complications or due to the level of sophistication that that problem might have.

Generalisation may have an impact on the design of a program and the quality of the overall quality of the final program because it can assist you with maintainability, development effort and development cost which are really important factors when creating software. This can be a positive impact on the software design and quality because it provides ways to deal with problems in a less expensive labour and cost-effective process.

**Key Components:**

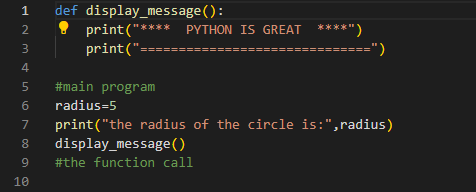
**Variable:**A variable in is a name that is a reference to an object that is a location to save data into binaries. It must start with an underscore but not a number and it must not be including any spaces but use underscores instead of spaces.

An Example is: Local Variable which is defined as a type of variable declared within programming block or subroutines. It can only be used inside the subroutine or code block in which it is declared. The local variable exists until the block of the function is under execution. After that, it will be destroyed automatically. A Global Variable in the program is a variable defined outside the subroutine or function. It has a global scope means it holds its value throughout the lifetime of the program. Hence, it can be accessed throughout the program by any function defined within the program, unless it is shadowed.

**Constant:** A constant is a type of variable which cannot be changed and usually used to declare and assign for different assignments.

An example of constants is Integer which is used when you only want a whole number (e.g., 4, 5, 10 and etc) to be inputted. Float which is used when you want a fractional number (e.g., 2.1, 8.4, 16.9 and etc) to be inputted. String is when you want a set of characters or a sentence (e.g., “Hello world!”) to be inputted. Boolean is when you only give two options of true or false (e.g., true or false).

**Key Processes:** Key processes are those that have maximum impact on the quality or successful outcomes of a product, service, or organization.

An example

**Repeated Process:** Repeated Processes are a list of commands that help a script run in a loop until the user manually stops it or has an automated stopping point in place by the programmer.

An example

Text

Description automatically generated

**Input:** Interact with or send information to the computer. An example of an input device can be a keyboard.

**Text

Description automatically generated**An Example

**Output:** An output device is any piece of computer hardware equipment which converts information into a perceptible form. An example of an output device is a monitor.

**Text

Description automatically generated**An Example

**Text

Description automatically generated*Key Components:*** Each component holds a sentimental value to the design of a program and the overall quality of the program due to the way each component cooperates with one another. For example, when writing a script each part comes together in order to form an application. This has an impact on the program design and overall quality due to how important they are to each other.

**Software for Storage – One Drive**

One drive is used in order to store or backup any type of data all the way from photos, videos, music or even important documents. It serves a big purpose in a lot of people’s lives from day-to-day tasks or even strictly backing up data daily, weekly or monthly. One Drive was made using C++ alongside the majority of other applications that Microsoft has created. There’s a lot of programs that are extremely similar to One Drive such as: Google Drive, Mega, Dropbox, Apple iCloud and a very large collection of others that peform very similary to One Drive with each program having its own benefits and drawbacks.

[6]

[5]

[4]

[3]

Graphical user interface, application, email

Description automatically generated

[8]

[9]

[7]

[2]

[1]

This is a screenshot of One Drive even though it might seem bloated with options to choose from its very straightforward and self-explanatory. [1] Is the list of directories that include files and the recycle bin where you can access any file that has been deleted up to 30 days ago. [2] Is the quick access tab which shows any directory or folders that you have recently accessed. [3] Is where you can upload or create new files within your One Drive folder. [4] The search bar is where you can search for any file or folder name that you have accessed, or you own within One Drive. [5] Is a gear which means the settings where you can access settings that are relevant to tweaking your One Drive to your liking. [6] The question mark is for the Help section where you can make or search for any queries regarding a concern or issue you might have. [7] Where you profile picture is displayed and be used to access your account settings or where you can sign out. [8] You can sort your displayed file and documents by newest, oldest, largest, smallest etc. [9] Takes the most space up since it displays your drive or the current directory that you are currently browsing.

