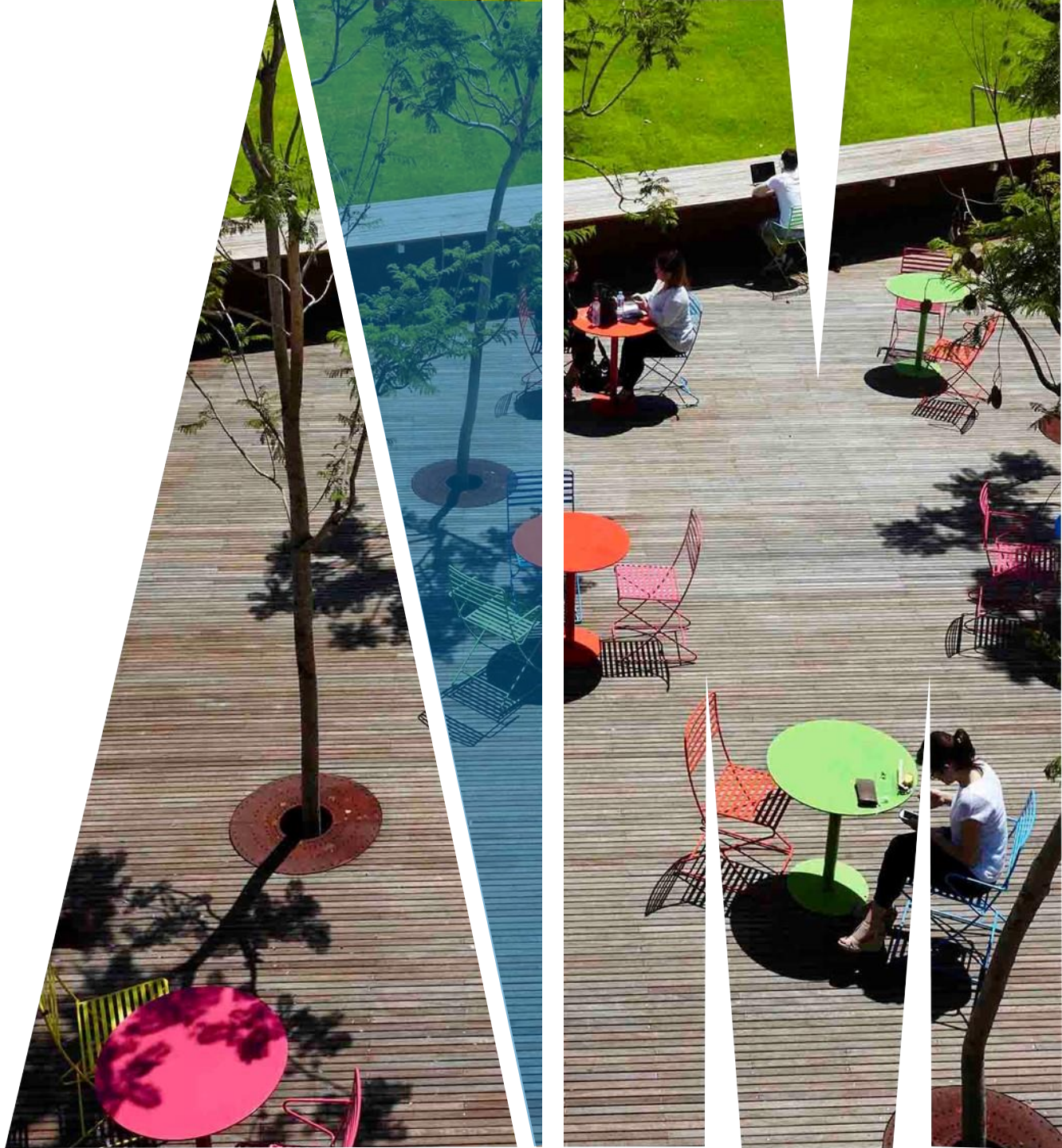
MONASH University

FIT2099 Object-Oriented Design and Implementation

Foundations of abstraction and separation of concerns



MONASH University

Outline

The cognitive load problem in programming

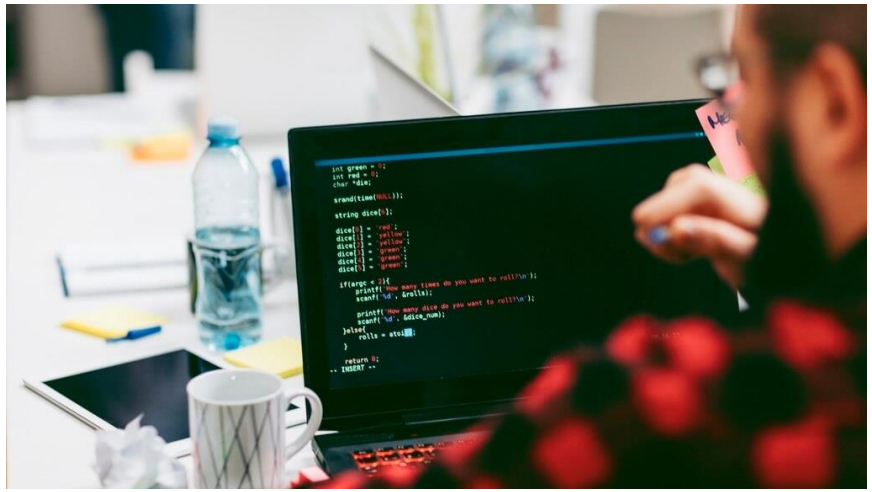
What is abstraction?

What is separation of concerns?

WHAT MAKES PROGRAMMING HARD?

Too many things to keep track of simultaneously

- Lines of code
- Variables
- Modules or source files
- Packages (or namespaces, or assemblies)



Complicated interactions between parts of the program

- dependencies between modules
- complex algorithms

All particularly significant when we need to *change* the program

Source: infoworld.com

