# Week 3 - Comprehension Questions

Week 3

1. **Definitions of:**

**Variables** - Labelled storage contains data that are stored in computer memory(RAM)
**Function** - An example of a sub-program - A block of call-able code that is often single serving and designed to do a certain task.

- A function is call-able in your code to execute its code.
- Functions are a name, you can create your own function
- Programming languages will often come with built-in functions.

**Comment** - A single line of human readable notes that is for the programmer or other programmers to read, both the interpreter and compiler will ignore comments in your code.

… →

**Q4:**

**Immutable data types** cannot be altered - They can only be updated with an entire new value or recreated
**Mutable data types** are often compiled of different components - The individual components that make up that connection can be changed and altered - Lists, sets, dictionaries.

**Q5:**

Programming languages have a finite amount of data it represents - To properly represent data in different contexts required we often need to change the data type from one to another.

- For example the input function(always takes a string) but to take a numeric value we need to convert it to something like int or a floor.

**Q6:**

Benefits of a module operator

1. When doing integer division, the modulus operator calculates the remainder
2. If you were to divide a number by two, if the remainder is 0 it is even, if its not its odd.
3. If you do A modules B and the remainder is 0 the B value is a factor of A

**Q7:**

Programming languages provide built in functions to prevent repetition

- Repeated code is built into languages to give simplicity to the programmer.

‘

**Q10: Difference between while loop and for loop.**

- While loop is a conditional based loop, as long as the condition attached is true it will continue to iterate until the condition is false.

- The for loop in python is called a for each loop, a for each loop is a traversal loop which is designed to traverse the individual items of a data set from start to finish.

Week 4

**Purpose of ASCII:**

The purpose is the following: Since computers only understand binary numbers of 0 and 1, humans have created a system to connect alphabetical characters, digits, and special characters to a group of binary numbers.

**Pixels:**

Every single pixel on an RBG screen follows a set of pixels - A pixel is a set of colours in RGB format to represent a colour. Each value can have a value of 0 to 255 which measures the intensity of that colour

- To represent the intensity we need 8 bits of data, and since each colour needs 8 bits of data we have 24 bits.
- The reason why hexadecimals are useful is because - To present 24 bits of data we only need 2 hexadecimals - 2 hexadecimals per colour - Therefore we only need 6 hexadecimals in total to represent a colour.

**Q7: Difference of compiler and interpreter**

A compiler translates the entirety of the code into something called a byte code or machine level code and then the computer executes the code.

An interpreter translates the line of code one by one into machine code, executes it → then moves on to the next.

**Q10: Why a program would need to be updated.**

The programming language was updated therefore you need to use new versions of code.
The program has a bug or an error → it is not running as intended, so it needs to be updated with the proper fixes.
Refactoring → You found a more optimized version of your code so you update your program with the new changes.

**Week 5:**

## Q2: Reason for translation process

Computers only understand machine level code, for the computer to execute the code, we must translate the code from high level code to machine code.

Programmers only understand high level code and it's much easier, which is why we have it.

## Q3: A & D of sequential search and binary search

**A of Linear search**: It is not dependent on a data set → It can have different types of data. Data type does not need to be consistent.
**A of Linear search**: The data set does not need to be sorted.
(linear search is sequential search)
**D of Linear search**: As the data set growths, the linear search gets very inefficient.

- The reason why it becomes inefficient: The Number of comparisons we have to make is directly related to the number of data items we have.

**A of binary search**: Extremely efficient and fast

- Because it continuously divides the data set into halves.

**D of Binary search**:
1. The data set has to be sorted
2. Each item within the data set must be comparable to each other.
Differences: Linear search has a big o notation of: o(N), binary is o(logn).

**Q4:**

**Design consideration 1:** Avoid using a while loop, use a for loop instead. Because a while loop does a conditional check at every iteration. → Therefore a for loop is more efficient.
**Design consideration 2:** When searching is involved, use data sets like: sets(hashmaps/dictionaries), Insteads of arrays or lists → Because searching is faster with hashmaps.
**Design consideration 3:** Finding early exit points in an iteration can speed up our algorithms.

- For example: when finding factors of a number it was faster to end at square root of n rather than going up to n.

**Q5**
-Bubble sort always creates pairs.
-Selection sort:
1. Has an empty list, bring the smallest one over and remove it.
2. Start at the left most value then swap with the smallest value. Then we go to index 1, find the second smallest value and swap it with the second smallest.

Insertion sort however: The very first value is considered to be sorted. Then we look at the second value and swap if it's not sorted.