

Exercise walk-through

Program Design ■ 12.2

eXercise

You must be able to

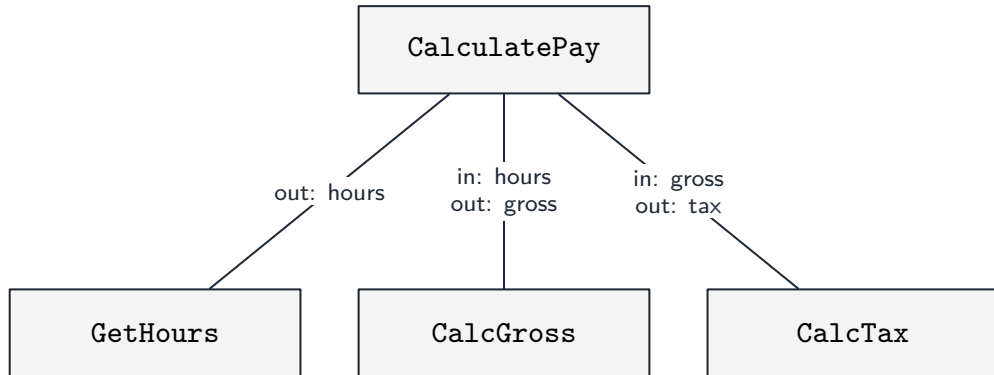
- read and draw a **structure chart** showing modules and the **parameters** between them
- read and draw a **state-transition diagram** showing states and events

Method — Structure chart

A **structure chart** shows how a program is **decomposed** into modules, and the **parameters** passed between them. Each module is a box; a line joins a caller (above) to a module it calls (below); the labels show data going **in** (down) and results coming **out** (up).

You can read each module's **signature** (its parameters and return) off the chart. Two extra symbols may appear: a **curved arrow** over the links means those modules are called repeatedly in a **loop** (iteration); a small **diamond** on a link means the module is called only **sometimes** (selection).

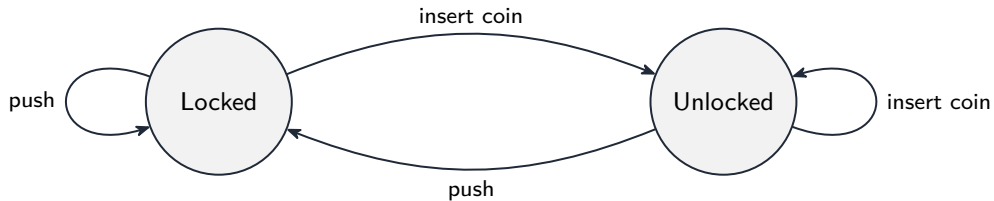
Method — Structure chart



Method — State-transition diagram

A **state-transition diagram** shows the **states** a system can be in (circles) and the **events** (labelled arrows) that move between them. Below is a turnstile:
It makes **missing transitions** easy to spot — *“what happens if I push while Locked?”*

Method — State-transition diagram



Practice 2.1 ■ 1 mark

State what a **structure chart** shows.

Solution 2.1

Method: Structure chart

Worked steps

The **hierarchical decomposition** of a program into **modules/subroutines** and the **parameters** passed between them [B1](#).

Practice 2.2 ■ 1 mark

In a structure chart, what do the **labels on the links** represent?

Solution 2.2

Method: Structure chart

Worked steps

The **parameters** passed **into** a module and the **results returned** from it [B1](#).

Practice 2.3 ■ 1 mark

State what a **state-transition diagram** shows.

Solution 2.3

Method: State-transition diagram

Worked steps

The **states** a system can be in and the **events** (transitions) that move it between them **B1**.

Practice 2.4 ■ 1 mark

In the turnstile diagram, which **event** moves it from Locked to Unlocked?

Solution 2.4

Method: State-transition diagram

Worked steps

insert coin (the coin event) **B1**.

Exam-style 3.1 ■ 4 marks

Draw a **structure chart** for a program MakeTea that calls BoilWater and AddMilk. AddMilk is passed the number of Cups. Show the parameter on the link.

Solution 3.1

Method: Structure chart

Worked steps

A correct chart: MakeTea at the top with lines down to BoilWater and AddMilk
M1 **A1** *hierarchy, both child modules, with Cups shown as a parameter in on the AddMilk link*
M1 **A1** *parameter shown, correct direction/label.*

Exam-style 3.2 ■ 4 marks

A vending machine has states Idle, Paid and Dispensing. Draw a **state-transition diagram** with these events: coin (Idle→Paid), select (Paid→Dispensing), done (Dispensing→Idle).

Solution 3.2

Method: State-transition diagram

Worked steps

Three states Idle, Paid, Dispensing **M1**, with arrows: Idle $\rightarrow(\text{coin})\rightarrow$ Paid **A1**, Paid $\rightarrow(\text{select})\rightarrow$ Dispensing **A1**, Dispensing $\rightarrow(\text{done})\rightarrow$ Idle **A1** — each transition labelled with its event.

Exam-style 3.3 ■ 1 mark

State **one** benefit of drawing a state-transition diagram.

Solution 3.3

Method: State-transition diagram

Worked steps

Any one: it makes **missing or invalid transitions** easy to spot; it documents the system's behaviour clearly for all events **B1**.

Exam-style 3.4 ■ 1 mark

In a structure chart, state what a **curved arrow** drawn over the links to two modules means.

Solution 3.4

Method: Structure chart

Worked steps

The two modules are called repeatedly in a **loop** — it shows **iteration** B1.

Exam-style 3.5 ■ 4 marks

A vending machine program is documented with a **state-transition diagram** and a **structure chart**.

(a) The machine has states Idle, Collecting, Dispensing and OutOfStock. **Complete** the state-transition table for the events shown.

current state	event	next state
Idle	coin inserted	_____
Collecting	enough money and item available	_____
Collecting	enough money, item sold out	_____
Dispensing	item delivered	_____

Exam-style 3.5 ■ 4 marks

- (b) **State** what a state-transition diagram shows that a structure chart does **not**.
- (c) **Explain** the meaning of the **curved arrow** symbol in a structure chart.
- (d) In the chart, ProcessSale calls CheckStock, which returns a Boolean. **Describe** how the parameter and its return value are drawn.
- (e) The team also produces a **program flowchart** for one module. **Explain** why all three documents are produced, rather than choosing whichever one the team prefers.

Solution 3.5

Method: Structure chart

Worked steps

(a) Idle + coin inserted $\rightarrow$ Collecting **B1**; Collecting + enough money and item available $\rightarrow$ Dispensing **B1**; Collecting + sold out $\rightarrow$ OutOfStock **B1**; Dispensing + item delivered $\rightarrow$ Idle **B1**.

(b) It shows the **states** the system can be in and which **events** move it between them **B1** — the behaviour over time, including what is not allowed, which a structure chart cannot express because it only shows which module calls which **B1**.

(c) A curved arrow means **iteration**: the modules it encloses are called **repeatedly** **B1**, for as long as the condition on the loop holds **B1**. (A diamond, by contrast,

Solution 3.5

marks selection.)

(d) Parameters are drawn as small arrows beside the line joining the two modules **B1**.

The parameter passed **down** to CheckStock — the item code — points from ProcessSale towards it **B1**; the Boolean returned points **back up** towards ProcessSale, and each arrow is labelled with the data it carries **B1**.

(e) Each shows a different view: the structure chart shows how the program is **decomposed** into modules and what data passes between them **B1**; the state-transition diagram shows the **behaviour** of the system as a whole **B1**; the flowchart shows the **detailed logic inside one module** **B1**. No single one of the three carries all of that.