**Software for Dangerous – Microsoft Windows Protector**

Microsoft windows protector is the preinstalled anti-virus used to protect your windows computer from any malicious malware and any threats to your data that is contained on your hard drive like ransomware malware. It serves a big purpose in every device that is used in today’s age due to the number of things that are now made digital there are a lot of people out for that data. Microsoft Windows Protector is made using C++ mostly since it is a Microsoft Program and just about every application Microsoft makes is C++. There are a lot of similar programs to Microsoft Windows Protector such as ESET NOD32, McAfee, Norton, AVG and a handful of others with their own pros and cons but the thing that other anti-virus has their downside is all great protection comes with a cost and this can be either annual or monthly.

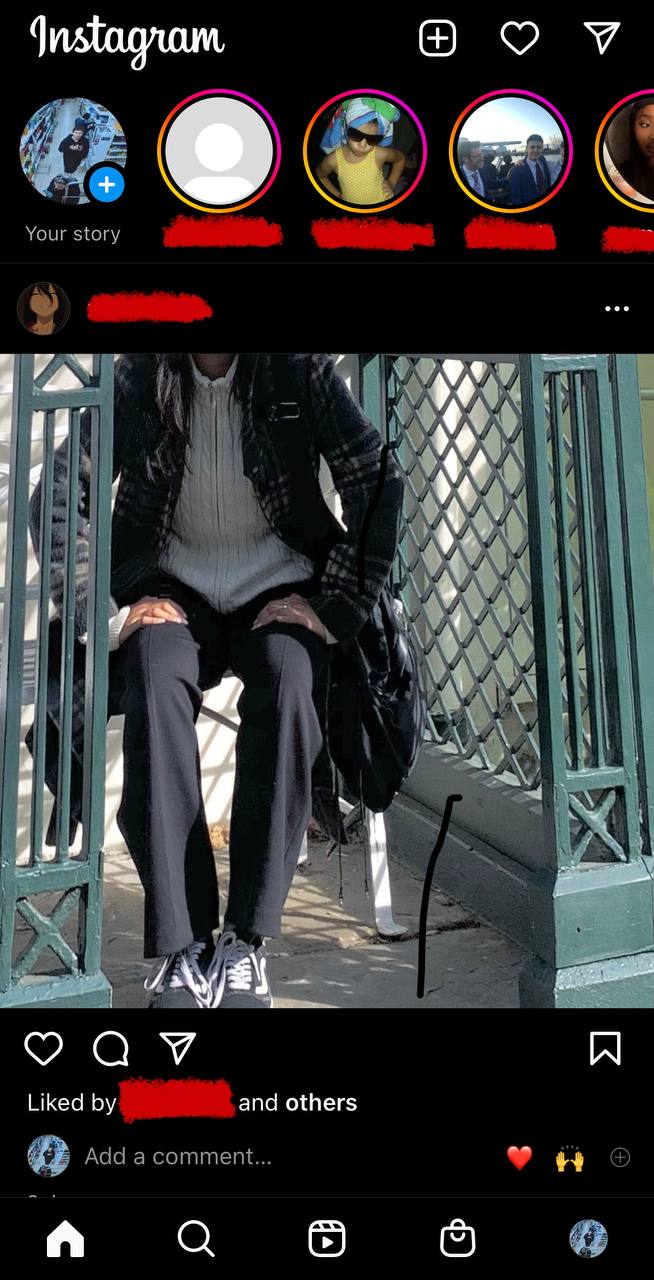
Graphical user interface, application

Description automatically generated

[1]

[2]

Windows Protector Defender keeps up with the minimalistic theme and keeps everything simple at eye’s reach without going to look for something on the search bar. [1] Is the list of the features that the anti-virus offers with small icons to keep it minimal as well as informative and including the settings at the bottom for any advanced users to navigate. [2]

**Software for Social Media - Instagram**

Instagram is a social media platform that is used to livestream and upload photos, videos to your feed or your story which unlike your feed it has a time limit of 24 hours to view. This program is used by around a billion users monthly, they can communicate with each other using DM’s also known as Direct Messages. Instagram was created using the programming language called python. There’s not a shortage of social media applications like Instagram such as: Facebook, Snapchat, Twitter and handful of others.

