Component 2 | 2.1.3| Thinking procedurally

Specification & learning objectives

	A Level	Specification point description
	2.1.3a	Identify the components of a problem
	2.1.3b	Identify the components of a solution to a problem
	2.1.3c	Determine the order of the steps needed to solve a problem
	2.1.3d	Identify sub-procedures necessary to solve a problem

Resources

PG Online textbook page ref: 268-271
Hodder textbook page ref: 15, 35
CraignDave videos for SLR 20

Identifying components of a problem

When approaching a problem it is better to break it into smaller sections. For example, a stock control system would be too difficult to investigate as one whole. It is much better to investigate each stage and then build the whole picture from those blocks. This is the processes of identifying the components of the problem.

Often at the end of this analysis the developer will have a clear idea of the different parts of the problems that need to be solved. They can then be the priorities, which are allocated to different teams and thus a development plan is produced.

Identifying components of a solution

When a problem has been broken down into components it is often the case that each of those component will need to be broken down into a series of parts to provide a solution.
A component of a problem would be to gain input from a user.

The actual solution may have several parts to it. There may be the different methods by which the input is received (e.g. barcode reader with keyboard entry). The entry must also provide information to the user about what is needed. The interface must be clear and easy to use.

Data entered must be validated to ensure that it can be processed correctly.
It can be seen that each of these components could then also be solved independently and then combined together to build a solution to that first component.

Determine the order of steps needed

Once the steps needed to solve a problem are identified, it is important to work out which steps need tackling first. Most tasks are allocated to programming teams so a priority must be planned.

It is also possible that a particular part of the problem will take longer to solve than others so it would make sense to start that task with one team while another team develops other parts so that when the parts are put together there is no delay in waiting for a particular part to be solved.

This problem can be made even more difficult if a part requires another solution to be solved before being tackled.

The project manager is responsible for allocating these tasks and it is very important to balance the order of tasks to minimise wasted time.

Identifying sub-procedures

As mentioned previously, a solution may have several components to one component of a problem. Equally, each solution component may have several sub-procedures in order to solve that problem.

Using the previous example, the act of validating the data entered could involve a procedure to convert a string input into a code, another routine could then take that code and break it into parts, which another set of routines could then validate to ensure that it matches the correct format.

Typical exam questions

Consider the following algorithm, expressed in pseudocode:

```
Sub Fibonacci (Max)
    n1 = 1
    n2 = 0
    For counter = 0 to max
        Output(n2)
        n2 = n2 + n1
        n1 = n2 - n1
    Next
End Sub
```

The code outputs a Fibonacci sequence of numbers: 0,1,1, 2, 3, 5, 8, 13, 21, 34, 55.
The next number is found by adding up the two numbers before it. This can be expressed as: Fn = Fn-1 + Fn-2 where F0 = 0 and F1 = 1

1. Write a similar algorithm using recursion that only returns the final result. E.g. Fibonacci(10) = 55.
Annotate your pseudocode to show parameter passing and a terminating condition. **[6]**

Component 2 | 2.1.3| Thinking procedurally

Target: Overall grade:

Minimum expectations & learning outcomes

☐	You must illustrate how a problem can be expressed using a flowchart.
☐	Using the classic game of battleships: Identify the component parts of the problem by drawing a structure diagram for an implementation of the game.
☐	You must illustrate the board and pieces the game would use for context.
☐	Write the pseudocode for at least two component parts of the solution.
☐	Answer the exam questions.

Feedback

Breadth	Depth	Presentation	Understanding
☐ All	☐ Analysed	☐ Excellent	☐ Excellent
☐ Most	☐ Explained	☐ Good	☐ Good
☐ Some	☐ Described	☐ Fair	☐ Fair
☐ Few	☐ Identified	☐ Poor	☐ Poor

Comment & action required

Component 2 | 2.1.3| Thinking procedurally

Reflection & Revision checklist

Confidence	Clarification
☹ 😐 ☺	Candidates need to be able to deconstruct a program and identify the component parts that will make it up, for example listing the parts or completing a structure diagram.
☹ 😐 ☺	Candidates may be given some component parts and be asked to complete these from a written description or pseudocode for a program.
☹ 😐 ☺	Candidates need to be able to identify the steps that will need to take place to complete the algorithm or program and be able to write these in a suitable format, or put a given list into the correct order to produce a working program.
☹ 😐 ☺	Candidates may need to write pseudocode or draw a flow chart to show this sequencing of steps.
☹ 😐 ☺	For a given scenario, candidates need to be able to identify where sub-procedures may be used, and then write appropriate pseudocode for these sub-procedures, making use of parameters where appropriate.
☹ 😐 ☺	Candidates may be given a structure diagram that they will need to interpret, or complete to identify these sub-procedures.