On the difficulties to deliver a 1st-semester course "Foundations of Programming" at HU: prerequisites, contents, success rate

Klaus Bothe, Olga Schiemangk
Humboldt University Berlin

16th Workshop "Software Engineering Education and Reverse Engineering
Jahorina, BiH, 22th – 26th August 2016

The problem

Difficulties at a 1st-semester course "Foundations of Programming“ at HU

- Completely different prerequisites from school:
  One lecture for so different students?
- Don‘t know the number of really interested students
- Contents and methodology of the lecture

Contents

Bachelor at HU: Schedule Overview

1. Semester	Fundamentals of Programmierung (OO, Java)		Basics of theoretical informatics				Mathematics 1: Linear Algebra		Second subject
2. Semester	Algorithms and data structures		Compiler construction		Digital systems: basics and architectures		Informatics and Society		Second subject
3. Semester	Software Engineering		Logic in informatics		Seminar		Mathematics 2: Analysis		Second subject
4. Semester	Operating systems		Database systems		Communication systems: basics and network architectures		Mathematics 3: Numerics & Stochastics		Second subject
5. Semester	Semester Project		Elective modules						Second subject
6. Semester	Bachelor thesis		Elective modules						Second subject

Bachelor at HU: Schedule Overview

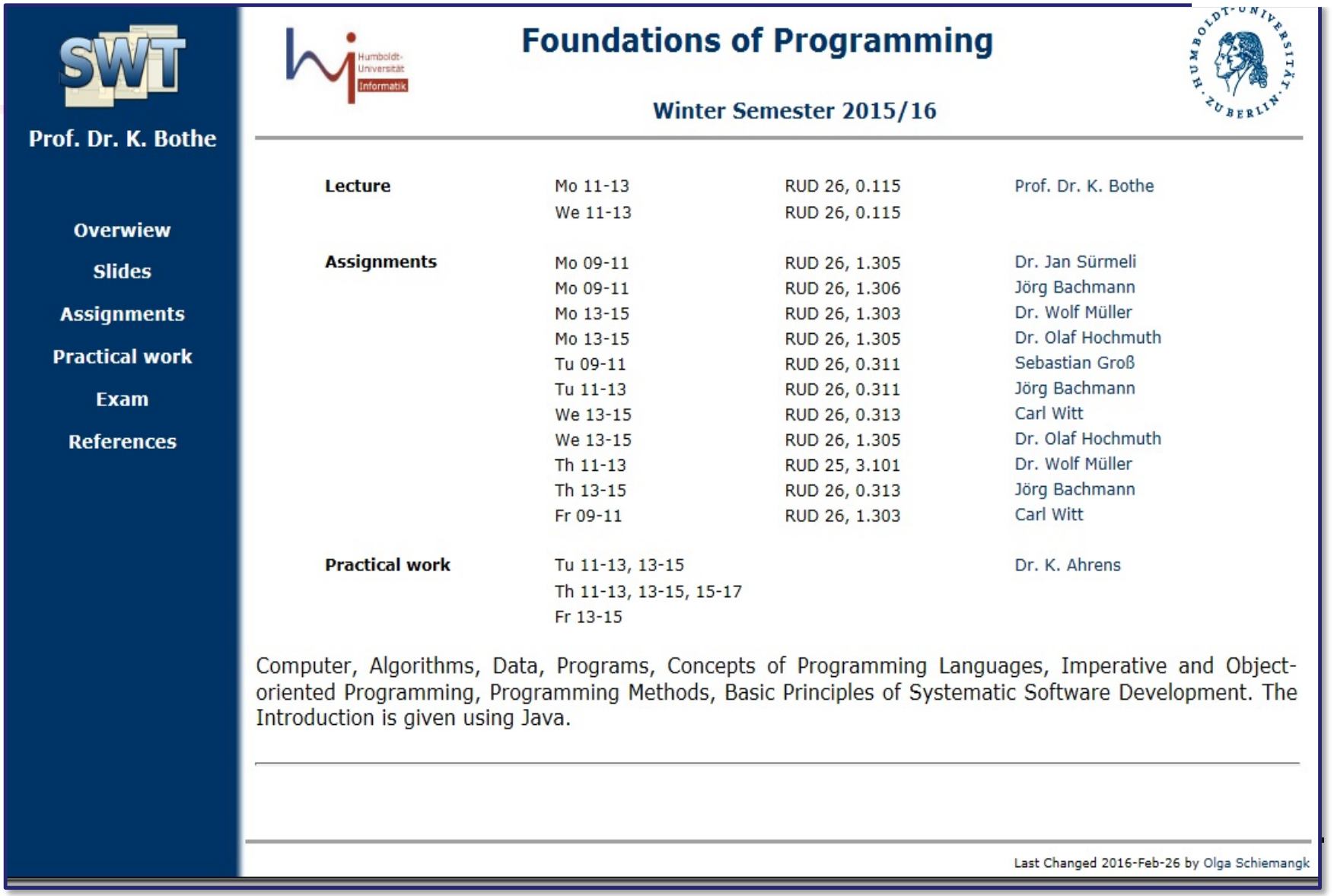
1. Semester	Fundamentals of Programmierung (OO, Java)		Basics of theoretical informatics						
2. Semester	Algorithms and data structures		Compiler construction		Digita bas arch				
3. Semester	Software Engineering		Logic in informatics		Se				
4. Semester					mmunication ns: basics and k architectures		Mathematics 3: Numerics & Stochastics		Second subject
5. Semester	Project				ive modules				Second subject
6. Semester	Bachelor thesis		Elective modules						Second subject

Organisation:
- Lecture: 4 lecture hours / week
- 16 weeks
- assignments
- practical work

