

Introduction to Programming

Instructor: Musard Balliu, `musard@chalmers.se`

`http://www.cse.chalmers.se/~musard`

Today's Plan

- ▶ Welcome
- ▶ Course Organization
- ▶ Course Overview
- ▶ A Taste of Java Programming

Welcome

The team

Instructor: Musard Balliu

Teaching Assistants:

- ▶ Jerker Ersare
- ▶ Jonas Kahler
- ▶ Ibtissam Karouach
- ▶ Jiaxin Li
- ▶ Rafael da Silva Martins
- ▶ Omar Abu Nabah
- ▶ Petroula Theodoridou
- ▶ Ivo Vryashkov

The team

Instructor: Musard Balliu

Teaching Assistants:

- ▶ Jerker Ersare
- ▶ Jonas Kahler
- ▶ Ibtissam Karouach
- ▶ Jiaxin Li
- ▶ Rafael da Silva Martins
- ▶ Omar Abu Nabah
- ▶ Petroula Theodoridou
- ▶ Ivo Vryashkov

Course webpage:

<http://gul.gu.se/public/courseId/68749/lang-en/publicPage.do>

Google

Course Organization

Organization

Lectures: Room Alfons, Patricia building

- ▶ Mondays 9:15 - 12:00
- ▶ Wednesdays 9:15 - 12:00
- ▶ No lectures in the week Sept. 28 - Oct. 4
- ▶ Breaks?

Exercise sessions:

- ▶ Mondays 13:15 - 15:00
- ▶ Wednesdays 13:15 - 15:00
- ▶ Thursdays 15:15 - 17:00
- ▶ TAs will be there

Written exam: Wednesday, October 28, 8:30 - 12:30

Organization

Examination and Grading:

- ▶ pass with distinction (VG), pass (G), fail (U)
- ▶ composed out of grade for three assignments and grade for exam

Organization

Examination and Grading:

- ▶ pass with distinction (VG), pass (G), fail (U)
- ▶ composed out of grade for three assignments and grade for exam
- ▶ To pass the course with a VG, you need to get a VG in the exam and at least two VGs in the assignments
- ▶ To pass the course (G) you need to pass the exam and all three assignments

Organization

Assignments:

- ▶ Assignments consists in small programming projects
- ▶ You are allowed to work in groups of 2 students
- ▶ To be solved by a given deadline (strict) and send to me via GUL
- ▶ To be presented during exercise sessions to TAs
- ▶ You have at most 3 chances to pass the assignments

Organization

Literature:

- ▶ In the first part of the course (Lecture 1-8):
Introduction to Programming Using Java, David Eck.
Available online. This is a very good book for this course, in particular because it has approximately the same structure.
- ▶ For the second part of the course:
Java: Learning to Program with Robots, Course Technology 2007. Out of print, but available at:
<http://www.learningwithrobots.com>.
- ▶ For use as reference (and to bring to exam): Y. Daniel Liang
Introduction to Java Programming, Comprehensive, any edition from the 7th on.
Any other comprehensive book will do just as well. If in doubt, try to find a table of contents and email me.

Organization

Communication:

- ▶ Ask in person during the lectures and in the break
- ▶ Check the course webpage
- ▶ Exercise supervision sessions
- ▶ Course email address: `dit948.programming@gmail.com`
- ▶ Instructor email address: `musard@chalmers.se`
- ▶ Direct feedback
- ▶ Course evaluation group?
- ▶ **Messages in GUL will NOT be answered**

Course Overview

An approximate picture of an early computer:

An approximate picture of an early computer:



An approximate picture of an early computer:



- ▶ slow, unreliable, expensive, needed lots of down-time
- ▶ extremely flexible, could be instructed in natural language

