

Programming – Lecture 1

Welcome

Administrivia

Introduction (Chapter 1)

- Brief history of computing
- CS, Algorithms, SW Engineering, Errors
- Compilation vs. Interpretation

We

Lecturer:

Prof. Reinhard von Hanxleden,
R1117, rvh@informatik.uni-kiel.de

Staff (*Mitarbeiter/innen*):

Christoph Daniel Schulze (lead),
R1112, cds@

Sören Domrös,
R1113, sdo@

Steven Smyth,
R1118, ssm@

Alexander Schulz-Rosengarten,
R1113, als@

Student Staff

(*Hilfskräfte*):

Jeremia Böhmig

Sascha Ecks

Thora Luise Fiedler

Johanna Menzel

Patrick Michalski

Jette Petzold

Hendrik Sauer

Massoud Vincent Shariyari

You?

Please visit

<http://pingo.upb.de/643250>

Literature

- "The Art and Science of Java,"
by Eric Roberts
- "Java ist auch eine Insel,"
by Christian Ullenboom
- See also wiki \leadsto Begleitmaterialien

Code of Conduct

I:

- Start and finish on time
- Try to deliver high-quality lectures
- Listen to your concerns
- Make sure you have a fair chance of passing this class
- Prepare you for what's ahead

You:

- Are **punctual**
- Do not disturb others (**laptops/tablets only in last row**)
- Work on problems yourself (**no plagiarism/code sharing**)
- **Check your schedule** before registering your time-preferences for the practical class
- Familiarize yourself with infprogoo.de and the linked wiki, including requirements for "*Endklausurzulassung*"

Programming – Lecture 1

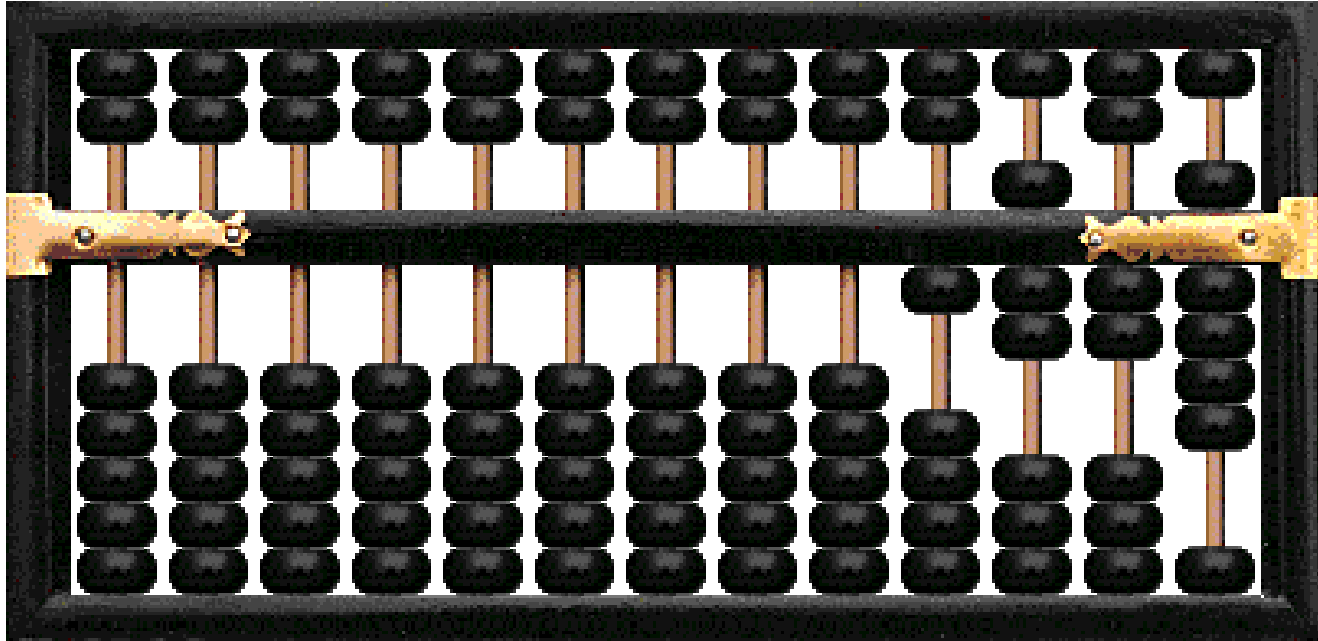
Welcome

Administrivia

Introduction (Chapter 1)