Substantial effort:
12 ECTS

Contents and goals:
- Imperative programming
- Object-oriented programming
- Java

GdP Website: Organisation



SWT

Prof. Dr. K. Bothe

Overwiew
Slides
Assignments
Practical work
Exam
References

Humboldt-Universität Informatik

Foundations of Programming

Winter Semester 2015/16

HUMBOLDT-UNIVERSITÄT ZU BERLIN

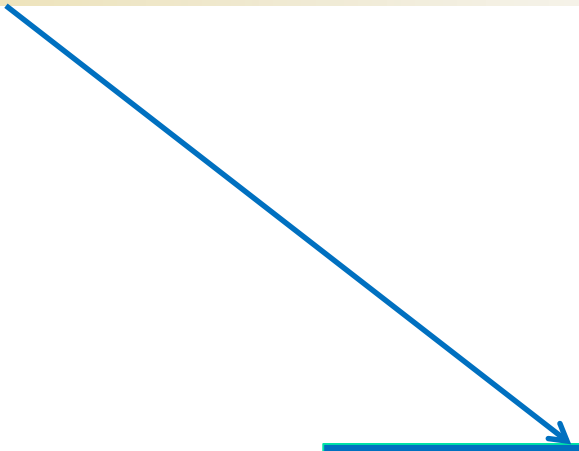
Lecture	Mo 11-13	RUD 26, 0.115	Prof. Dr. K. Bothe
	We 11-13	RUD 26, 0.115	
Assignments	Mo 09-11	RUD 26, 1.305	Dr. Jan Sürmeli
	Mo 09-11	RUD 26, 1.306	Jörg Bachmann
	Mo 13-15	RUD 26, 1.303	Dr. Wolf Müller
	Mo 13-15	RUD 26, 1.305	Dr. Olaf Hochmuth
	Tu 09-11	RUD 26, 0.311	Sebastian Groß
	Tu 11-13	RUD 26, 0.311	Jörg Bachmann
	We 13-15	RUD 26, 0.313	Carl Witt
	We 13-15	RUD 26, 1.305	Dr. Olaf Hochmuth
	Th 11-13	RUD 25, 3.101	Dr. Wolf Müller
	Th 13-15	RUD 26, 0.313	Jörg Bachmann
	Fr 09-11	RUD 26, 1.303	Carl Witt
Practical work	Tu 11-13, 13-15		Dr. K. Ahrens
	Th 11-13, 13-15, 15-17		
	Fr 13-15		

Computer, Algorithms, Data, Programs, Concepts of Programming Languages, Imperative and Object-oriented Programming, Programming Methods, Basic Principles of Systematic Software Development. The Introduction is given using Java.

Last Changed 2016-Feb-26 by Olga Schiemangk

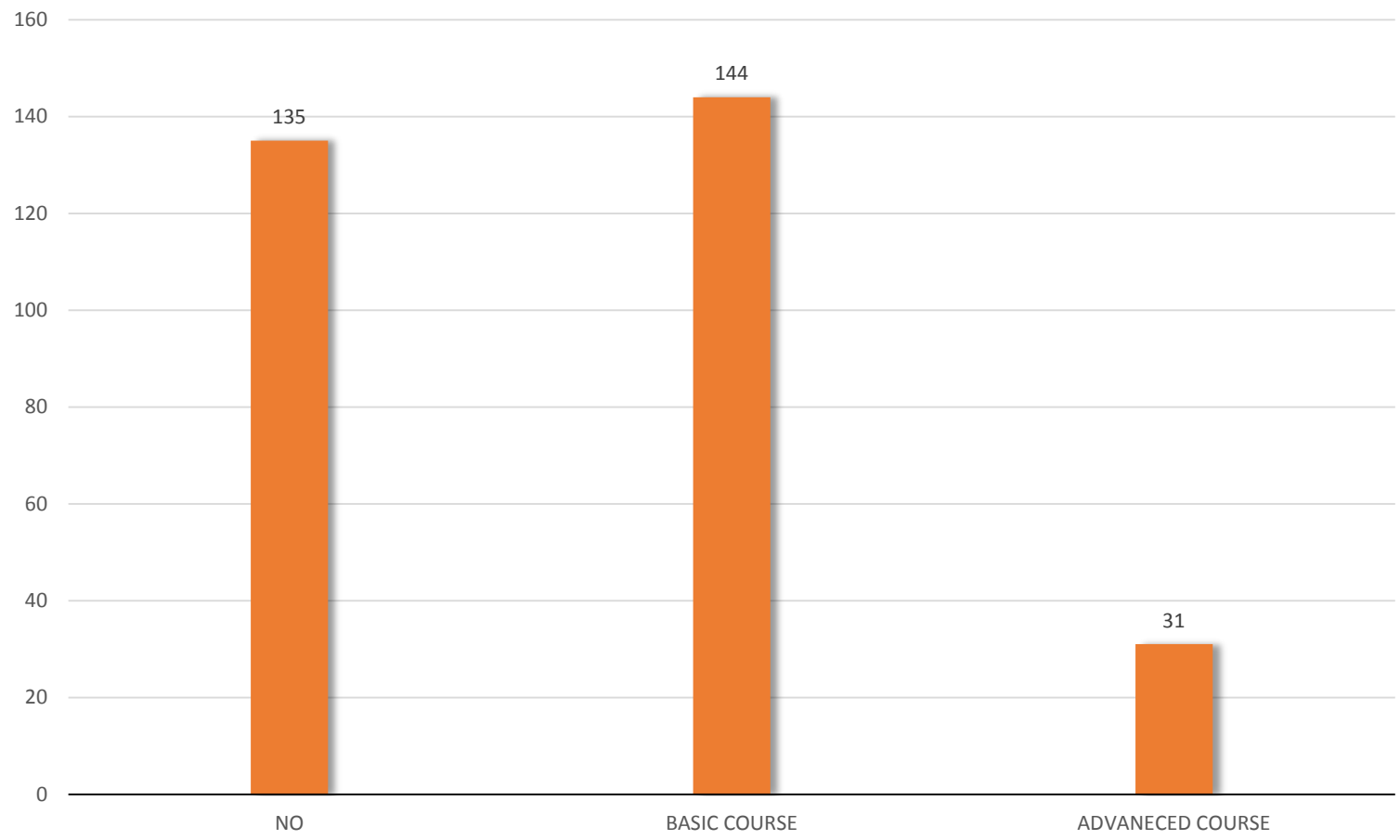
Contents

- Bachelor Curriculum at HU
- Prerequisites of new students
- How many students?
- Course contents
- Results
- Summary