MAKING PROGRAMMING EASIER FOR DEVELOPERS

We want to design our software in such a way as to make it **easier to create, maintain, extend, and modify**

End users use programs….

but classes and packages are used by **developers**

… if we make their working lives easier, we will produce software more efficiently:

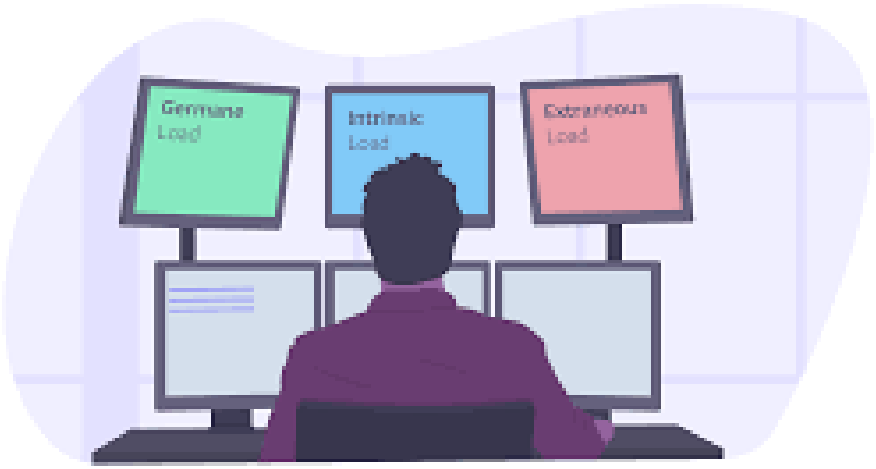
- make iterative development easier
- respond more readily to changes in requirements or in the development environment

MAKING PROGRAMMING EASIER FOR DEVELOPERS

Key factor to consider: **cognitive load**

- limit to human working memory
- high cognitive load leads to more mistakes, slower development

It is harder to debug than writing the code in the first place!



WHAT IS ABSTRACTION?