- Brief history of computing
- CS, Algorithms, SW Engineering, Errors
- Compilation vs. Interpretation

A Brief History of Computing





Charles Babbage
(1791 – 1871)



Augusta Ada Byron,
Lady Lovelace
(1815 – 1852)



CC BY-SA 3.0

Konrad Zuse
(1910 – 1995)



CC BY-SA 3.0

Z1

Computer Science

The science of problem solving in which the solutions happen to involve a computer

Solving a problem by computer:

1. Algorithmic design
2. Coding

Algorithm

Ninth-century Persian mathematician Abu Ja'far Mohammed ibn Mûsâ al-Khowârizmî

Properties:

1. Clearly and unambiguously defined
2. Effective
3. Finite

Software Engineering

Discipline of writing programs so that they can be understood and maintained by others

Programming Errors

Distinguish

- Syntax errors
- Bugs

*„The first step [in all of my inventions] is an intuition, and comes with a burst, then difficulties arise – this thing gives out and [it is] then that ‘**Bugs**’ – as such little faults and difficulties are called – show themselves [...].“*

Thomas Edison, 18 Nov 1878

9/9

0800 Antan started
 1000 " stopped - antan ✓

13 ⁰⁰ MC (032)	MP - MC	1.982147000	1.2700	9.037847025
(033)	PRO 2	2.130476415		9.037846995 connect
	connect	2.130476415		4.615925059(-2)
		2.130676415		

Relays 6-2 in 033 failed special speed test
 in Relay .. 11.000 test.

Relay
 2145
 Relay 337

1100 Started Cosine Tapc (Sine check)
 1525 Started Mult + Adder Test.

1545



Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

~~1630~~ 1630 antan started.
 1700 closed down.

[U.S. Naval Historical Center Online Library]

Page from the Harvard Mark II
 electromechanical computer's log, 1947

Why Java in InfProgOO?

- First of all: why ask that question? This class is about *programming principles*, not about a particular *programming language*.
- Having said that: to really learn the principles, one should write real programs, for which a concrete language X must be chosen.
- Choice of X should be guided by a few questions
...

