



F4.4 Decomposition and abstraction

Functions and structured code · GCSE · OCR J277 2.1.1, AQA 8525 3.1.1,
Edexcel 1CP2 1.1.1 · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

Structure diagrams, and hiding the details of a job behind a function.

- Decomposition: breaking a problem down
- Abstraction: hiding the details
- Building from the bottom up, testing as you go

Questions

5 marks in all

- What is decomposition? [1 mark]
 - ☐ A Breaking a problem down into smaller, more manageable sub-problems
 - ☐ B Removing unnecessary detail
 - ☐ C Putting code in alphabetical order
 - ☐ D Deleting unused code
- What is abstraction? [1 mark]
 - ☐ A Removing unnecessary detail to focus on what matters
 - ☐ B Breaking a problem into parts
 - ☐ C Writing code in capitals
 - ☐ D Testing a program
- A simulator's mat keeps the walls and distances but not the carpet colour. Which idea is that? [1 mark]
 - ☐ A Abstraction
 - ☐ B Decomposition
 - ☐ C Iteration
 - ☐ D Casting
- What does a structure diagram show? [1 mark]
 - ☐ A A problem broken down into smaller sub-problems, top to bottom
 - ☐ B The order lines run in
 - ☐ C The values of variables as a program runs
 - ☐ D How data moves over a network
- Why keep a delivery route as a list of stops, separate from the code that drives? [1 mark]
 - ☐ A The route can change without changing the functions
 - ☐ B Lists run faster than functions
 - ☐ C Python requires it
 - ☐ D Functions cannot use numbers

The task: three stops

Write `go_to(x, y)` and `signal()`, then use them to visit three stops in order, signalling at each: **A** at (30, 10), **B** at (30, 60) and **C** at (0, 60), all in cm from where the robot starts. The robot starts in the bottom left of the mat.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def go_to(x, y):
    """Drive to (x, y), in cm from where the robot started."""
    here_x, here_y = position()

def signal():
    """Light and beep at a stop."""
    led("green")
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/f4-4-decomposition-and-abstraction/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add a fourth stop and a trip back to the start, by changing only the list of stops.
2. Draw a structure diagram for a robot that tidies balls into a box. Which boxes could reuse `go_to`?
3. Make `go_to` return how far it drove in total, and print the length of the whole route.