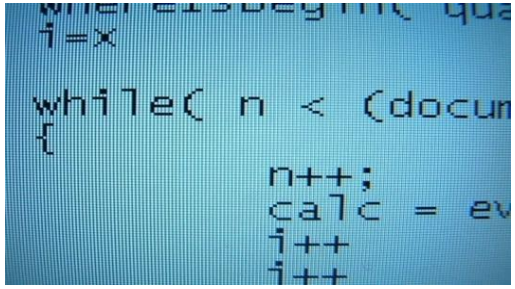
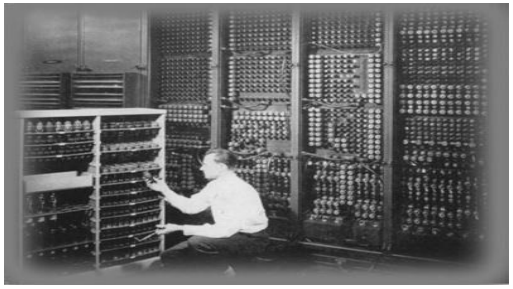




CS1PR16

Introduction to the module

Staff Involved in Teaching

- This module will be team-taught by:
  - Julian Kunkel
  - Chris Maynard
  - A teaching fellow will join us for Spring
- Our contacts are provided in BlackBoard
- Supported by:
  - Student assistants

Outline

- A: General Information (not assessed)
  - Module Aims
  - Module Outline
  - Organisation of the Module
  - Prescribed Learning Journey
  - Questions
- B: Algorithmic Thinking (Part 1 of 2)

Computer Science

***Study of computers**, their **design** (computer architecture), and their **uses** for **computation, data processing, and systems control**, including **design** and **development** of computer hardware and **software**, and **programming**.*

*The field encompasses **theory**, mathematical activities such as **design and analysis of algorithms**, **performance studies** of systems and their components, and estimation of reliability and availability of systems by probabilistic techniques*.

[Britannica Concise Encyclopedia]

Computational science applies CS to scientific problems like Biology or Physics

***Information technology (IT)** is the use of computers to*

*store, retrieve, transmit and manipulate **data or information** effectively and efficiently.*

[Wikipedia]

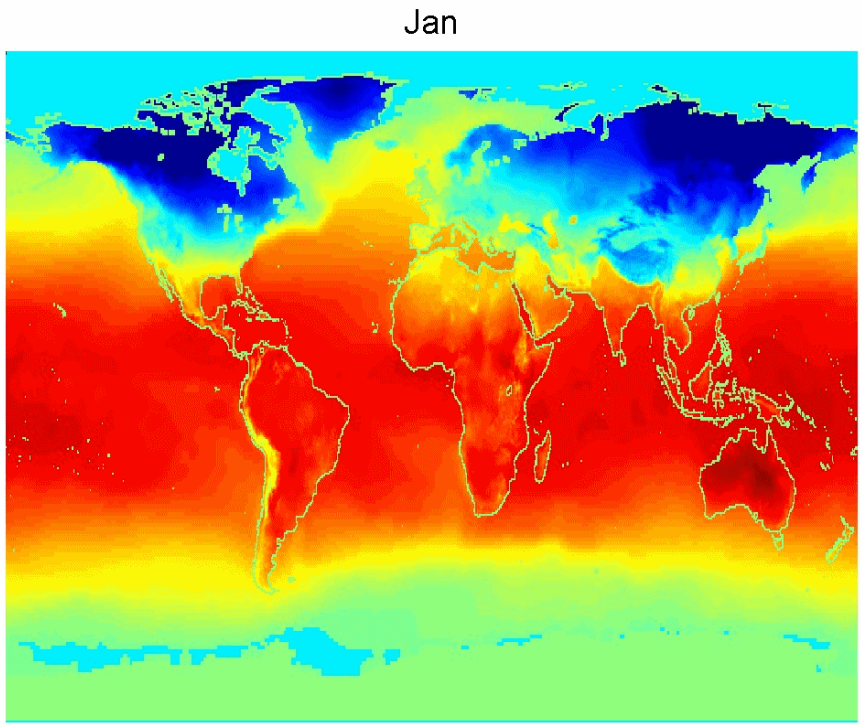
Computer science is no more about computers than astronomy is about telescopes.