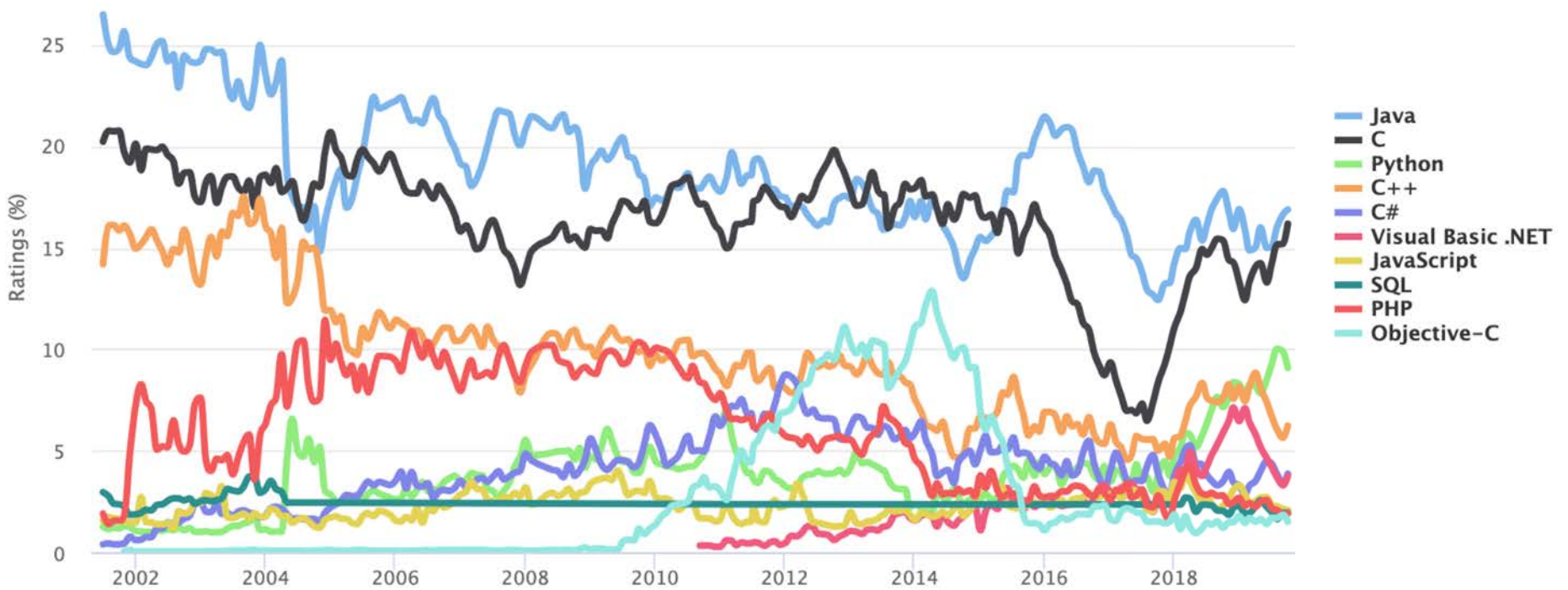
Questions to Ask Concerning X

1. Does X illustrate imperative/object-oriented programming?
2. Is it reasonably easy to learn X, are there good resources available for learning X?
3. Does the teaching staff have good expertise on X?
4. Do later classes and software projects at CAU also use X?
5. Does knowledge of X help me after graduation?

Clear “yes” to all of these for X = Java.

Programming Language	2019	2014	2009	2004	1999	1994	1989
Java	1	2	1	1	3	-	-
C	2	1	2	2	1	1	1
Python	3	7	6	6	22	20	-
C++	4	4	3	3	2	2	2
Visual Basic .NET	5	9	-	-	-	-	-
C#	6	5	5	7	17	-	-
JavaScript	7	8	8	9	13	-	-
PHP	8	6	4	5	-	-	-
SQL	9	-	-	89	-	-	-
Objective-C	10	3	26	36	-	-	-
Lisp	32	17	16	13	14	5	3
Pascal	219	15	14	88	7	3	21

www.tiobe.com



See also <http://www.tiobe.com/tiobe-index/programming-languages-definition/>

Java – the undisputed winner

Java still continues to top the most popular programming languages charts as it did a year ago. According to TIOBE's data, Java has secured the first and second positions more than any other languages for about a couple of decades. A large number of renowned companies use Java to develop software and applications so if you happen to know Java, you definitely won't have to struggle to find a job. The major reasons behind the popularity of Java are its portability, scalability and a large community of users.