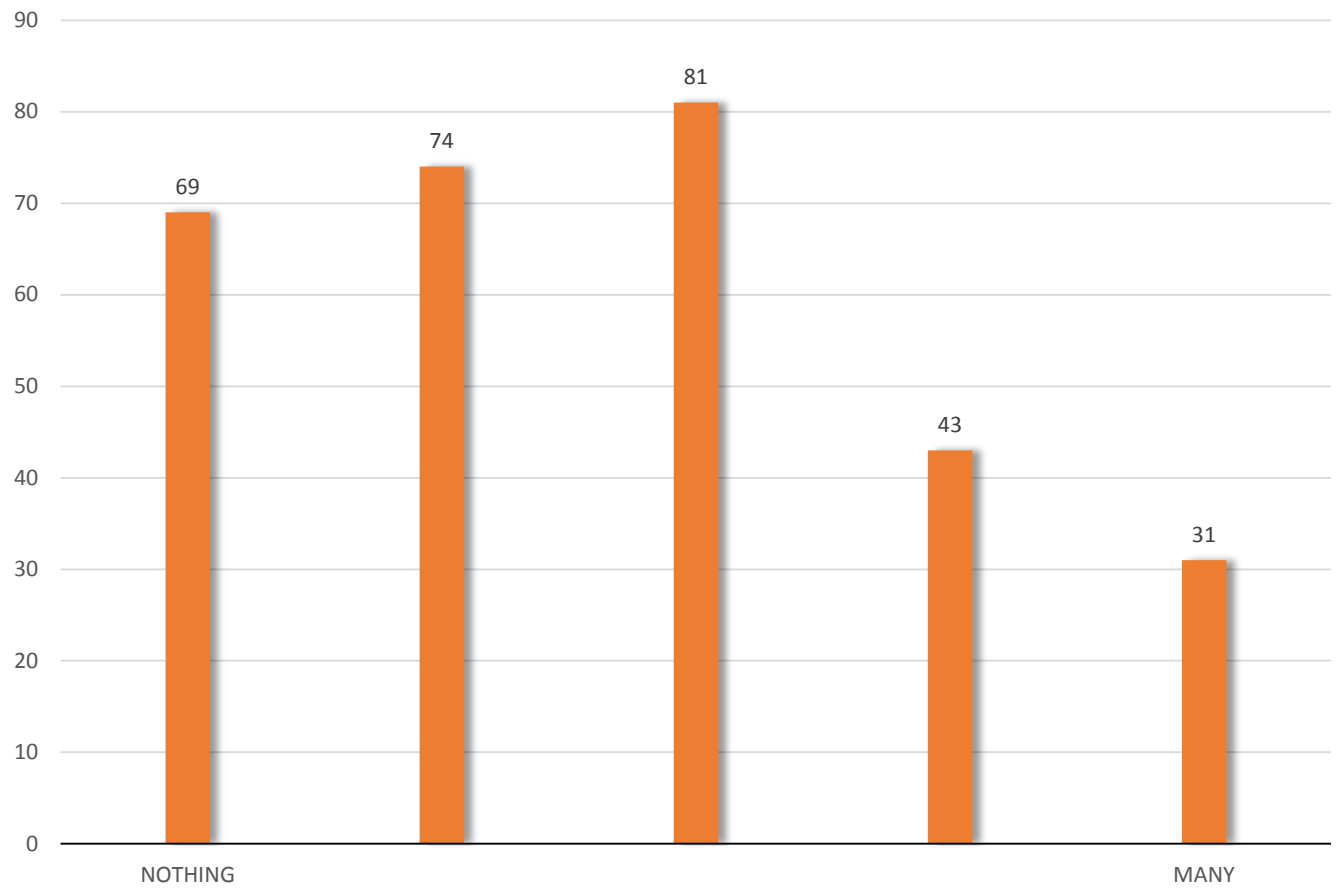


Questionnaire
(at the beginning)

