
University of
Southampton

University of Southampton

COMP1202 - Introduction

Son Hoang
(adapted from Prof David Millard's slides)

COMP1202 (AY2023-24)

University of Southampton

Part 1

Organisation

University of Southampton

Welcome to Programming I

Son Hoang
(t.s.hoang@soton.ac.uk)

Heather Packer
(hp3@ecs.soton.ac.uk)

Jian Shi
(jian.shi@soton.ac.uk)

Mohammad Soorati
(jian.shi@soton.ac.uk)

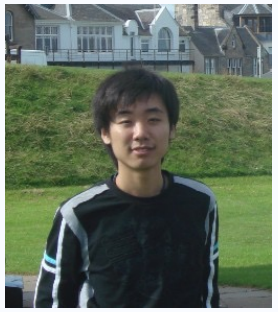
Purpose of this Module

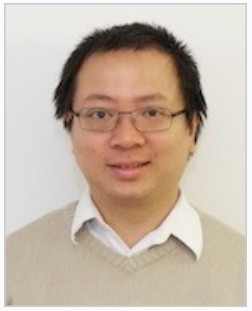
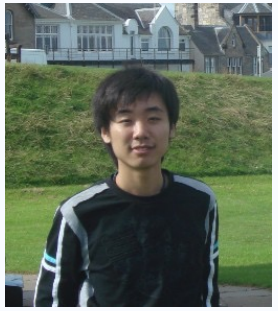
"This module aims to introduce students to the **principles of programming** using an object-oriented approach and to provide them with the programming skills necessary to continue the study of computer science. **Java is used as the introductory language**."

- Writing and running programs
- Compilation, Interpretation and the Java Virtual Machine
- Variables, Objects, Primitives and Scope
- Methods
- Computational Thinking
- Constructors
- Loops and Arrays
- Collections and Iterators
- The Java Library
- Integrated Development Environments
- Testing and Debugging
- Software Design (What makes a good program)
- Super and Sub Classes (Inheritance)
- Polymorphism and Dynamic Binding
- Abstract Classes and Interfaces
- Designing Applications (Moving from problem to solution)

Main Course Structure

- Two 1-hour lectures each week
  - Monday 13:00 (B32/1015)
  - Tuesday 16:00 (B32/1015)
- Also a 2-hour lab each week
  - (worth 15%, more to follow)
  - Tuesday 09:00-11:00, 14:00-16:00
  - Zepler Level 3 Lab (B59/3237)
- Other Assessments
  - Coursework (set in Week 4, due in Week 11, worth 45%)
  - Exam (after X-mas, worth 40%)
- More details on the next few slides


About Labs
Overview

- 2 labs every Tuesday
  - 09:00-11:00
  - 14:00-16:00
- Starting October 3rd (i.e., tomorrow)
- B59 (Zepler Building) Level 3 (R 3237)

- **Practice common courtesy**
- **There are no dumb questions**
- **Let someone know when you are stuck**
- **DO NOT** cut and paste code
- **DO NOT** share your code
- We treat **Academic Integrity** (AI) issues **seriously**

About Labs

Code Functionality Test Harness

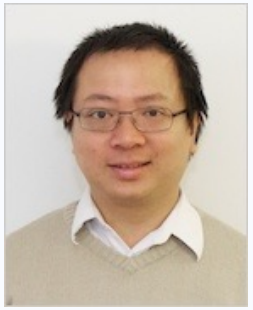
- Code functionality is assessed using a **test harness**
- You will **receive an email** detailing any
  - Errors and whether your code passed the tests
  - Styling problems (incorrect indentation, naming problems, etc.)
- **Make sure you follow the lab instructions carefully**, otherwise, your code might not meet the spec
  - That includes spelling in outputs!
- You must **structure** your code **exactly** the same as specified in the lab, otherwise, the test harness will not work
- **Submit your code well before the deadline**, otherwise, you might encounter issues making you miss the deadline

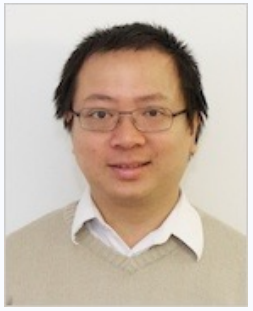
About Coursework and Exam

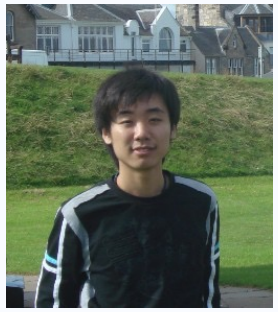
- Coursework
  - Set in Week 4
  - Due in Week 11 (before the X-mas break)
  - Feedback in the new year (before the exam)
- Exam
  - Multiple choice computer-based exam (QuestionMark OnDemand)
  - In-person (unless changed due to the pandemic situation)
  - A Mock Exam is available from the module website

