OCR Computer Science A Level

2.1.1 Thinking Abstractly

Concise Notes

Specification:

2.1.1 a)

- **The nature of abstraction**

2.1.1 b)

- **The need for abstraction**

2.1.1 c)

- **The difference between abstraction and reality**

2.1.1 d)

- **Devise an abstract model for a variety of situations**

The nature of abstraction

Representational abstraction

- Removing excessive details to represent a problem using only the key features
- Must analyse what is relevant to a scenario and simplify a problem based on this

Data abstraction

- Details about how data is being stored are hidden
- Programmers can use data structures without knowing how they are implemented

Layers of abstraction

- Large, complex problems are split into layers of abstraction
- Each layer has a different role, with the highest layers being closest to the user
- These are usually responsible for providing a user interface
- The lowest levels perform tasks such as interacting with machine components

Abstraction by generalisation

- Grouping together similarities within a problem to identify what kind of problem
- Allows problems to be categorised as being of a particular type
- A common solution can be used to solve these problems

Procedural abstraction

- Allows programmers to utilise functions without knowing how they are implemented
- Used in decomposition and manipulating data structures
- Models what a subroutine does without considering how, as once a subroutine has been written, it can be reused as a black-box

The need for abstraction

- Abstraction allows non-experts to use of a range of systems or models by hiding information that is too complex or irrelevant to the system's purpose
- Enables for efficient software design as programmers can focus on core elements rather than unnecessary details
    - Reduces the time spent on a project
    - Prevents a program from getting unnecessarily large
- Programming languages use layers of abstraction:
    - Low-level languages directly interact with computers but are difficult to write
    - High-level languages abstract the machine code that is executed when a program is run by providing easy-to-use syntax similar to natural language
    - Makes developing programs easier
    - High-level languages are easier to learn and use than assembly language or machine code
    - Makes coding accessible to non-specialists
- The TCP/IP model is an abstraction for how networks function, separated into four layers: application, transport, internet and link
    - Each layer deals with a different part of the communication process
    - Each layer does not need to know how other layers function

The difference between abstraction and reality

- Abstraction is a simplified representation of reality
- Entities are represented as computational structures eg. tables and databases
- Real-world values can be stored as variables and constants
- Objects in object-oriented programming are an abstraction for real-world entities
    - Attributes represent the characteristics of an object
    - Methods represent the actions a real-world object is able to perform

Devise an abstract model for a variety of situations

When devising an abstract model given a scenario, you must consider:

- What is the problem that needs to be solved by the model?
- How will the model be used?
- Who will the model be used by?
- Which parts of the problem are relevant based on the target audience and purpose of the model?