According to dictionary.com,

> *"the act of considering something as **a general quality or characteristic**, apart from concrete realities, specific objects, or actual instances."*

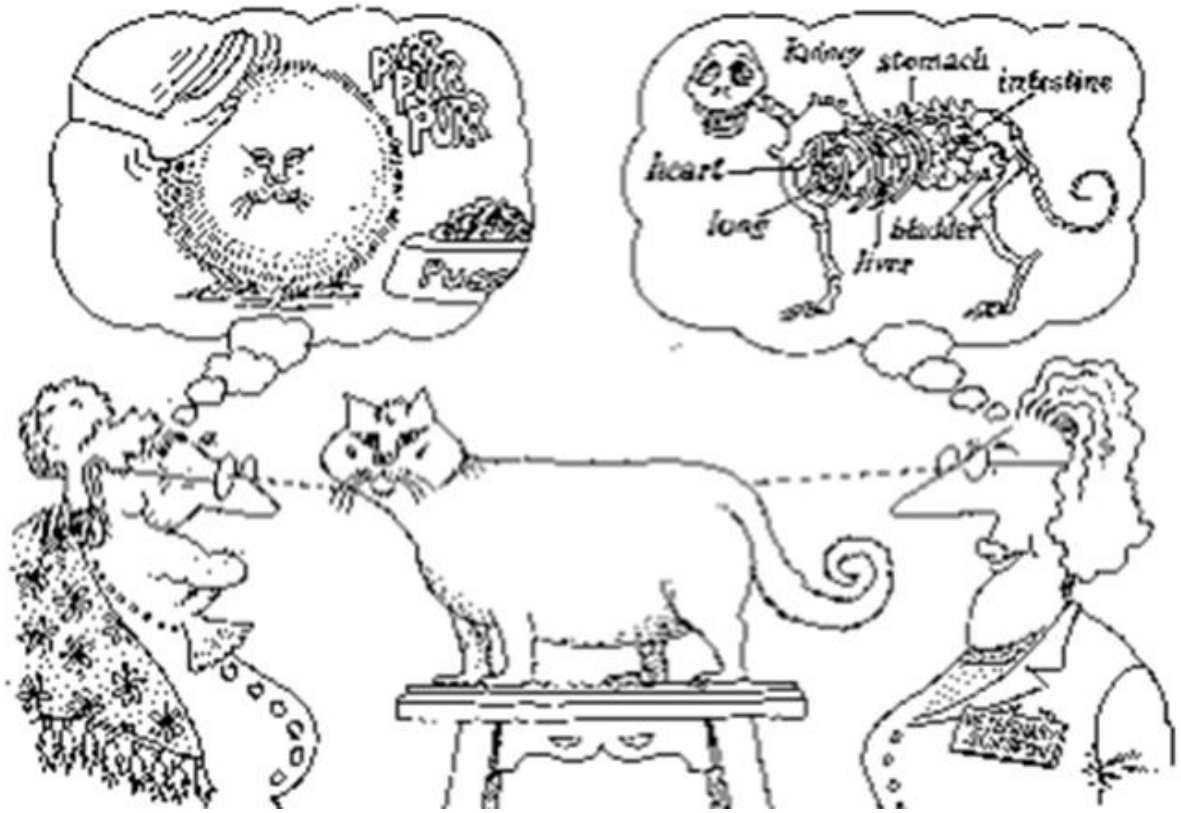
To a software developer, this means deciding

- **what information** do we need in order to represent some item or concept?
- **what should we expose to the rest of the code** (i.e. make public) so that we will be able to use this part easily?

WHAT INFORMATION DO WE NEED?

What **information** is relevant depends on how it will be used.

The veterinarian and the owner are both considering the same cat, but the ways they think about it (their mental models or **abstractions**) depend on their **relationship** to the cat: medical professional versus carer.



HOW DOES ABSTRACTION TRANSLATE INTO CODE?

At its simplest, we perform abstraction whenever we **bundle** related items together, **name** the bundle, and then **use it**

This includes:

- collecting lines of code together into a **method** or function
- collecting related data into a **class**
- grouping classes into **packages**

All of these things can be used **without much thought about their internals**…

- but only if they are well-designed!

OTHER RELATED CONCEPTS: ENCAPSULATION AND INFORMATION HIDING

It can be hard to distinguish between **encapsulation, information hiding** and **abstraction**.

Most of the language features that enable encapsulation also enable abstraction, so if you're trying to understand these concepts purely at source code level, you're likely to get confused.