Additional Streams

- Space Cadets
  - For people who are more **experienced programmers**
  - Run by Son and Frederick Nash (Freddie)
  - Optional weekly challenge and discussion
    - Friday 16:00-18:00 (B2/1039)
    - Weeks 1-7 (i.e., starts this week!)
- Ground Controllers
  - For people who are **new to programming**
  - Run by Son and Jian
  - Optional weekly workshop (Zepler Level 3 Lab (B59/3237))
  - Weeks 1-10 (i.e., starts this week!)

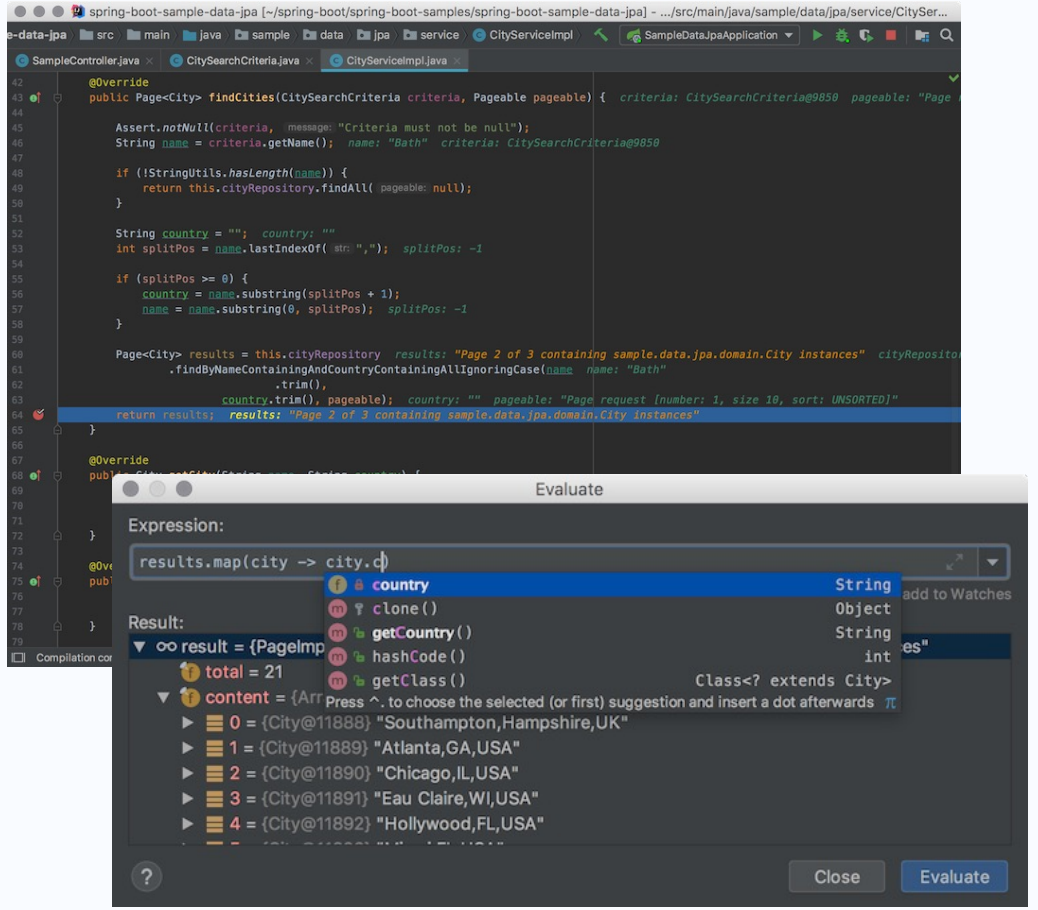






IntelliJ

- The main course text is *Objects First with BlueJ* (available from the library)
  - BlueJ is an environment designed for learning
- Although we will start with using Java from the command line, we will soon introduce you to the IntelliJ environment
- IntelliJ is a more professional environment (compare to BlueJ).



University of Southampton

A Dirty Secret!!!

- No matter how we teach you will mainly learn through practice!