[4]

[3]

[5]

[1]

[2]

Instagram is also a beginner friendly user interface as you can see it’s very straightforward in what you want to do. [1] is where the story dashboard is where you can upload a story or view other people’s story. [2] This is where you can you see where someone has posted also known as the feed. [3] This is where you can like, comment, share or save a post that you have seen on a feed. [4] This dashboard where you can go home, search, reels, market and your profile picture where you can go to your profile. [5] At the top is where you can see the direct messages, liked and where you add a post to your personal feed.

**Software for Search Engine – Google**

Google is a search engine which was created in order to help you get results for any type of search you want to look for. Google properly has the biggest purpose out of all the software because google has been used or is being used to create software when researching for intel considering software design google will help you with any extra information. Google was made using Java Script mainly but use a handful of other programming languages in the back end. Software that is like google is another search engine such as duckduckgo, ask.com, Bing or even yahoo etc.

**Graphical user interface, text, application

Description automatically generated**

[1]

[2]

[3]

Google is the most user-friendly interface out of everything because all you must do is type a word or phrase and follow it up with enter. [1] At the top right is where you can find Gmail, images and a folder of all the other google websites bundle into one place alongside your profile picture. [2] Is the search bar which can be used by simply entering a phrase which is followed by results or you can speak into your microphone instead of typing or, where you can use your camera to take a picture of something which will bring up results related to that picture. [3] is where you continue to show the results or when pressing the im feeling lucky box your text from the search box will just take you directly to a website that is related to the sentence.

**Software for Productivity – Teams**

Teams has been used throughout lockdown in order to keep people connected and it also used to upload files and documents which helps collaborating. It serves a big purpose because it helped a lot of people be able to communicate when they can’t physically be in the same place at the same time. Microsoft teams was made using JavaScript which breaks the trend in Microsoft software because most of them are made on C++. Software that is like Microsoft teams are Google Meets, Zoom and handful of others that have each advantage and disadvantages.

**Graphical user interface, application

Description automatically generated**

[3]

[1]

[2]

Like the other software application this is also a simple interface. [1] Is where you can find your chats when you’ve recently sent a text to someone, calendar where you can track upcoming events such as meetings or assignments, assignments is where you can find every type of assignment you have due and have completed, activity is where you can find any notifications that you might of missed, calls is where you can find your recent call log, assist ai is where you can use ai to help you with your messages, the three dots are used to bring up more features alongside the apps section, help is where you can find any help you need that is related to the Team application. [2] This is where you can see the current groups you’re in. [3] This is where you can join groups or calls and where you can sort in which order you want to see your groups and view your hidden groups.

**Software for Entertainment – YouTube**

YouTube is a software for entertainment which commonly used be by half of the population and is easily the most influential piece of media to come out. YouTube serves a big purpose its today’s society because it can be very useful for a lot of things from pure entertainment to promotion for your personal product and a lot more. YouTube is programmed on JavaScript mainly but also uses type script. Currently YouTube is at the top of its league but there are other similar platforms such as Twitch, Rumble, Daily Motion and handful of other not as popular platforms.

Graphical user interface, website

Description automatically generated

[2]

[3]

[1]

[4]

YouTube’s main page is ram packed with a lot of content that is available for you to digest and this can vary from videos to even mixes of music YouTube has created based on the type of content you consume. [1] At this side bar is the basic features related to your account such as library and subscription which shows content from people you subscribe to. [2] At the top it contains the search bar, upload button, notifications and your profile where you can change settings based on your account. [3] Is where there is a list of categories for you to choose from based on your recent activity. [4] Finally is where all the videos are displayed for you to click on.

***Procedural Programming Language***

Procedural programming is breaking down a problem into sections in order to make it easier to solve problems due to them being segmented. The program data is in forms of variables and each function consists of computational thinking statements in order to solve a problem.