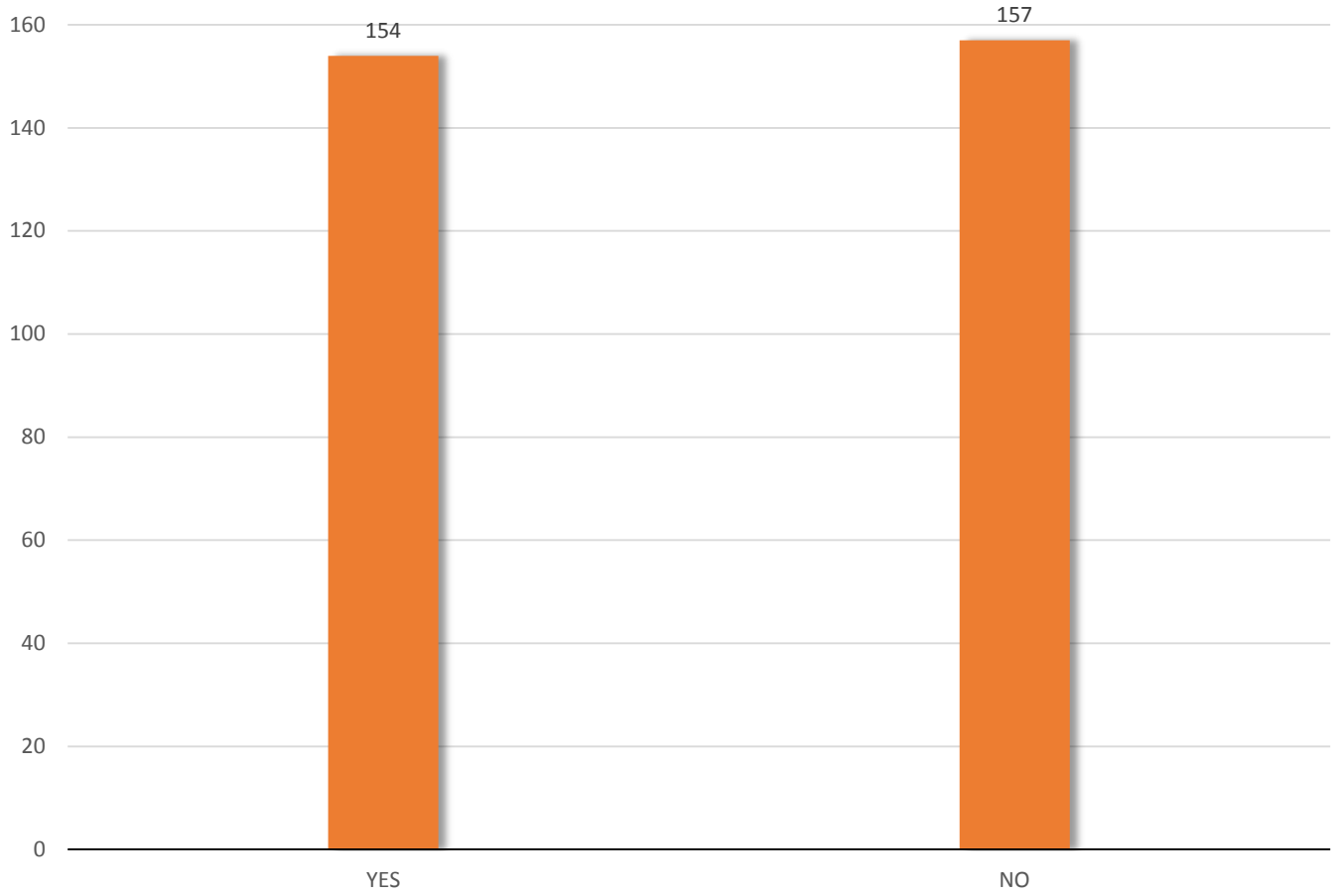
Did you attend a course „Informatics“ at school?



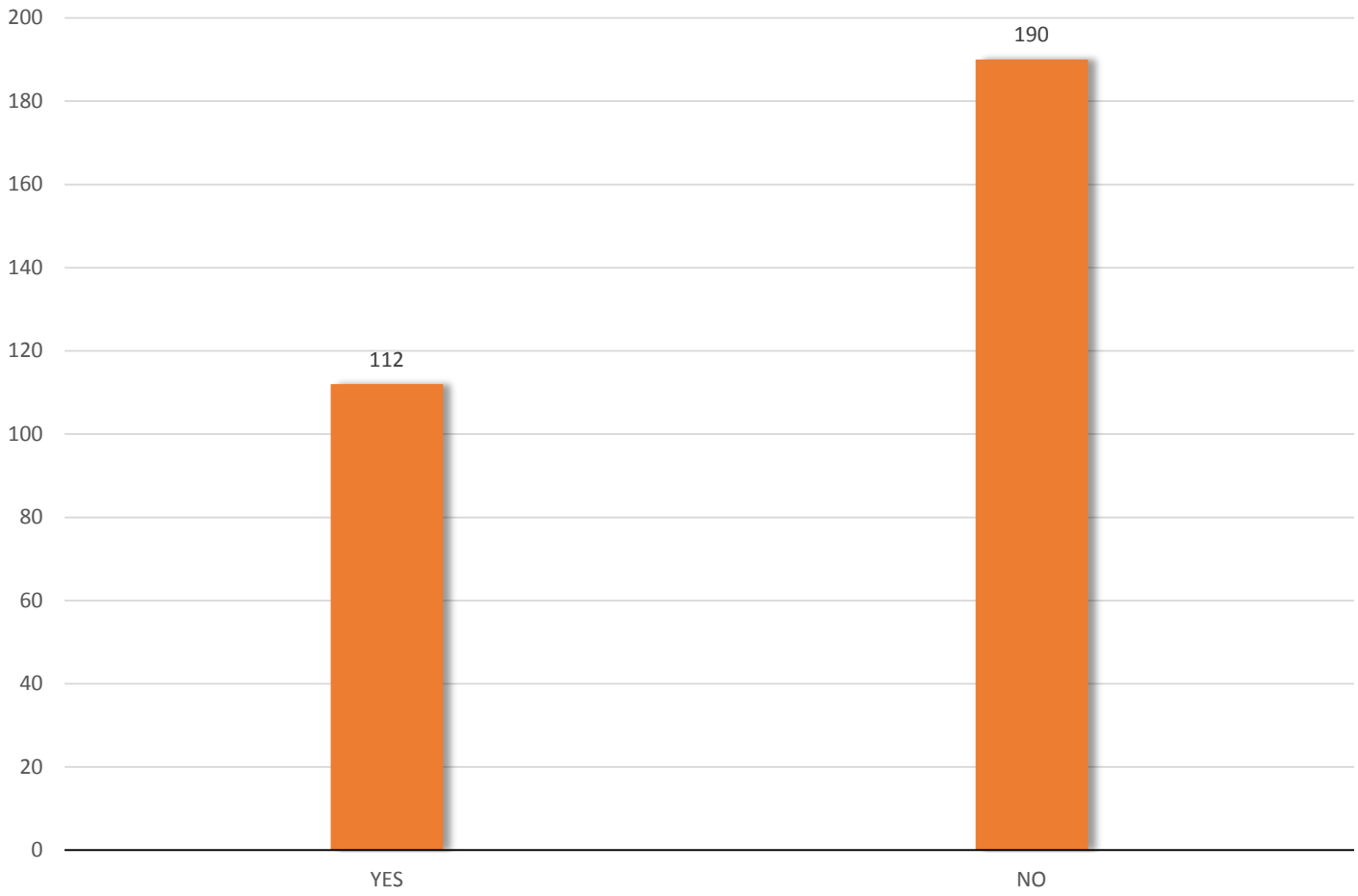
Did you have knowledge of programming bevor studies?