- Programming is the single most important skill for a computer scientist or IT specialist
  - Systematic thinking and problem solving
  - Abstraction and data modelling

- Did we mention that you need to PRACTICE?

University of Southampton

PRACTICE!

"I've often thought that sucking less every year is how humble programmers improve. You should be unhappy with code you wrote a year ago."

Jeff Atwood, https://blog.codinghorror.com/sucking-less-every-year/

(Co-Founder of stackoverflow.com)

"It is true. I have no talent. What I do have is a lot of practice. And I am not talking about occasionally dabbling in Ruby on the weekends. I am talking about the kind of practice where I beat code that isn't working into submission (though often times the code wins)."

John Nunemaker,

http://www.railstips.org/blog/archives/2010/01/12/i-have-no-talent/

Your Lecturer: A Warning from History

Online Notes Wiki

https://secure.ecs.soton.ac.uk/module/2324/COMP1202/29509/

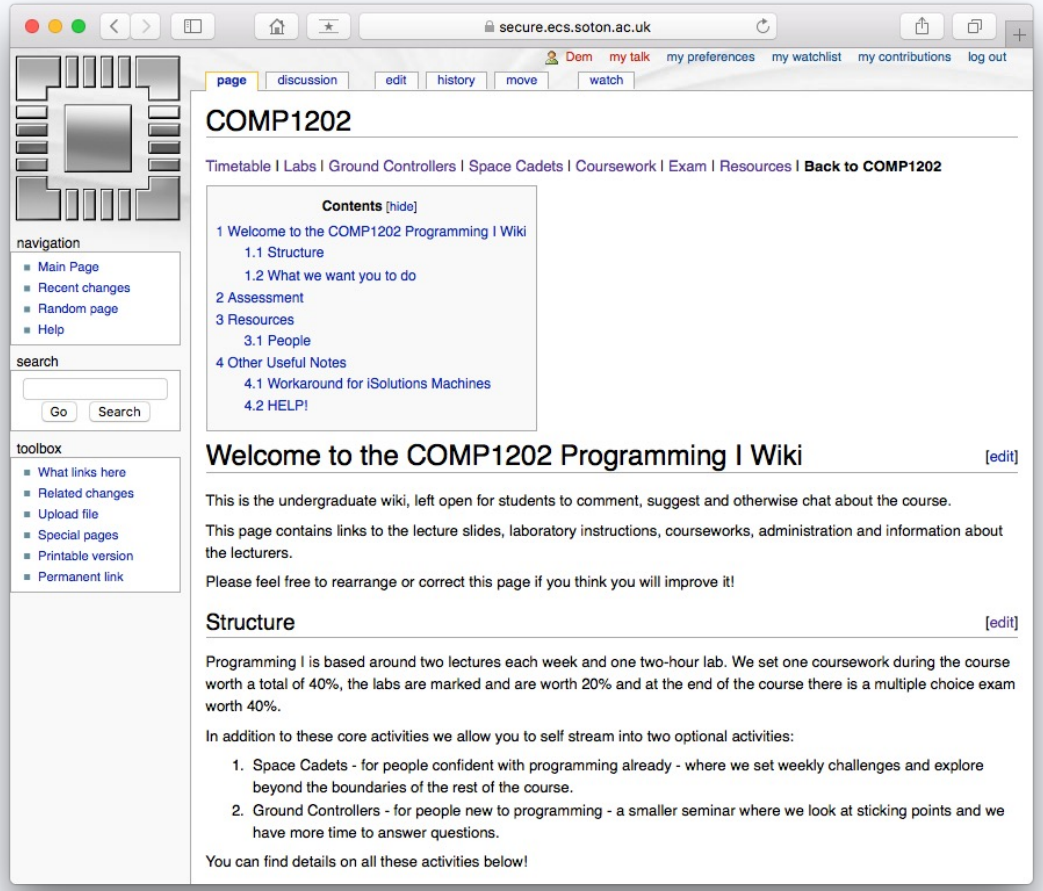
(Southampton campus)

leads to

https://secure.ecs.soton.ac.uk/student/wiki/w/COMP1202

Material is on different platforms

- EdShare (Slides and Summary Videos)
- Blackboard (Quizzes)
- MS Teams + Panopto (Recordings)



Timetables

- Your timetable shows all of the sessions/labs allocated to your modules

- Modules do not always use all of those sessions!
  - For example, we have optional labs (GC and SC)

- So check the notes pages of each course for more info and adjust your timetable accordingly!