History

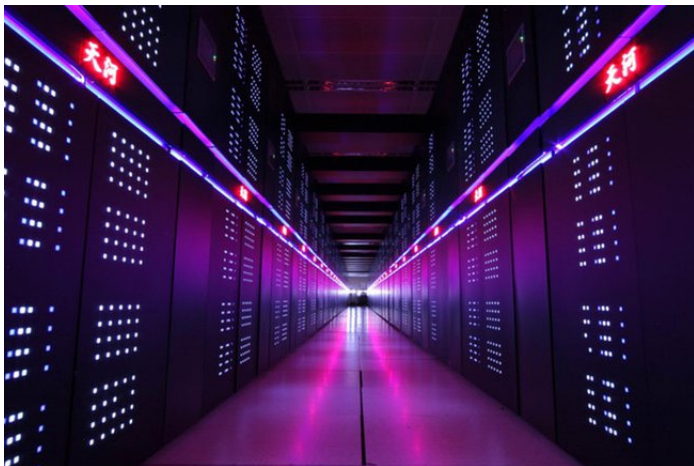


Supercomputers: Tianhe-2

Supercomputers: Tianhe-2



Supercomputers: Tianhe-2



- ▶ fast, cheap, huge memory
- ▶ need to program it

Computability at a glance

A mathematical view

-Problems as mathematical functions

▶ $f : \text{Input} \rightarrow \text{Output}$

-Algorithms

▶ An complete, unambiguous procedure to solve a problem

-Computing Machines

▶ Von Neumann machine

▶ Other models: Turing machine, quantum machine, RAM machine

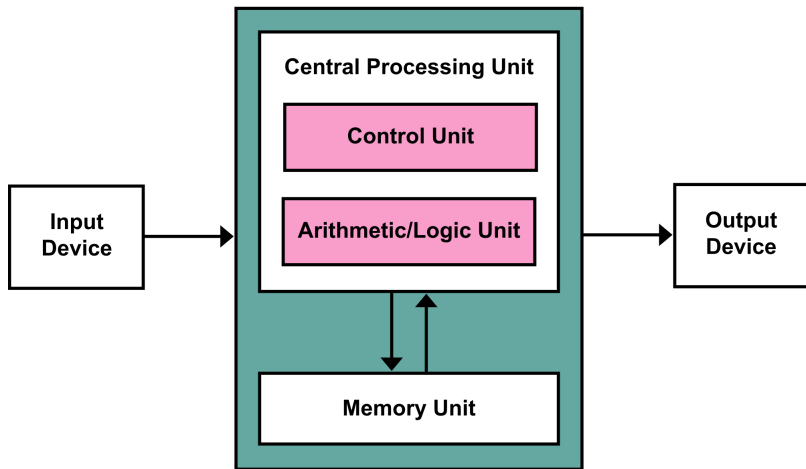
-Programming languages

▶ JAVA

▶ C, C++, Assembly

Von Neumann architecture

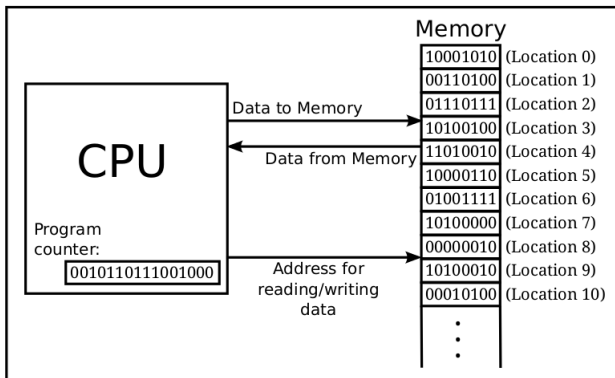
Mental landscape



CPU and Memory

First zoom in

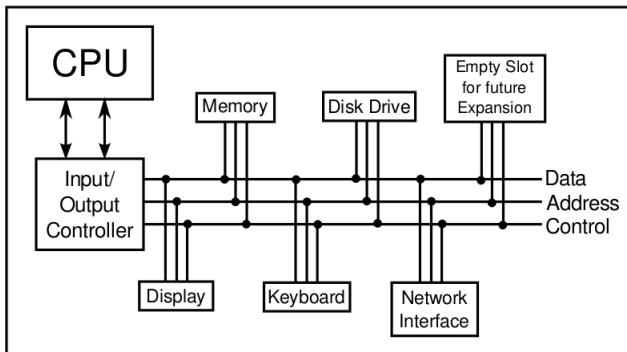
Mental landscape



- CPU and Memory
- Fetch/Decode/Execute cycle
- Binary encoding and machine language

