Exercise 3

a)

In a general sense, abstraction is a way of handling complexity within computer science. We reduce complex processes down to their essential features so we can understand the general idea and how to interact with them without needing to understand every aspect to that process or how it actually occurs within the hardware. Fundamental types of abstraction within programming are control abstraction and data structure abstraction.

Data structure abstraction involves hiding details about the data as it is held within the system and presenting it in a human readable way. For example, for a given dictionary we may not be interested in the exact structure of the dictionary in memory but only what each value represents and what the relevant key would be for that value.

Control abstraction involves a similar process for methods that occur within the machine. For example, loops may involve many jumps from different locations within memory and then the updating of values within the loop. This may not be relevant for the programmer but only the way in which they can interact with the loop is important. Abstraction is the process of hiding this implementation and only representing that which is relevant.

b)

The syntax of a programming language is how to structure commands so that the computer understands the desired action. For example, when making calculations a programmer needs to know WHERE to place the relevant symbols i.e. +, -,*, /, = and the variables to create a valid expression. Semantics on the other hand denote the meaning of the symbols used and how certain meanings are appropriate within the other semantic meanings in the expression. For example, a sentence like 'Arianna plays flight' is syntactically correct but the meaning of the noun flight is incorrect alongside the other semantics within the sentence.