Reminder COMP1202 Timetable

https://secure.ecs.soton.ac.uk/student/wiki/w/COMP1202/Timetable

Week	Start Date	Topic	Section of Objects First Textbook	Mon 13-14 lecture - B32/1015	Tue 16-17 lecture - B32/1015	Labs (Tue 9-11, 11-13) - B59/3237	Ground Controllers (Tue 14-16) - B59/3237	Space Cadets (Fri 16-18) - B2/1039
1	02/10/23	Introduction	1.1, 1.2	(0) Starting Out (sh) ▪ EdShare ▪ Please take the "00 - Starting Out" quiz on Blackboard	(1) Introduction to Java (sh) ▪ EdShare,	Lab 0: Lab Preparation (hp)	GC1. Odd Time (sh)	SC0. Introduction (sh)
2	09/10/23	Programming in Java	1.5, 2.15, 2.5	(2) Variables, Objects, Primitives and Scope (js) ▪ EdShare ▪ slides	(3) Methods (js) ▪ EdShare ▪ slides ▪ code	Lab 1. Hello World and Conditionals (hp)	GC2. ECS Course (sh)	SC1. Challenge 1 Review (sh)
3	16/10/23	Building Better Programs/Objects	2.4, 3.10, 3.11	(4) Constructors (js) ▪ EdShare ▪ slides with demo code	(5) Computational Thinking (ms) ▪ EdShare	Lab 2. Classes, Conditionals and Control Statements (hp)	GC3. Text Book (sh)	SC2. Challenge 2 Review (sh)
4	23/10/23	Loops and Arrays	4,4, 4.9, 4.10, 4.12	(6) Loops and Arrays (sh) ▪ EdShare ▪ Blackboard quizzes	(7) Collections and Iterators (sh) ▪ EdShare	Lab 3. Methods, Object Interaction and Testing (hp)	GC4. Loops and Arrays (sh)	SC3. Challenge 3 Review (sh)
5	30/10/23	Environments and the Java Lib	6.3, 6.6	(8) The Java Library (sh) ▪ EdShare	(9) Using IDE (sh) & Academic Integrity (AI Officer) ▪ EdShare	Lab 4. Loops and ArrayLists (hp)	GC5. Phone Book (js)	SC4. Challenge 4 Review (sh)
6	06/11/23	Inheritance	10.2, 10.3, 10.4, 10.6, 10.7	(10) Super and Sub-classes (js) ▪ EdShare ▪ Slides ▪ Handout	(11) Polymorphism (js) ▪ EdShare ▪ Slides	Lab 5. Arrays, HashMaps and APIs (hp)	GC6. Module Management (js)	SC5. Challenge 5 Review (sh)
7	13/11/23	Building Better Classes	12.3, 12.6, 3.13	(12) Abstract Classes and Interface ▪ EdShare (js) ▪ Slides ▪ Handout.pdf	(13) Testing and Debugging (sh) ▪ EdShare	Lab 6. Inheritance, Method Overriding, Polymorphism and Exceptions (hp)	GC7. Tax Calculator (sh)	SC6. Challenge 6 Review (sh)
8	20/11/23	Object-Oriented Design	8.3, 8.4, 8.6	(14) Software Design (ms) ▪ EdShare	(15) Designing Applications (ms) ▪ EdShare	Lab 7. Scope, Test Harnesses and Debugging (hp)	GC8. Assessment (js)	n/a
9	27/11/23	Coding in Practice	n/a	TBA	TBA	Lab 8. Method Overloading, Interfaces and Class Hierarchy (hp)	GC9. File Reader (sh)	n/a
10	04/11/23	Revision and Other Languages	n/a	Languages Balloon Debate (sh) ▪ EdShare	Revision Lecture (sh) ▪ EdShare	Lab 9. Handling Files (hp)		n/a
11	11/12/23	Back-up	n/a	n/a	n/a	Coursework surgery	n/a	n/a
12	16/01/23	Back-up	n/a	n/a	n/a	n/a	n/a	n/a

University of Southampton

How to Get Help

- **Refrain** from emailing (get lost in the avalanche)
- **Do not** drop by the office (often busy)
- **Do** come down to ask questions at the end of the lecture
- **Do** ask questions in the sessions
- **Do** ask questions in the labs
- **Do** ask each other!
  - Be careful of **academic integrity**.
- **Do** ask questions on Discord (COMP1202 area)
- **Do** ask questions Programming Helpdesk (Weekdays: 10:00—12:00, 14:00—16:00)
  - on Discord (https://discord.ecs.soton.ac.uk)
  - via email (helpdesk@ecs.soton.ac.uk)

University of Southampton

Part 2

Starting Out

University of Southampton

In this Part

- What is Programming?
- Programming Paradigms

- A Taste of Things to Come
  - Classes and Objects
  - Variables
  - Logic
  - Data Structures
  - Skills

What is Programming?