Text

Description automatically generatedAn example of Procedural Programming:

Examples of computer procedural languages are BASIC, C, FORTRAN, Java, and Pascal. Procedural languages are some of the common types of programming languages used by script and software programmers. They make use of functions, conditional statements, and variables to create programs that allow a computer to calculate and display a desired output.

***Object Orientated Language***

Object-oriented programming is a programming language paradigm (paradigm is a new method of thinking about a problem or situation). In OPP the code can be broken down and reused in a code where some properties or behaviours may be the same. This can be useful in improving the developer’s ability to quickly prototype software and just increase its life span and functionality since its being updated instead of being constantly remade.

Text

Description automatically generatedAn Example of Object Orientated Languages

Examples of object orientated languages are Java, C#, Python, Ruby, PHP and TypeScript. Object orientated languages are more advanced than procedural orientated programming and is commonly used by web designer for websites or even for databases for banks.

***Machine Language***

Machine Languages is a collection of binary digits or bits that a computer reads and interprets. Machine language is the only language a computer is capable of understanding. Machine language is mostly used for physical hardware for it communicate with each component and let them work together.

An example of Machine Languages:

Examples of Machine Languages are C++, Java, and Virtual Basic. A computer cannot directly understand the programming languages used to create computer programs, so the program code must be compiled. Once a program's code is compiled, the computer can understand it because the program's code is turned into machine language.

***Mark-Up Language***

Mark-Up Language is a computer language that consists of easily understood keywords, names, or tags that help format the overall view of a page and the data it contains.

Text

Description automatically generatedAn example of Mark-Up Language:

To create any markup language file, any text editor can be used. A markup language is not a programming language. It is special markings, interspersed with plain text, which, if removed or ignored, leave the plain text as a complete whole. Mark-Up is mostly used to create websites that’s why the heavy use of HTML.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Command Words | Apt, | Apt is used in linux in order to make the command do something | /safe | /safe is used to start the command prompt in safe mode |
| Constants | Integer | Used when inputs are whole numbers | Float | Used when inputs are numbers containing decimals |
| Variable | Binary | Yes or No outcomes | Nominal | Data with not organised structure |
| Local and Global Variables | def foo():  y = "local" | Local Variable which is defined as a type of variable declared within programming block or subroutines. | def foo():  print("x inside:", x) | A Global Variable in the program is a variable defined outside the subroutine or function. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Data Types – String | Name=str(input(“enter your name”)) | Data that is a collection of characters | Occupation str(input(“enter your occupation”)) | Data that is a collection of characters |
| Data Types – Integer | Num1=int(input(“enter an odd number”)) | Data that contains numbers with decimals | Num2=float(input(“enter an even number”)) | Data that contains numbers with decimals |
| Data Types – Real | Weight in KG | Data that has a unit | Height in cm | Data that has a unit |
| Data Types – Boolean | True or False | Data that is either true or false | Yes or No | Data that is either true or false |
| Data Types – Character | Upper or Lower Case | Data that is changed into lower- or upper-case letters | Lower Case or Upper case | Data that is changed into lower- or upper-case letters |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Assignment | int x = 5 | Adding an existing value | x := x + 1 | Gives an value existence |
| Input | Keyboard | Inputs letters into your computer | Webcam | Transmits image into a digital form |
| Output | Speakers | Outputs sound | Printer | Prints digital images into a physical form |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Sequence | Chronological order | Steps instructed to be performed | Step by step analysis | Combining everything together |
| Selection | OUTPUT “How old are you?”  age ← USERINPUT | List of Actions | IF age > 16 THEN  OUTPUT "You are old enough to drive a car"  ELSE  OUTPUT "Come back when you are older!"  ENDIF | List of Actions |
| Iteration | print | Repeated terms | input | Another repeated term |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Logical AND | score > 50 AND game\_over == False | Example that  evaluates to True | score > 100 AND user\_name == "Abdi" | Example that  evaluates to False |
| Logical OR | score > 50 OR user\_name == "Terri" | Example that  evaluates to True | score == 100 OR game\_over == True | Example that  evaluates to False |
| Logical NOT | NOT user\_name == "Terri" | Example that  evaluates to True | NOT score > 20 | Example that  evaluates to False |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Subroutines | If total=10<x  Print(“number is too big”) | This is known as the countdown subroutine | If total=15<=x  Print(“number too small”) | Only prints result you want |
| Functions | Print(“hello world”) | Something that is used frequently | X=str(input(“enter your name”)) | Another line of code that can be used frequently |
| Procedures | abc | A line of code broken down to make easier to understand | String over str | Shortened version of a command to simplify things |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| 2D Arrays | int sum = 0; | Makes finding a answer quicker by using shorter term | int count = 0; | Find solution easier |
| 3D Arrays | float y[2][4][3]; | High performance structure plans | [[[0, 0], [0, 0]], [[0, 0], [0, 0]]] | A square list of zeros |
| Splitting Arrays | While loop:    Print(“hi world”) | Separate code to make it look easier to run through | a=float(length.get()) b=float(width.get()) c=float(height.get()) | Making code more organised |
| Joining Arrays | const x = ["a", "b", "c"];  a.join(""); // 'abc' | Joining code together without space | const x = ["a", "b", "c"];  a.join(" + "); // 'abc' | Joining code with a plus |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Language Example | Language One Description | Language Example | Language Two Description |
| Open File | f = open("testing.exe", "r") | Opening a file | f = open("D:\\myfiles\testing.txt", "r") | Opening a file in a specific directory |
| Read from File | f = open("C:\Users\odise\OneDrive - Capital City College Group\School Work", "r") | Read from code | f = open("testing.txt", "r")  print(f.read(5)) | Read a specific line |
| Write to File | f = open("testing.txt", "a")  f.write("altered some stuff!") | To write files | f = open("texting.txt", "w")  f.write("I overwrote this work on monday!")  f.close() | To overwrite files |
| Close File | os.remove("testing.txt") | Delete a file | quit | Close a file |

