

Section 3 - Design

The WJEC specification says:

- a)** Input and output: Specify, design and document screen layouts, reports and other forms of input and output required to create the user interface.
- b)** Data structures and methods of access: Design and document all data structures that will be required to produce the output for the solution to the problem together with the method of accessing the data in that data structure. Ensure that all data entered into the system is valid.
- c)** Processing stages: Design programming routines to be used to handle and process data within the proposed solution to achieve each objective. Document these designs using a structured convention such as pseudocode.

Things to consider doing:

- i)** Put a heading, **3a Input and output**.
- ii)** You need to design all the different screens your user uses to enter data. You then need to design all the screens your user looks at to get information from. You then need to design anything that comes out of the printer, for example, reports. A great way to design the screens is to start by drawing an outline 'map' of the whole system, showing the different input and output screens and reports and how they are interconnected using the menus, sub-menus and buttons. This will show someone clearly and on one diagram how the whole system fits together. Then start designing each screen individually. You teacher will explain what you need to consider, but consistency is very important; specify and use the same fonts, colour scheme, spacing, layout, logo and so on.
- iii)** Put a heading, **3b Data structures and access**.
- iv)** State what data structures you need and why you need them. What data will they hold? Specify the data each will hold and any validation rules that should apply to the data. How will the data in each structure be accessed? The easiest way to provide the evidence for this is draw up a table with relevant headings.
- v)** Put a heading, **3c Algorithms**.
- vi)** You should identify those parts of your solution that need coding, ideally in functions. You should then write algorithms for each of those functions. You can use whatever method is easiest for you to write the algorithms, either pseudo-code or flow diagrams, for example. You could start by simply writing out in short English statements what you want the code for a function to do, and then turn your statements into something more akin to pseudo-code.