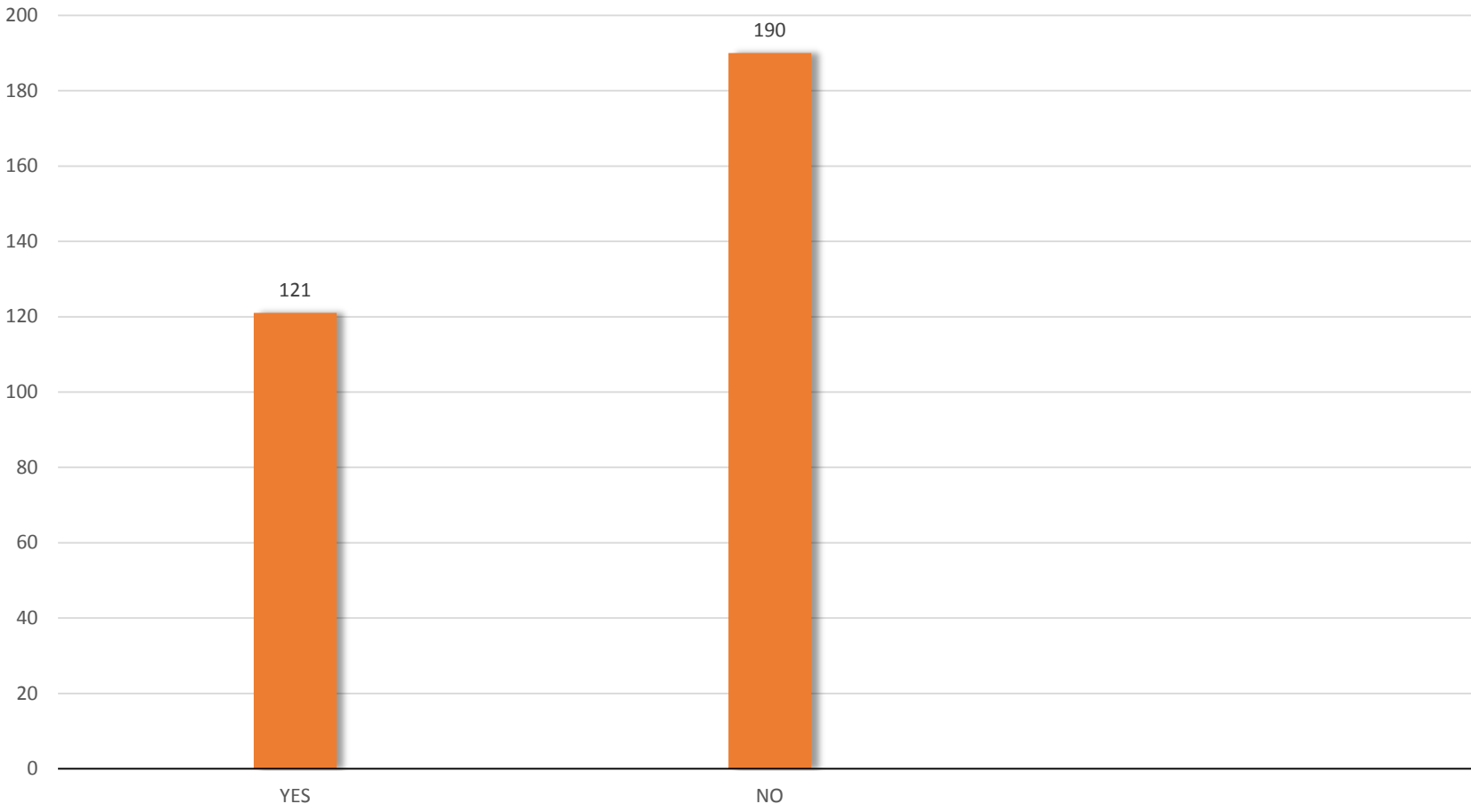
Did you attend an advanced course „Mathematics “ at school?



Do you know what object-oriented programming is?



Do you know some sorting algorithms (e.g. Quicksort)?



Conclusion:

- one course for all new students
  not easy to design

Contents

- Bachelor Curriculum at HU
- Prerequisites of new students
- How many students?
- Course contents
- Results
- Summary

How many students in WS 2015/16?

426 students enrolled:

- 309 BA informatics
- 117 others (teachers, Biophysics ...)

