1 Introduction

**Objectives**
- Develop problem-solving skills
- Useful to all engineers
- Develop basic programming skills
- Useful in some CAE applications
- Necessary for computer and other engineers.
- Introduce main themes of computer science
- Of general interest to all ???

2 Introduction

**Computers are Pervasive**
- When did you last interface with a computer?
- Or with a device that interfaces with a computer?
- Or with a human service provider who depended on a computerised device to deliver that service?

3 Introduction

**Problem-Solving**
- Compute the shortest route that visits a list of cities?
- Find the web pages most likely to satisfy a query?
- Produce monthly paychecks for TCD employees.
- Identify a car from an image of its number plate.
Programming and Problem Solving

- **Algorithm**
  - A sequence of precise instructions which leads to a solution

- **Program**
  - An algorithm expressed in a language the computer can understand

---

**An Algorithm**

Determine how many times a name occurs in a list of names.

1. Input the list of names.
2. Input the name being checked
3. Set a counter to zero.
4. For each name on the list
   - If the list name is the same as the name being checked
     - Add 1 to the counter.
5. Output the counter value as the answer.

---

**Pseudocode Algorithm**

Determine how many times a name occurs in a list of names.

1. Input the name-list.
2. Input the test-name.
3. Set a counter to be 0.
4. FOR EACH name on name-list
   - IF name is the same as test-name
     - THEN Add 1 to the counter
5. Output counter.

---

**A program fragment in C++**

Computes the sum of even numbers from 1 to an entered number

```cpp
firstexample.cpp
```
Other things you might learn

- Computers are “stupid”.
- Programming is easy, a bit tedious.
- Problem solving is challenging and fun.
- Some things can’t be computed.

Practicalities

- Lectures
  - Mon 1-2 Goldsmith Hall
  - Wed 1-2 Goldsmith Hall
  - Fri 12-1 Hamilton 3 (McNeil)
- Labs
  - Start this week, check timetable
  - 2 hours per week
  - Attendance compulsory
- Coursework
  - Completed in labs
  - Contributes 20% to overall mark.
  - Marks posted weekly

Textbook & Webpage

  - We’ll cover Chapters 1-7, 9 and possibly 16.
  - Will also be used in 2E3
  - Strongly recommend that you buy it.
  - TCD library Call No. 500.16424 P69°3
  - Get the online version (in full or by chapter) at http://www.cengagebrain.co.uk/shop/en/GB/storefront/cmea?cmd=CLIHeaderSearch&fieldValue=9781133526346

Learning in 1E3

- Lectures
  - Primary source of important concepts, learning activities, peer learning, …
- Laboratory sessions
  - Your chance to apply the concepts and practice programming
- Textbook
  - Supplementary concepts
  - Assumed that you read along and grasp stuff that is not covered in lectures/labs.
Programming Language

- We’ll use C++
- C++ is an object-oriented (OO) language
- We’ll focus on the procedural aspects, not objects and classes.
- C++ is the OO extension of C
- C is a high-level language but allows low-level control of machines.
- C is the language used by UNIX programmers.

In-class Polling

- Download Turning Technologies’ ResponseWare app from the AppStore or Google Play or
- go to (don’t search for) www.responseware.eu
- Enter the Session ID “CS1E03”. There is no need to enter any other details.

I have prior experience of

A. C++ or Java or other programming
B. Writing html, php, etc. but not real programming
C. Limited editing of computer code files
D. Never even looked inside a code file.

Sample problems

- For the rest of today we’ll look at some simple programming problems and how we would approach them.
**Cm to yds, ft, inches**

- Design an algorithm to convert a number of centimetres to the equivalent number of yards, feet and inches.
- 1 yard = 3 feet; 1 foot = 12 inches; 1 inch = 2.54 cm
- You can assume that there is a way to compute integer division and remainder

**Better-value pizza**

- Design an algorithm to determine which pizza supplier is better value.
- Input is price and diameter of two pizzas.
- Output is “first” if the first pizza’s price per square inch is less than the second’s.

**Which is probably NOT part of the better value pizza program?**

ppsi = price per square inch

A. \( \text{radius1} = \text{diam1} / 2 \)
B. \( \text{area2} = \pi * \text{radius2} * \text{radius2} \)
C. \( \text{ppsi2} = \text{area2} / \text{price2} \)
D. if (ppsi1 < ppsi2) print “first is better value”

**Next ...**

- Before the next lecture
  - Skim Chapter 1 of the textbook
  - Look through “1a Notes for Chapter 1”
- Next day we’ll
  - Look at a full C++ program and identify all the parts
  - See how programs are entered, compiled and run in our environment.