OTHER RELATED CONCEPTS: ENCAPSULATION AND INFORMATION HIDING

Here's a guide:

- we use **encapsulation** when we bundle things together
- we use **abstraction** when we decide which things should be bundled together
  - we also use **abstraction** when we decide how things should look from outside (i.e. when we design a class's public interface)
- we use **information hiding** whenever we use an encapsulation mechanism that doesn't allow access from outside
  - private or protected modifiers: keep implementation details hidden
  - local variables: no access from outside the method
  - defensive copying: prevent external code from accessing internal data structures

NOTE: we will dive deep into these in later weeks).

EXAMPLE:
STRING IN C AND JAVA

The language C has a string datatype… sort of

```
char *string = "Hello, world!";
for (int i = 0; string[i] != '\0'; i++) {
  …
}
```

To use C strings, you need to know that they are represented internally as arrays of char – so their type is pointer to char. You also need to know that the end of the string is marked by a null, '\0' (literally a zero in memory).

EXAMPLE:
STRING IN C AND JAVA

This is the String type in Java

```
String string = "Hello, world!";
```

Do you know how Java strings are represented internally? Do you want to have to care?

It is much easier to use Java strings than C strings because Java presents the programmer with a better abstraction in which more of the implementation details are hidden.

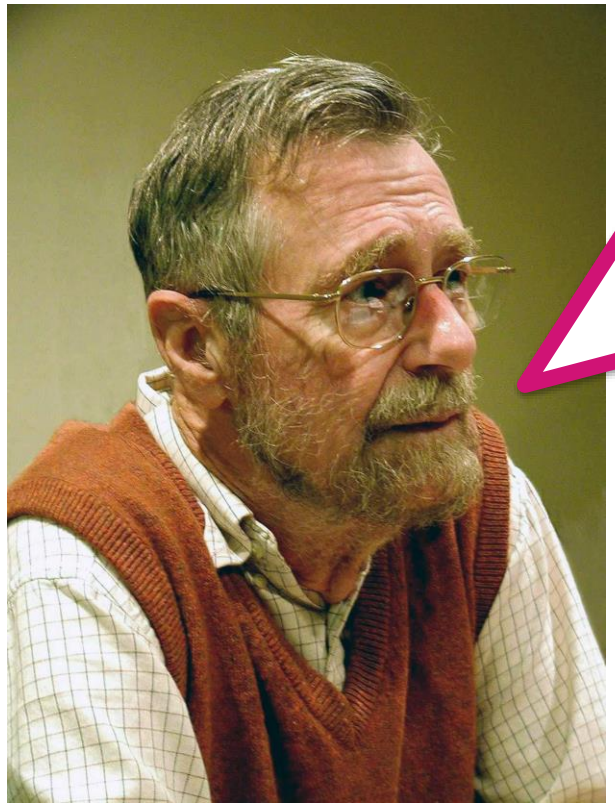
EXAMPLE:
STRING IN C AND JAVA

The language JAVA String class

public class String	
String()	*create an empty string*
int length()	*length of the string*
int charAt(int i)	*i th character*
int indexOf(String p)	*first occurrence of* p *(-1 if none)*
int indexOf(String p, int i)	*first occurrence of* p *after* i *(-1 if none)*
String concat(String t)	*this string with* t *appended*
String substring(int i, int j)	*substring of this string (*i *th to* j-1 *st chars)*
String[] split(String delim)	*strings between occurrences of* delim
int compareTo(String t)	*string comparison*
boolean equals(String t)	*is this string's value the same as* t*'s ?*
int hashCode()	*hash code*

Java String API (partial list of methods)

SEPARATION OF CONCERNS



It is what I sometimes have called "the separation of concerns", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by **"focusing one's attention upon some aspect"**: it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, **the other is irrelevant**.

It is being **one- and multiple-track minded simultaneously**.