44 % not interested

Enrolled – but not interested

309 enrolled students in BA Informatics, BUT

- 44% (137 of 309 did not study at all) **1)**
- 13% (41 not successful at the exam)
- 43% (131 successful)

1) Enrolled to wait for another study (e.g. in Medicine, Psychology); Informatics is free to enroll (no upper limit)

Ruduction rate over the weeks

In the beginning:

423 enrolled for assignments and Java lab

1. task: 368 solutions
2. task: 335 solutions
3. task: 299 solutions
4. task: 292 solutions
5. task: 256 solutions
6. task: 232 solutions

Exam:

202 participants (failed: 38, succeeded: 164)

Contents

- Bachelor Curriculum at HU
- Prerequisites of new students
- How many students?
- Course contents
- Results
- Summary

Course contents ...?

Foundations

Compiler, Syntax, Data ...

Imperative Programming

Statement, variable, algorithm, data type ...

Object-oriented Programming

Abstract data type, class, object; inheritance, polymorphism, generics, exception handling, event handling ...

Java

Objects first?

Website: Contents 1 (excerpt)

Chapter	Slides	Program examples in Java
Part I: Foundations		
I.1 What is informatics?	I.1-1s.pdf, I.1-4s.pdf	
I.2 Computer	I.2-1s.pdf, I.2-4s.pdf	
I.3 Basics of working with computer	I.3-1s.pdf, I.3-4s.pdf	
I.4 Data	I.4-1s.pdf, I.4-4s.pdf	
I.5 Algorithms	I.5-1s.pdf, I.5-4s.pdf	
I.6 Programming languages	I.6-1s.pdf, I.6-4s.pdf	
I.7 Syntax: Grammar, EBNF	I.7-1s.p1.pdf, I.7-4s.p1.pdf I.7-1s.p2.pdf, I.7-4s.p2.pdf	
I.8 Software development	I.8-1s.pdf, I.8-4s.pdf	
Part II: Concepts of imperative languages		
		Programs-Part-II.pdf
II.1 Preliminary note	II.1-1s.pdf, II.1-4s.pdf	
II.2 Compiler, interpreter, virtual machine	II.2-1s.pdf, II.2-4s.pdf	Hello.java
II.3 Essential components of imperative languages	II.3-1s.pdf, II.3-4s.pdf	Temperature.java Keyboard.java
II.4 Standard libraries: Java-API	II.4-1s.pdf, II.4-4s.pdf	
II.5 Choice: conditional statements	II.5-1s.pdf, II.5-4s.pdf	
II.6 Iteration: loop statements	II.6-1s.pdf, II.6-4s.pdf	TemperatureTable.java
II.7 Methods	II.7-1s.pdf, II.7-4s.pdf	Factorials.java

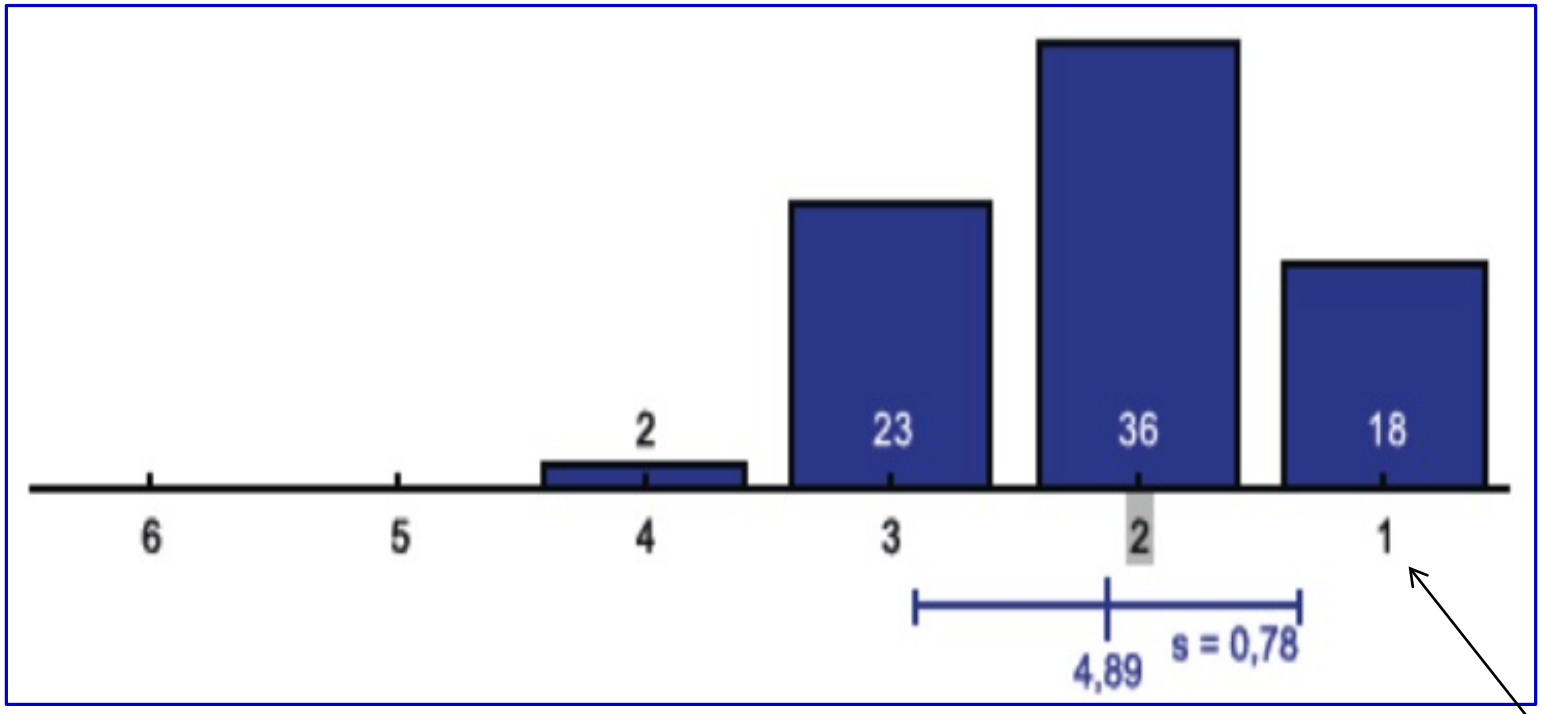
Website: Contents 2 (excerpt)