<https://fossbytes.com/most-popular-programming-languages/>

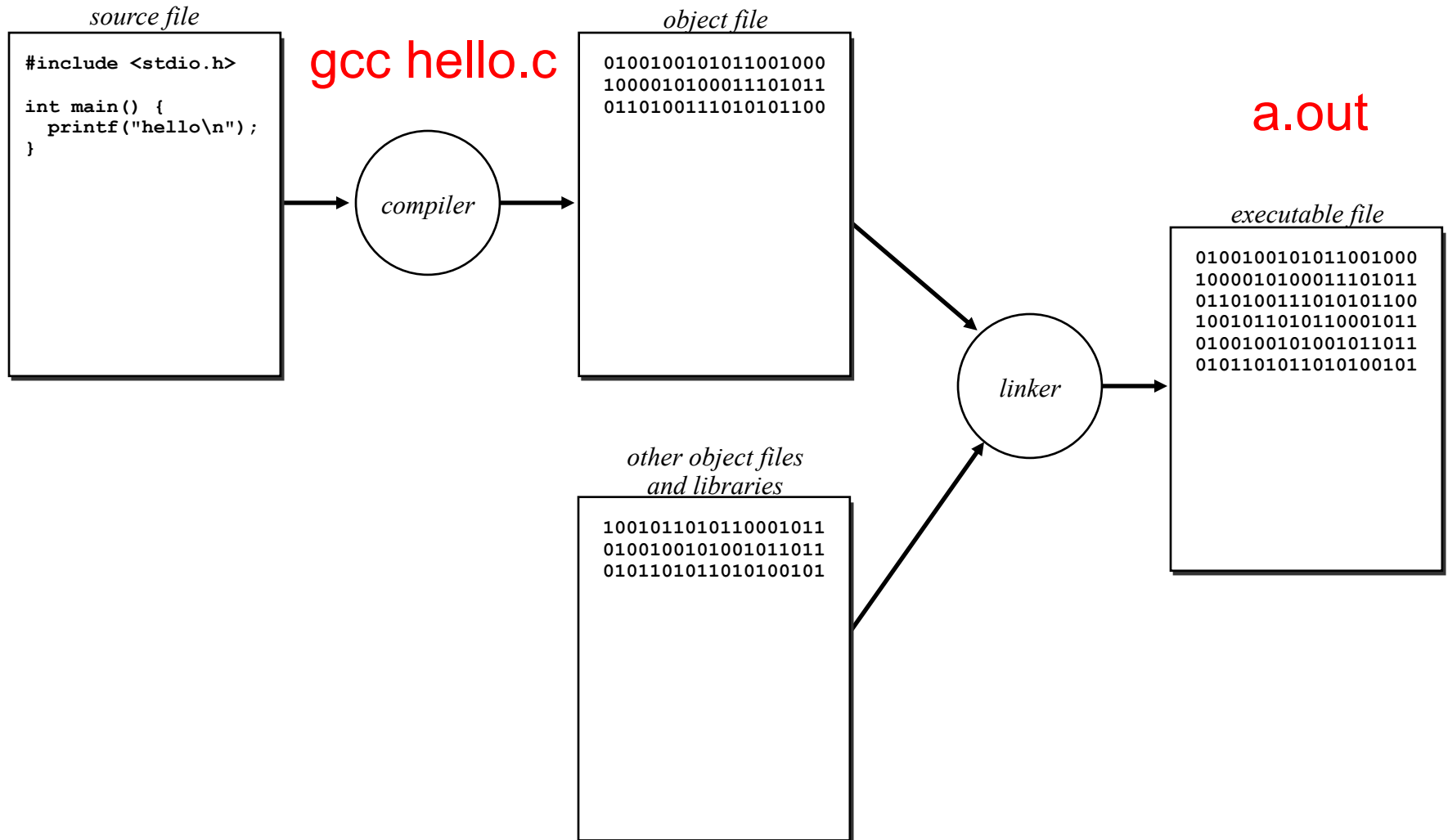
June 2018

Java – Design Goals

- Simple, object oriented, familiar
- Robust, secure
- Architecture neutral, portable
- High performance
- Interpreted, threaded, dynamic

White Paper: The Java Language Environment
James Gosling and Henry McGilton, May 1996