[Michael Fellows]

Example: Weather Forecasting

- Computation of future weather!
  - By simulating of physical processes
  - e.g., radition by the sun
- Significant compute requirements
  - 1000s of servers must work together
- Input: Data collected over the past days
  - temperature, rainfall, pressure
- Output: Prognostics of rain/temperature

- Highly interdisciplinary and challenging!



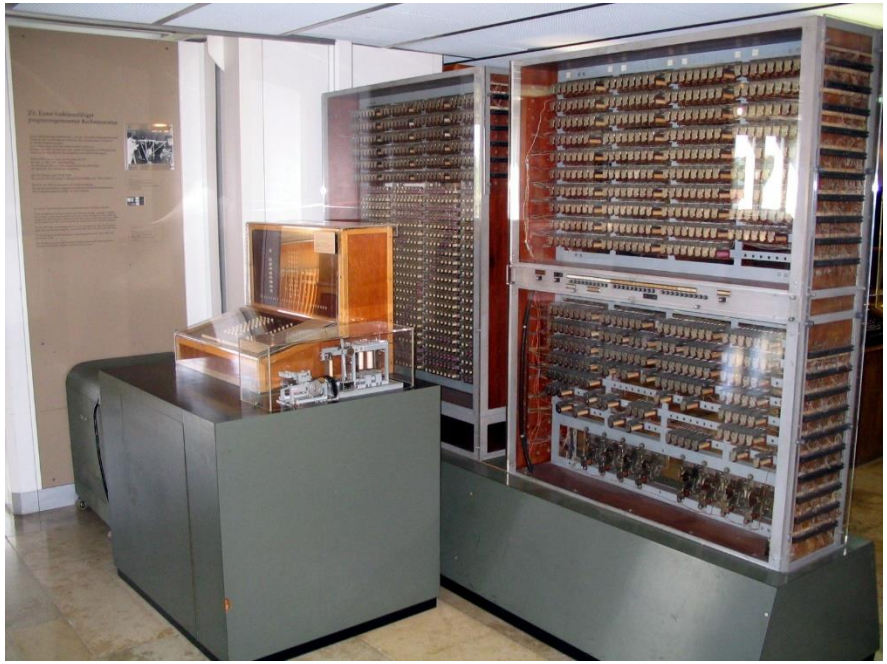
History

*The Z3 was the **world's first working programmable, fully automatic digital computer**. The **Z3** was built with **2,600 relays.** (electromechanical design).*

[Wikipedia]

Year: 1941
Frequency: 4-5 Hz
Average calculation speed:
- Addition: 0.8 s
- Multiplication: 3 s