Second zoom in

Mental landscape



-OS and Devices

Compilers and Interpreters

Mental landscape

- ▶ Machine code: directly executed on CPU
- ▶ High-level code: most programs are written in JAVA, C, C++ ecc.
- ▶ Compiler: Translate a program from one language to another
- ▶ Interpreter: Translates a program from one language to machine code instruction by instruction and executes it on a CPU

Compilers and Interpreters

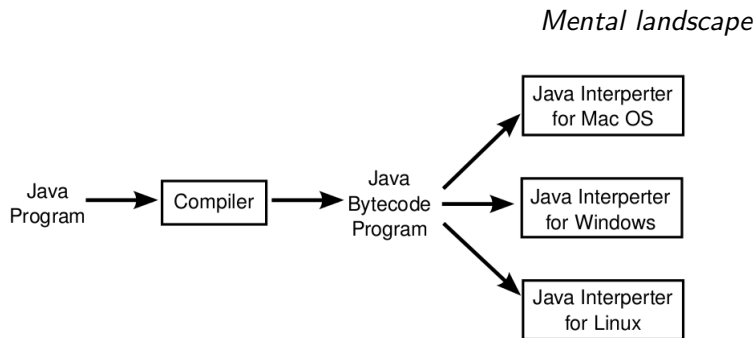
Mental landscape

- ▶ Machine code: directly executed on CPU
- ▶ High-level code: most programs are written in JAVA, C, C++ ecc.
- ▶ Compiler: Translate a program from one language to another
- ▶ Interpreter: Translates a program from one language to machine code instruction by instruction and executes it on a CPU

Compilers vs Interpreters

- ▶ CPU-independence: interpreters allow to execute a program meant for one type of CPU on a completely different type of CPU
- ▶ Interpreter much smaller and easier to implement

Java and JVM



- ▶ Java compiler: translates Java program into intermediate language (bytecode)
- ▶ Java Virtual Machine: machine that executes bytecode programs, like an interpreter
- ▶ Flexibility, security
- ▶ Just-in-time compiler (JIT)

INTRODUCTION TO JAVA

Introduction to Java

Course objectives:

- ▶ basics of imperative programming
- ▶ basics of the object-oriented programming
- ▶ using and understanding APIs, e.g.:
 - ▶ graphical user interfaces (GUIs)
 - ▶ multithreaded programming
 - ▶ database connectivity

Fundamental abstractions

Introduction to Java

Imperative programming:

- ▶ Data and instructions
- ▶ Variables, types and assignments
- ▶ Control structures and subroutines

Object-oriented programming

- ▶ Bottom-up vs top-down
- ▶ Classes and objects
- ▶ Information hiding and polymorphism
- ▶ Inheritance and GUIs

Introduction to Java

The only way to learn a new programming language is by writing programs in it. The first program to write is the same for all languages:

Print the words:

Hello, World!

Introduction to Java

The problem is that even a simple program such as “Hello, World!” requires a certain amount of machinery to run in Java.

To simplify matters, you should use the files `Template.java` and the small `dit948` library available from the course home page.

Ask the supervisors to help you use these.

“Hello, World!” in Java, beginner style

We begin by opening a new file, appropriately entitled HelloWorld.java.

We then copy the contents of the file `Template.java` in the new file and make the appropriate changes. In this case, we uncomment the line `// import static dit948.SimpleIO.*;` since we want to do some printing and change the word `Template` to `HelloWorld`.

We then enter the instructions to do the printing inside the curly braces that delimit the *body* of the main function.

The end result

Exercises

- ▶ Read the first chapter of *Introduction to Programming Using Java* by David Eck <http://math.hws.edu/javanotes/>
- ▶ Install JDK 1.7 on your computer and make sure you can compile and run the “Hello, World!” program