Design Notations

Be able to describe

- Structure Diagrams
- Dataflow Diagrams
- Pseudocode
- Unified Modelling Language (UML)
  - Use Case Diagrams
  - Class Diagrams
  - Sequence Diagram
- Wireframing
A structure diagram will be created as part of the **top-down** analysis of the software specification.

This allows the developers to break this complex problem description into a series of smaller sub-problem descriptions.

These sub problems can be regarded as modules within the system and they themselves may be further divided into smaller (and hopefully simpler) sub problems.

A structure diagram is organised to show the **level or hierarchy of each sub task**. The sequence of operations in the program is read from top to bottom, going left to right.

This problem can be broken down into four main modules, two of which can be further broken down into sub tasks.
The data used by the different modules can be represented by a data flow diagram.

Here the modules are shown as circles and the physical devices are shown as rectangles.

Note that not all data is needed by every module.

Like a structure diagram, a data flow diagram reads from left to right. A data flow diagram can be used to decide on what data needs to be passed as parameters to each module.
Pseudocode is often referred to as a "halfway house" between an algorithm (written in English) and final code.

Pseudocode is useful to show the detailed logic within modules.

Pseudocode is a method of describing an algorithm in a way which makes it easy to understand and ultimately convert into source code in whatever language the programmer wishes to use.

Pseudocode is often described as an informal description of an algorithm, and can be written in a wide variety of styles.

The SQA has created a more formal 'Pseudocode' called Scottish Reference language. The formal description can be found [here](#).

It is possible to build a syntax checker for any algorithm written using it. This is not normally an option for pseudocode. Because of its formal nature, Scottish reference language can be syntax checked [here](#).

```plaintext
PROCEDURE linearSearch(ARRAY OF INTEGER numbers, INTEGER highestIndex)
    DECLARE itemToFind Initially getValidItem(1, 100)
    DECLARE found as BOOLEAN Initially false
    DECLARE counter Initially 0
    REPEAT
        SET found TO numbers[counter] = itemToFind
        SET counter TO counter + 1
        UNTIL found OR counter > highestIndex
    IF found THEN
        SEND itemToFind & " found at position" & (counter - 1) TO DISPLAY
    ELSE SEND "Item not found" TO DISPLAY
    END IF
END PROCEDURE
```
Design Notations

UML – Unified Modelling Language
(object orientated languages)

- Created to help programmers design systems in OOP
- Formal system for creating modules and defining different types of diagram needed in the creation of an information system
- Like Pseudocode, UML is language independent
- More details [here](#)
Use case diagrams contain **Actors** (the people or entities who interact with the system) and **Use Cases** which are the procedures which they interact with.

The Use Cases in the diagram are identified by the circled text and the Actor's interactions with them are identified by the arrows.
Class diagrams show the classes of objects needed by the system and how they are related.

Each Class object will describe instance variables and methods for that class.

- `flightNumber : Integer`
- `departureTime : Date`
- `flightDuration : Minutes`
- `delayFlight (numberOfMinutes : int) : Date`
- `getArrivalTime () : Date`
We can show the relationship between the Playlist, Track and HendrixTrack classes.

The Track class is associated with the Playlist class. The HendrixTrack class inherits its properties from the Track class.
Design Notations
UML – common diagrams
Sequence Diagrams

- Shows interactions between objects in the system
- Solid lines show calls, dotted lines data returns
Design Notations
Wireframing (User Interface)

- Primarily used to design the User Interface
- Show layout of text, graphics and navigation structures
- Can be a rough labelled sketch or a much more complicated design using a wireframe authoring application
- Helpful for producing prototypes and including clients in the design iterations