- Wikipedia (2023)
  - "is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building an executable computer program … The purpose of programming is to find a sequence of instructions that will automate the performance of a task (…) on a computer, often for solving a given problem."
- thefreedictionary.com / dictionary.com (2023)
  - "the act or process of planning or writing a program"
- Oxford Learner's Dictionary (2023)
  - "the process of writing and testing programs for computers."

What is Programming?

- Wikipedia (2023)
  - "is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building an executable computer program … The purpose of programming is to find a sequence of instructions that will automate the performance of a task (…) on a computer, often for solving a given problem."
- thefreedictionary.com / dictionary.com (2023)
  - "the act or process of planning or writing a program"
- Oxford Learner's Dictionary (2023)
  - "the process of writing and testing programs for computers."

What is Programming?

- Wikipedia (2023)
  - "is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building an executable computer program … The purpose of programming is to find a sequence of instructions that will automate the performance of a task (…) on a computer, often for solving a given problem."
- thefreedictionary.com / dictionary.com (2023)
  - "the act or process of planning or writing a program"
- Oxford Learner's Dictionary (2023)
  - "the process of writing and testing programs for computers."

What is Programming?

- Wikipedia (2023)
  - "is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building an executable computer program … The purpose of programming is to find a sequence of instructions that will automate the performance of a task (…) on a computer, often for solving a given problem."
- thefreedictionary.com / dictionary.com (2023)
  - "the act or process of planning or writing a program"
- Oxford Learner's Dictionary (2023)
  - "the process of writing and testing programs for computers."

University of Southampton

Programming Flavours

- Procedural (e.g., **C** or Pascal)
  - Fixed list of instructions
  - Do this, Do that, if this then do that

- Declarative
  - More like declaring rules (or grammar)
  - Behaviour emerges from the rules being applied
  - Examples
    - Functional Programming (e.g. Scheme or **Haskell**)
    - Logic Programming (e.g. Prolog)

University of Southampton

Object-Oriented Programming

The main idea

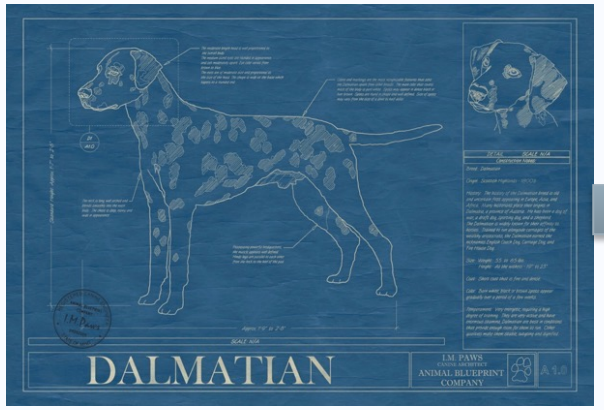
- Everything is a Thing
- A Program is made up of Things interacting
- Things have both properties and behaviours
- E.g. Dogs
  - What are the properties of a dog?
  - What can a dog do?

Classes and Objects

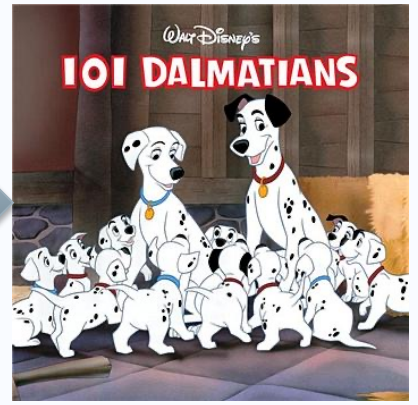
All the properties and behaviours of a Dog can be packaged in a **class**

- A class is like a Blue Print
- We can build many unique dogs from the same Blue Print
- These are called **objects**
- They all have similar properties and behaviours as defined by the class

1 Class



101 Objects



http://www.animalblueprintcompany.com/

Building Blocks

- Objects and Classes are specific to Object-Orientated Programming

- But there are more common, more fundamental Programming Principles that we will be covering in the course