Weight: 1 ton



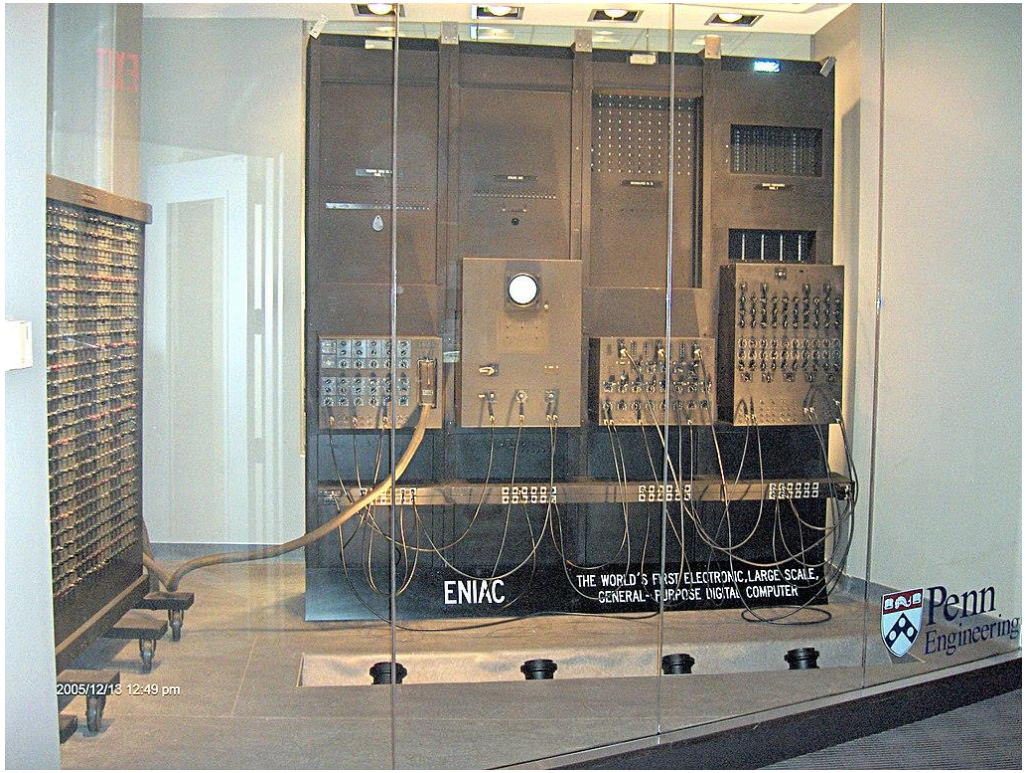
History

- *The **ENIAC** was the first electronic general-purpose computer. It was […] able to solve "a large class of numerical problems".*

[Wikipedia]

Year: 1945
Frequency: 100 kHz
Average calculation speed:
- Addition: 200 μs
- Multiplication: 3 ms

Weight: 27 tons



Today

- Smartphones have a compute power a billion times faster
  - Frequency: Gigahertz
- The most powerful computer (Summit, US) has a peak performance of
  - 122.3 PFlop/s ($P=10^{15}$): Millions of Billions of floating-point operations per second

    [Top500 list, June 2019]
  - 250 Petabyte of storage
  - Consists of
    - 9200 Microprocessors
    - 27,000 graphic cards
  - Analogy: a 122 Petahertz computer
    - (ca. 30,000x a decent gaming PC)

- How can we make use of a computer to solve complex problems?
  - Computers use algorithms to transform data into useful information!

Module Aims

University of Reading

- This module introduces students to computer programming
  - Extensive overview useful for any subsequent programming language
  - Foundations in algorithms and data structures
  - System internals
  - C and C++ programming languages as a demonstration
- In our labs, we will learn a variety of supportive tools and concepts
  - Focus will be on the Linux operating systems (in the Autumn term)
- Learning objectives: (students should be able to)
  - Design and implement moderately complex algorithms in both C and C++
- For more details see the module description

Module Outline: A High-Level Overview

1. Autumn: Fundamentals of programming and the C-language
   1. Problem-Solving
   2. Elementary Algorithms and Data Structures
   3. Introduction to C Programming: Syntax, Semantics
   4. Structured Program Development
   5. Core algorithms and data structures
2. Spring: Object-oriented programming and the C++ language
3. Supplementary in both parts
   - Computing Concepts
   - System Internals: How does a computer work?
   - Tools and environments to develop, analyse programs

Note: There is a steep learning curve in this module as you are new to study at the university. Keep it up: Persevere and Communicate.

This Module in Relation to CS Topics

Topics that will be involved in this module (excerpt from Wikipedia)

4.1 Theoretical computer science

4.1.1 Data structures and algorithms

4.1.4 Programming language theory

4.1.5 Formal methods

4.2 Computer systems

4.2.1 Computer architecture and computer engineering

4.2.2 Computer performance analysis

4.2.5 Computer security and cryptography

4.3 Computer applications

4.3.2 Human-computer interaction

4.4 Software engineering

[Wikipedia]

Why C?

University of Reading

- C is one of the most important programming languages
  - Tiobe Index (C is #2, C++ is #4): Employers appreciate it!
- C allows teaching fundamentals hidden in other languages
- C is widely available on any system (even the Mars Rover)
- C allows using all available architectural features
- C is efficient and performant
- C forms the foundation of many other programming languages
- With this course you will learn the theory of programming

Ref: https://www.tiobe.com/tiobe-index/programming-languages-definition/

Organisation of the Module

- Lecture (2h / week)
  - Delivers the main concepts, some further reading (not necessarily everything will be read!)
  - Includes some active learning (group work)
  - Includes short surveys (bring your mobile or laptop!)
- Tutorial (1h / week; part of a two-hour teaching block in G56)
  - Introduces tools/environments in a presentation
  - Walk-through the initial steps (typical obstacles) together
- Exercise (prescribed 5h / week)
  - Self-study to practice lecture content (feel free to team up!)
  - Each task comes with an estimated time for you to spend on it
  - Contains introductory and harder tasks
  - Store your work in a Git Repository – the portfolio of the course; we provide a template
- Practical (1h / week) – follows the schedule after the tutorial
  - Part 1: Students present their solution/questions to exercise tasks
  - Part 2: We discuss the new exercise such that everyone understands the questions
  - Group work: Some time of practical/tutorial may be used for group work
- Master Class (2h / week) – optional tutorial and exercise discussion
- The first tutorial will provide details on how exercise and practical work!