-- E.W. Dijkstra, ”On the Role of Scientific Thought”

https://www.cs.utexas.edu/~EWD/transcriptions/EWD04xx/EWD447.html

MONASH University

By Hamilton Richards - manuscripts of Edsger W. Dijkstra, University Texas at Austin, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=4204157

SEPARATION OF CONCERNS

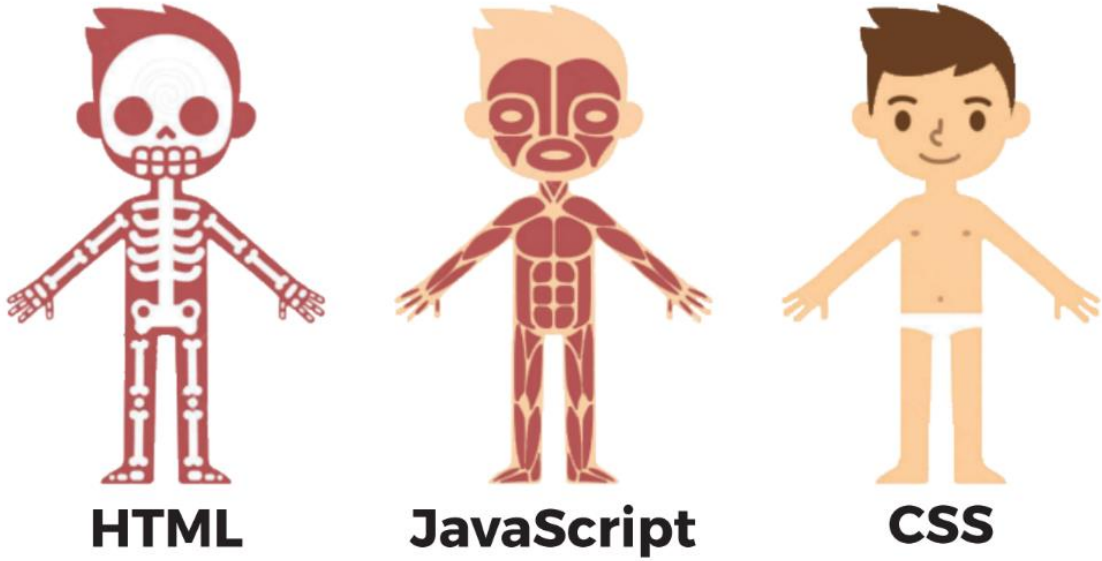
For our purposes, a **“concern” is a responsibility**

Every ‘module’ should have a **single, well-defined** set of responsibilities

Responsibilities should **overlap as little as possible** with other ‘modules’

- shared responsibilities often leads to repeated code

Unclear responsibilities make the ‘module’ **hard to use**.

