

COMP1005
Programming and Algorithms
Programming Concepts

Jamie Twycross

Overview

- Approaches to programming
- Algorithms and procedures

Approaches to Programming

Introduction

- Computers need to be told what to do
- Programming is (the **science** and **art** of) making a computer do what you want it to do
- A program describes what you want a computer to do
- Fundamentally, different ways (**programming paradigms**) of telling a computer what to do

Programming Paradigms

- **Declarative paradigm** - describe the properties of what you want
- **Imperative paradigm** - describe how to construct what you want
- Example - drawing a square:

Declarative

```
box width 5cm height 5cm
```

Imperative

```
moveto 0 0; lineto 100 0; lineto 100 100; ...
```

- Computers are naturally imperative
- But can use declarative descriptions

High and Low Level Languages

- Computers understand a basic set of instructions - **instruction set**
- Closely linked to **machine architecture** e.g. x86, PowerPC, Z80, ARM, ...
- In a **low level language** you essentially program using the instruction set:
 - machine code
 - assembly language
- **Assemble** a low level program
- But more often write programs in a **high-level language**:
 - C, Python, Java, JavaScript, HTML, Perl, R, ...
- **Compile/interpret** a high level program
- General differences between high and low level programs:
 - ease of use
 - portability
 - efficiency
 - memory footprint

What is a Program?

- A program is a **sequence of instructions**
- Computer executes them one by one
- Each instruction usually does one simple thing
- Instructions act on some **data**
- Together they build up to do complex tasks
- Types of instructions:
 - Mathematical operations.
 - Variable operation - load and store data
 - Conditionals - do this only on some condition.
 - Loops - repeat doing this.
 - Procedures - blocks of instructions for reuse.
- A true but potentially unhelpful answer!
- How to solve complex problems?

Functional Decomposition

- How do we solve a complex problem?
- General methodology:
 - break the problem down into smaller **simpler steps**
 - perform each step to solve the bigger problem
- This is called **functional decomposition**
- This is the way we think about and write **programs**
- Programming is **creative** – many interesting ways of solving a problem
- Often useful to think in terms of transforming input data to output data

Algorithms and Procedures

Algorithms

- An **algorithm** defines how to solve a problem computationally
- A clearly and unambiguously defined set of steps
- Different algorithms for doing the same task
- An algorithm is independent of language - pseudocode
- An algorithm is *not* a program
- A program often **implements** a number of algorithms

What is a Program?

- A program is a set of **algorithms** and **data structures**

Procedures

- A procedure is a block of instructions in a programming language
- Procedures are also called **subroutines** or **functions**
- An algorithm may be implemented as a single procedure
- More often, implemented using multiple procedures

A Recipe Analogy

- Victoria Sponge:
 1. Preheat the oven to 180C/350F/Gas 4
 2. For the cake, grease and line two 20cm/8in sandwich tins with baking parchment, and dust with flour
 3. In a food mixer, cream together the butter and sugar until pale and fluffy
 4. Gradually beat in the eggs, then add the flour and orange zest and mix until well combined
 5. Divide the mixture evenly between the two cake tins, then bake in the oven for 25-30 minutes, or until golden-brown and slightly springy to the touch

A Recipe's Procedures

- A recipe is an “algorithm” for making a cake:
 - Defined in terms of a set of steps or processes
 - Several steps may be grouped together into “procedures” for doing specific smaller tasks
- Can see how to make the cake clearly
- Smaller tasks common across several recipes
- Procedures are reused across several recipes

Procedural Programming Paradigm

- C is an **imperative programming language**
- C is a **procedural programming language**
- All instructions are grouped into procedures
- A procedure is a block of instructions that is given a name
- Invoke (call) the procedure and these instructions by name

Procedural Decomposition

- Procedural Decomposition: break a large program in to smaller procedures
- Can then implement more complicated tasks in terms of the smaller ones
- Small tasks are easier to:
 - implement
 - debug
 - maintain
 - reuse

Summary

- Covered a number of fundamental programming concepts:
 - declarative/imperative programming paradigm
 - high/low level languages
 - functional/procedural decomposition
 - programs/algorithms/procedures
 - procedural programming paradigm

Activities

- Do some reading around on the internet on the fundamental programming concepts covered in this lecture
- Wikipedia is a good start e.g.:
 - `https://en.wikipedia.org/wiki/Declarative_programming`
 - `https://en.wikipedia.org/wiki/Imperative_programming`
 - `https://en.wikipedia.org/wiki/Algorithm`
 - ...