Section 12.2 - Program Design

Layer 7: Application

Syllabus Content Section 12: Software Development

S12.2.1 Use a structure chart to decompose a problem into sub-tasks and express the parameters passed between the various modules/procedures/functions which are part of the algorithm design

- Describe the purpose of a structure chart
- Construct a structure chart for a given problem
- Derive equivalent pseudocode from a structure chart

> A structure chart:

- Purpose: used in structured programming to arrange program modules, each module represented by a box
- Tree structure visualizes relationships between modules, showing data transfer between modules using arrows.
- Example of a top-down design where a problem (program) is broken into its components.

This is where it gets annoying.
There is not one set way to draw a structure chart and Cambridge has not given us any guidance on it.

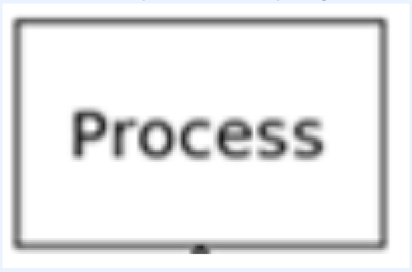
But there are some general rules that most people follow.
It's a top-down design (Start from top and work down)
Uses boxes for processes and functions and modules
Uses arrows to show how data moves from one module to another. These arrows are called "data couples"
Uses a different arrow to show if data is being checked. These arrows are called "flags" or "switch"

S12.2.2 Show understanding of the purpose of state-transition diagrams to document an algorithm

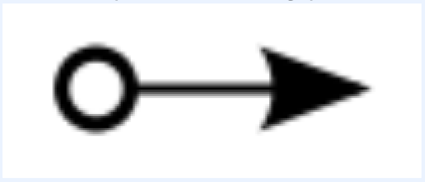
---

> Rules

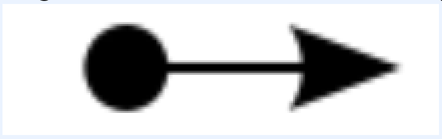
Process: Represents a programming module e.g. a calculation



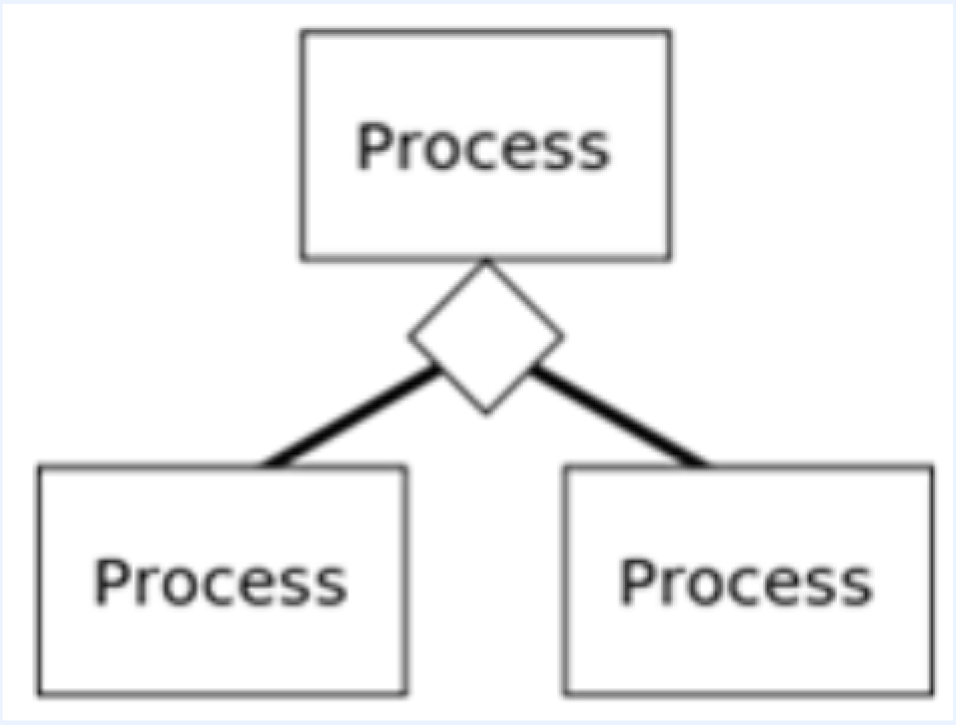
Data couple: Data being passed from module to module that needs to be processed



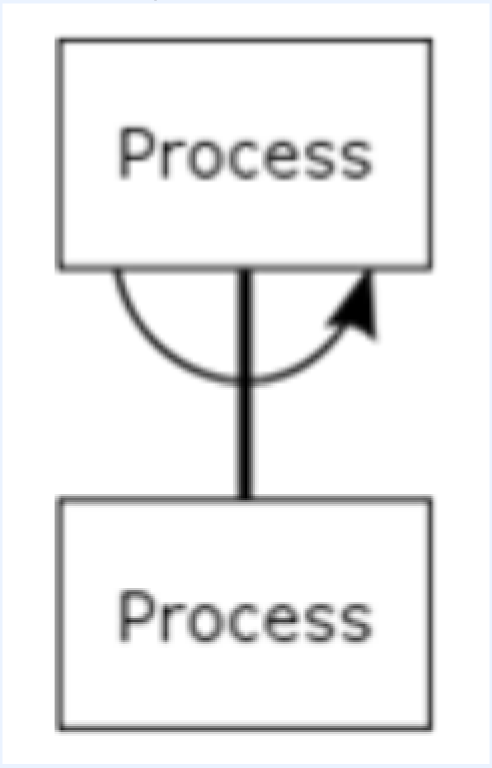
Flag: Check data sent to start or stop a process. E.g. check if data sent in the correct format



Selection: Condition will be checked and depending on the result, different modules will be executed



Iteration: Implies that module is executed multiple times


Process
Process