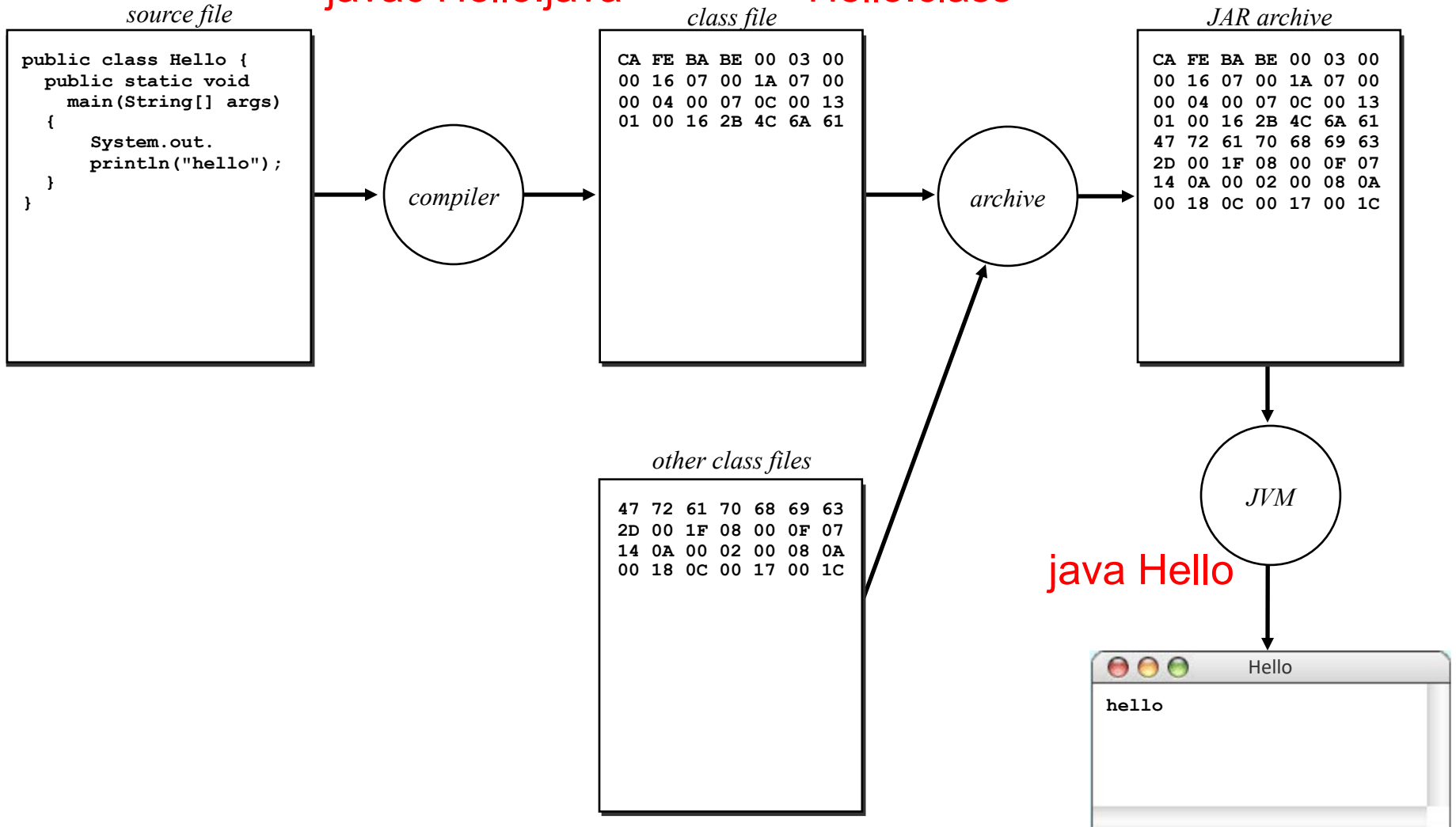
Classic Compilation Process



The Java Interpreter

`javac Hello.java`

`Hello.class`



Summary

- Long history of computing, recent acceleration
- Defined “CS”, “Algorithm”, “SW Engineering”
- Syntax errors vs. Bugs
- Discussed Java design goals
- Compilation vs. Interpretation