Part III: Object-oriented concepts and software development		
		Programs-Part-III.pdf
III.1 Basic concepts of object-oriented programming (1): abstract data types, objects, classes	III.1-1s.pdf, III.1-4s.pdf	Stack.java reversing.java reversing2.java
III.2 Object-oriented programming: basic examples	III.2-1s.pdf, III.2-4s.pdf	BracketStructur.java Time.java Schedule.java
III.3 Basic concepts of object-oriented programming (2): class variable and methods	III.3-1s.pdf, III.3-4s.pdf	TimeC.java ScheduleC.java
III.4 Component types	III.4-1s.pdf, III.4-4s.pdf	
III.5 Basic concepts of object-oriented programming (3): inheritance, polymorphism, dynamic binding	III.5-1s.pdf, III.5-4s.pdf	Time2.java
III.6 Basic concepts of object-oriented programming (4): generic classes	III.6-1s.pdf, III.6-4s.pdf	StackForChar.java BuildPairs.java BuildPairsBounds.java StackGen.java
III.7 Chained structures: lists	III.7-1s.pdf, III.7-4s.pdf	IntList.java List.java Stack1.java
III.8 Basic concepts of object-oriented programming (5): interfaces	III.8-1s.pdf, III.8-4s.pdf	ScheduleInt.java ScheduleAbstr.java ReversingNU.java KeyboardIApp.java Print.java
III.9 Exception handling	III.9-1s.pdf, III.9-4s.pdf	Exception.java TryCatch.java TryCatchAll.java Finally.java

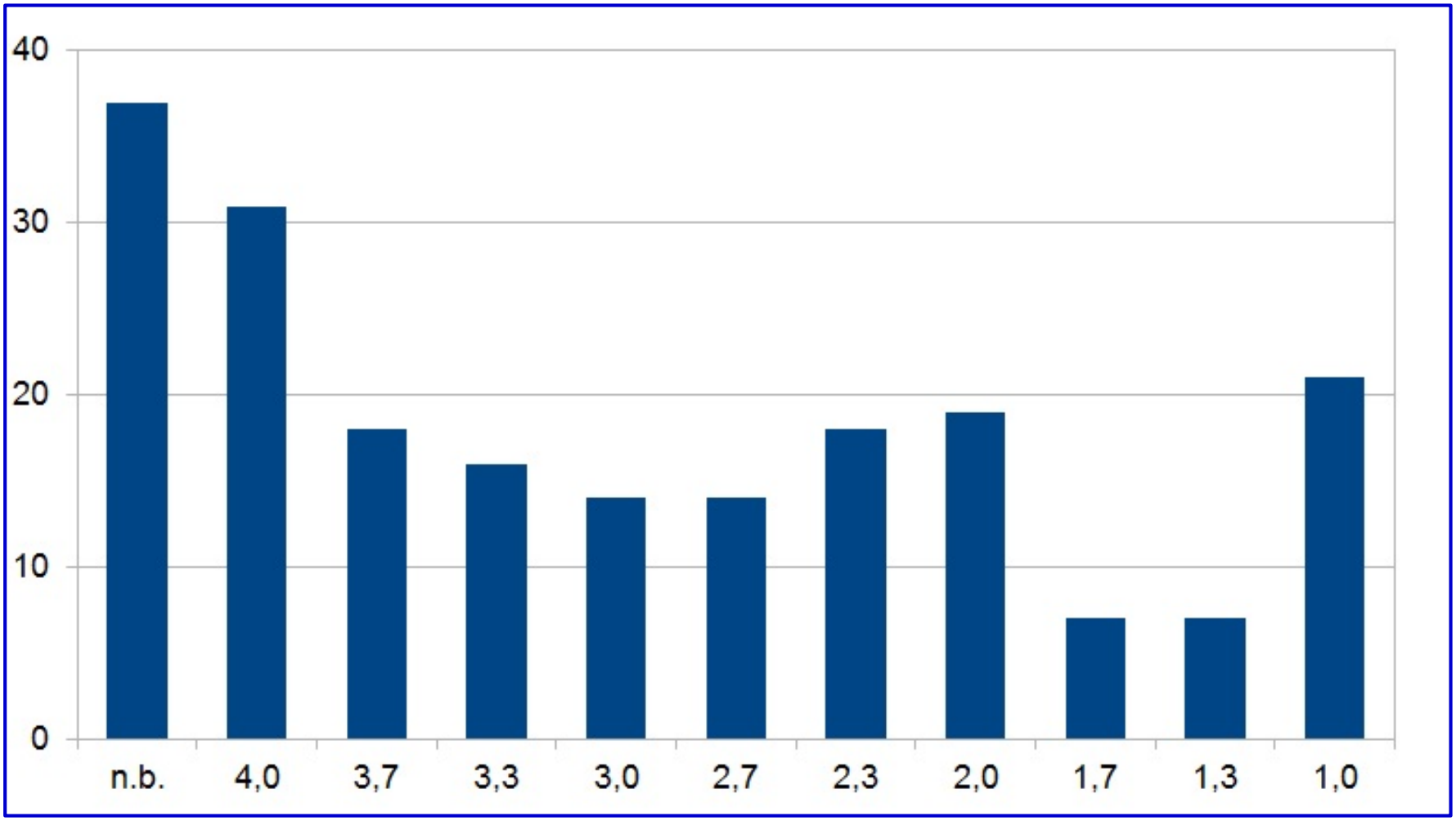
Contents

- Bachelor Curriculum at HU
- Prerequisites of new students
- How many students?
- Course contents
- Results
- Summary