**Iteration**

Iteration is when your device or software is told to repeat a set of instructions that were given by its creator in order to complete a certain operation. It will continue to repeat these instructions until there is a technical failure or theyre just an error. For example, when using python and you use a loop and the while true statement is true the program won’t stop working until it crashes.

**Mathematical Logic**

Mathematical Logic is the act of using formal mathematics in order to solve a problem by using things such as methods, theories or other ways of completing a mathematical solution. This can vary from actual number to problems. For example, when using python you will use a lot of math skills in order to write a script that is well structured such a car agency script which depends on more than and less than.

**Propositional Logic**

Propositional logic is the act of fundamental logic used when accessing any sort of problem and contains values such as true or false but never both. For example if you write a script to check if the number entered is over ten you will write an if statement which will appear true or false based on the statement given to it.

**Efficiency and Performance**

Efficiency is important because you might be using hardware that might be very limited, so your program must use the right amount of power from your device. Performance is also very important because software should never be jittery and full of lag because the purpose of using a piece of software is to make something easier not harder. Every piece of software uses a set amount of CPU, RAM and Disk Access due to how power demanding it is. For example, CPU can be used when rendering an image, RAM is used to store information that needs to be used quickly and Disk Access is used when the software is storing or retrieving information.

**Maintainability**

This is important to keep in mind when developing software because a program should be able to give you the option to make maintenance easy for you to perform. For example, instead of individually changing code you should be able to change a large amount of your code simultaneously.

**Portability**

This is important because portability is key when it comes to developers because they should be able to work on their script at any time and at any place in order to adjust on the move or just work without any compromises.

**Reliability**

This is important to keep these in mind when developing software because a program should always be fast and reliable whenever you want to complete a set due task. For example, when you run a program, you must be able to depend on it and it must work at least 95% of the time.

**Robustness**

Robustness is how well the application works with the user and if it is compliant with the users wants and needs. For example, all users might not use the same programming language in order to code so the software should give them a wide variety of choices when it comes to coding or just using a regular everyday program.

**Usability**

This is important because when a new program comes out or if it’s just an alternative of another program must be easy and friendly for new users to use. For example, if the applications are not well organised and not easy navigate the use then users will become frustrated. This can end up with negative feedback and an overall bad reputation on your behalf.