Supportive Activities

- Master Class (2h / week) – optional tutorial and exercise discussion
  - Monday 11-13 (time tabled)
  - It is an offer from us for you, your attendance is optional
  - E.g., supporting you if you struggle with the exercises or have concerns
- Extreme Programming Club
  - Weekly club where we practice algorithmic thinking
  - Top notch employers are searching for employees with good algorithmic skills
  - Schedule will be determined based on subscribers
  - Visit this URL to express your interest:
    http://bit.ly/cs-clubs
- R.U.Hacking Society Activities (next slide)

Student-led tech society

7th October > VR Night + Pub

28th October > Hacktoberfest'19

December > MLH Local Hack Day

Sign-up here: ruhacking.me

Follow us for updates!

R.U.Hacking

@r.u.hacking

@ReadingUniHack #ruhacking

Newsletter here

Video about the last 24-hour Hackathon

Role of Exercises and Group Work

- Exercises and group work are essential to learning programming skills
  - Skills like algorithmic thinking and programming are like learning a language: it needs practice
  - Your code is mostly automatically assessed: we provide a test framework
  - Exercises provide a starting point for further study
- Some quizzes are provided on Blackboard for your self-study
- Group work: Active learning and fun in groups of 2-4
  - Discuss/Feedback exercises of peers
  - Brainstorm/Design/Solve a small task
- It is imperative for your learning to make the best of these opportunities!

Communication & Online Resources

- Effective communication is a key ingredient for success
  - If you have any problem, communicate, there is no shame to ask! We will practice it!
- Blackboard
  - Announcements, lecture notes, exercises sheets
  - Supplementary material, links
  - Achieved marks will be published as well
- Webpage of the lecture
  - Means to upload your exercises (will be discussed in the first tutorial)
- Slack channel CS3DP for communication (join here)
  - Please use it for any purpose around the topic
    - To retrieve help solving exercises (but do not share solutions!)
    - To share interesting information/link, to ask questions
    - To find peers to work with
  - You can use other slack channels as well!
- Online training courses include Newcomers, Basic, Advanced
  - We are still updating the material; feedback is welcome!

Module Assessment

- Exercises, i.e., practical assignments: 70%
- Written exam: 30% (May/June 2020)
- Practical work is composed of:
  - A weekly assessment handed out one week before (see assessment for details)
    - To work on in weeks
      - 2, 3, 4, 5, 7, 8, 9, 10 (Autumn)
      - 2, 3, 4, 5 (Spring)
    - Each worth 2%
  - A bigger development project from week 5 to week 10 in Spring (worth 31%)
- Two tests in week 5 and week 10, one test in Spring (5% each)
- Success criteria to pass the module: you are required to achieve **both**:
  - A minimum of 30% in the **practical work** (assignments + final project)
  - 40% overall (assessment + exam)

Prescribed Learning Journey

- Or: How to obtain good marks?
- Understand learning outcomes (provided in each slide deck)
- Participate in tutorial and exercises
  - To understand the topic, types of questions, and how to solve issues
  - To get feedback from the lecturers (e.g., if you present exercises) and from peers
  - Schedule time for the exercises, best to team up in learning groups
    - Do the prescribed 5h/week!
  - Always do the easy tasks, if you are busy you may miss some harder tasks
  - Partial solutions are better than no attempt
  - Check the new exercise sheet quickly before the practical
- Do the quizzes
- Do further reading on topics you are interested in
- Team up again to prepare for the exam
- Ask questions to colleagues and us
- We will support your learning journey, but ultimately YOU are responsible for it

Questions?

- Any questions about the module organisation?

- Note: There is a steep learning curve in this module
  - You are new to study at the university
  - We cover many tools and concepts in the tutorial
  - It will pay off to learn them!

- Keep it up: Persevere and Communicate!