Feedback of students: quality of lecture (anonymous)



Exam: Results



Failed

Best grade

Contents

- Bachelor Curriculum at HU
- Prerequisites of new students
- How many students?
- Course contents
- Results
- Summary

Summary

- Number of students not clear
- Completely different prerequisites of new students: a compromise for course contents
- Success rate: 43%
- Structure: 3 parts
  - Overview
  - Imperative programming
  - Object-oriented programming

Thank you for attention

Appendix:

Complete contents

Website: Contents 1

Chapter	Slides	Program examples in Java
Part I: Foundations		
I.1 What is informatics?	I.1-1s.pdf, I.1-4s.pdf	
I.2 Computer	I.2-1s.pdf, I.2-4s.pdf	
I.3 Basics of working with computer	I.3-1s.pdf, I.3-4s.pdf	
I.4 Data	I.4-1s.pdf, I.4-4s.pdf	
I.5 Algorithms	I.5-1s.pdf, I.5-4s.pdf	
I.6 Programming languages	I.6-1s.pdf, I.6-4s.pdf	
I.7 Syntax: Grammar, EBNF	I.7-1s.p1.pdf, I.7-4s.p1.pdf I.7-1s.p2.pdf, I.7-4s.p2.pdf	
I.8 Software development	I.8-1s.pdf, I.8-4s.pdf	
Part II: Concepts of imperative languages		
		Programs-Part-II.pdf
II.1 Preliminary note	II.1-1s.pdf, II.1-4s.pdf	
II.2 Compiler, interpreter, virtual machine	II.2-1s.pdf, II.2-4s.pdf	Hello.java
II.3 Essential components of imperative languages	II.3-1s.pdf, II.3-4s.pdf	Temperature.java Keyboard.java
II.4 Standard libraries: Java-API	II.4-1s.pdf, II.4-4s.pdf	
II.5 Choice: conditional statements	II.5-1s.pdf, II.5-4s.pdf	
II.6 Iteration: loop statements	II.6-1s.pdf, II.6-4s.pdf	TemperatureTable.java
II.7 Methods	II.7-1s.pdf, II.7-4s.pdf	Factorials.java TimeTable.java
II.8 Expressions, simple operator types	II.8-1s.p1.pdf, II.8-4s.p1.pdf II.8-1s.p2.pdf, II.8-4s.p2.pdf	Unicode.java
II.9 Programming style guides	II.9-1s.pdf, II.9-4s.pdf	
II.10 Arrays	II.10-1s.p1.pdf, II.10-4s.p1.pdf II.10-1s.p2.pdf, II.10-4s.p2.pdf	Echo.java PrimeNumbers.java Months.java
II.11 Recursion, complexity of algorithms	II.11-1s.p1.pdf, II.11-4s.p1.pdf II.11-1s.p2.pdf, II.11-4s.p2.pdf	Power1.java Hanoi.java
II.12 Search and sorting algorithms for arrays	II.12-1s.pdf, II.12-4s.pdf	search.java Quicksort.java merge.java Hash.java

Website: Contents 2

Part III: Object-oriented concepts and software development		
		Programs-Part-III.pdf
III.1 Basic concepts of object-oriented programming (1): abstract data types, objects, classes	III.1-1s.pdf, III.1-4s.pdf	Stack.java reversing.java reversing2.java
III.2 Object-oriented programming: basic examples	III.2-1s.pdf, III.2-4s.pdf	BracketStructur.java Time.java Schedule.java
III.3 Basic concepts of object-oriented programming (2): class variable and methods	III.3-1s.pdf, III.3-4s.pdf	TimeC.java ScheduleC.java
III.4 Component types	III.4-1s.pdf, III.4-4s.pdf	
III.5 Basic concepts of object-oriented programming (3): inheritance, polymorphism, dynamic binding	III.5-1s.pdf, III.5-4s.pdf	Time2.java
III.6 Basic concepts of object-oriented programming (4): generic classes	III.6-1s.pdf, III.6-4s.pdf	StackForChar.java BuildPairs.java BuildPairsBounds.java StackGen.java
III.7 Chained structures: lists	III.7-1s.pdf, III.7-4s.pdf	IntList.java List.java Stack1.java
III.8 Basic concepts of object-oriented programming (5): interfaces	III.8-1s.pdf, III.8-4s.pdf	ScheduleInt.java ScheduleAbstr.java ReversingNU.java KeyboardIApp.java Print.java
III.9 Exception handling	III.9-1s.pdf, III.9-4s.pdf	Exception.java TryCatch.java TryCatchAll.java Finally.java TryInTry.java KeyboardTry.java OpeningHours.java
III.10 Software development: requirements analysis and problem definition	III.10-1s.pdf, III.10-4s.pdf	
III.11 Object-oriented software architecture	III.11-1s.pdf, III.11-4s.pdf	
III.12 From concept to implementation	III.12-1s.pdf, III.12-4s.pdf	Maze.java MazeTest.java Mouse.java MouseMaze.java Easel.java SoftFrame.java
III.13 Trees: efficient searching and sorting algorithms	III.13-1s.pdf, III.13-4s.pdf	Tree.java Traverse.java TraverseTest.java
III.14 Applets	III.14-1s.pdf, III.14-4s.pdf	TempApplet.java TempApplet.html
III.15 Events	III.15-1s.pdf, III.15-4s.pdf	EyesApplet.html EyesApplet.java Eyes.java
III.16 Threads & processes	III.16-1s.pdf, III.16-4s.pdf	ThreadBasicTest.java ThreadSleep.java