University of Southampton

Variables

- Like algebra
  - x = 4
- Not so much like algebra
  - name = “Rover”
- Object properties are stored in variables

Methods

- Blocks of code that define a sequence of actions
- Object behaviours are defined by methods
- Often this will use the object's properties

```
printDogsDetails()
{
   print name;
   print age;
}
```

Logic

https://blackboard.soton.ac.uk/webapps/blackboard/content/launchAssessment.jsp?course_id=_221896_1&content_id=_6319336_1&mode=view

True or False – The basis of all decisions

Meaning	Formula	x = 3	x = 4	x = 11
x is equal to 4	x == 4	True or False?	True or False?	True or False?
x is greater than 4	x > 4	True or False?	True or False?	True or False?
x is greater than or equal to 4	x <= 4	True or False?	True or False?	True or False?
x is not equal to 4	x != 4	True or False?	True or False?	True or False?
x is greater than 2 and x is smaller than 8	x > 2 && x < 8	True or False?	True or False?	True or False?
x is greater than 8 or x is smaller than 2	x > 8 || x < 2	True or False?	True or False?	True or False?

Control Flow

- If statements

```
if size<10
    dog says “Yip Yip”
else
    dog says “Ruff Ruff”
```

- Loops

```
while number<10
    dog says “Bark!”
    number = number + 1
```

Side Note:

Human readable versions of programs are called **Pseudocode**.

They look like real programs but are not as precisely defined.

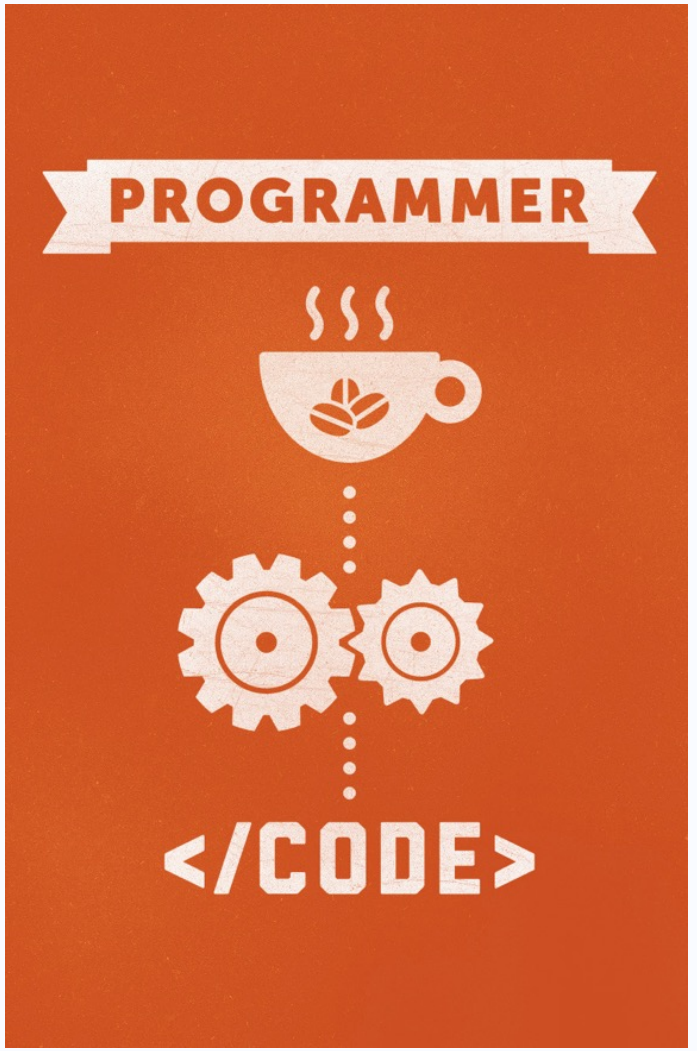
They are good for communicating ideas and showing structure.

Data Structures

- More complex data …
- Arrays:
  - Like Lists, Tables, Matrices
- HashMaps
  - Associate a key with a value
  - Like a dictionary

Skills

- Computational Thinking
- Program Design
- Choosing Tools (IDEs)
- Testing/Debugging



University of Southampton

Summary

- What is Programming?
- Programming Paradigms
- A Taste of Things to Come
  - Classes and Objects
  - Variables
  - Logic
  - Data Structures
  - Skills

University of Southampton

YOUR QUESTIONS

Quizzes