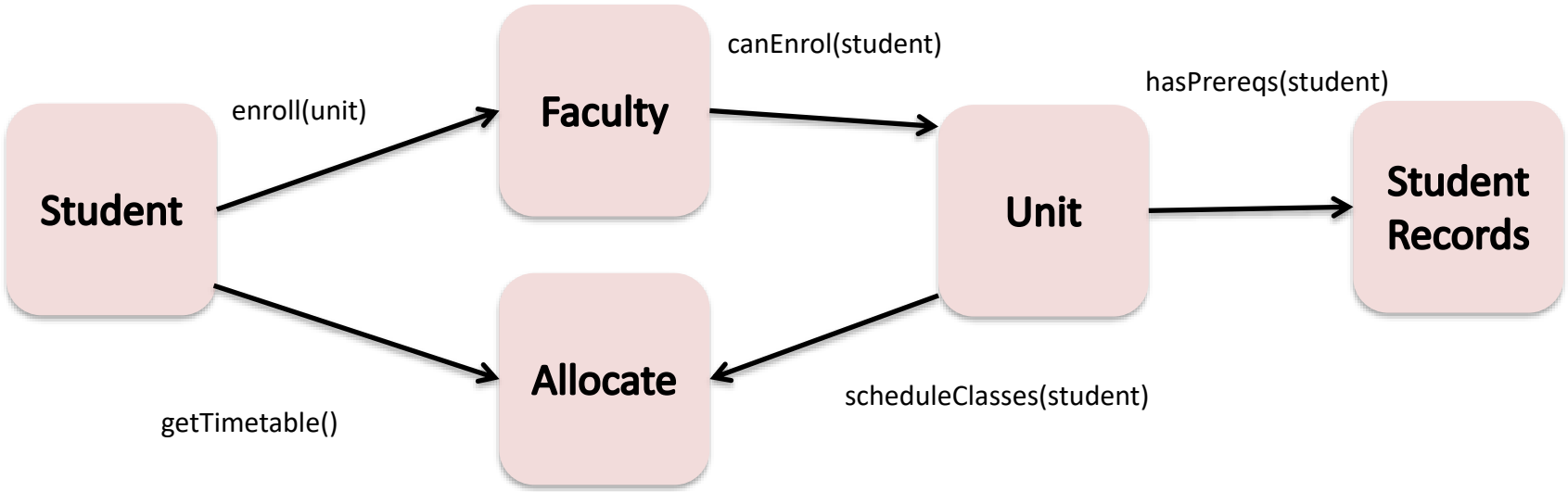


SEPARATION OF CONCERNS

Not about the code, it's about **the conceptual structure underlying the code**.

SEPARATION OF CONCERNS IN OBJECT ORIENTED DESIGN

Each class (or 'module'/ component) addresses a separate concern (or has a very specific responsibility).



NOTE: this concept is closely related to the Single-Responsibility Principle (SRP) that we will cover in later weeks!

Summary

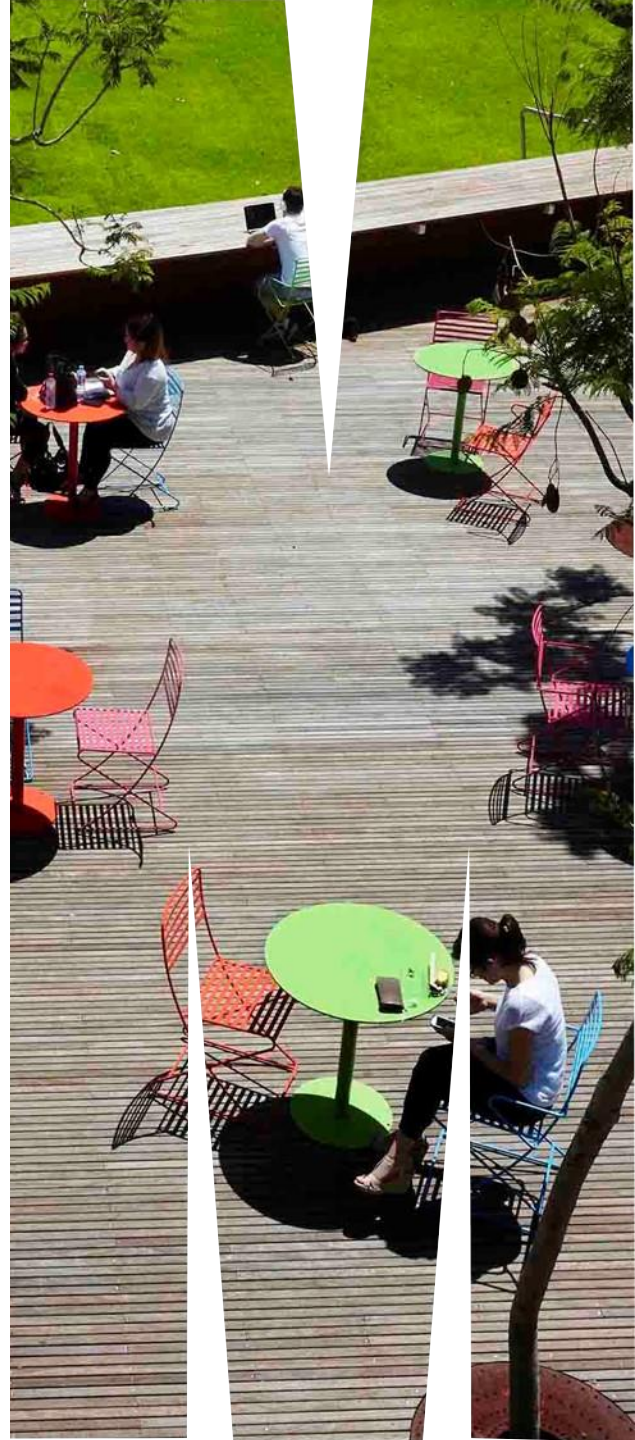
The cognitive load problem in programming

What is abstraction?

What is separation of concerns?

MONASH University

Thanks



MONASH University