



DAAD

16th Workshop “Software Engineering Education and Reverse Engineering”
Jahorina, 22 - 26 August 2016.

Transition from One- to Two-Semester(s) of Fundamentals of Programming

G. Banjac and D. Brdjanin

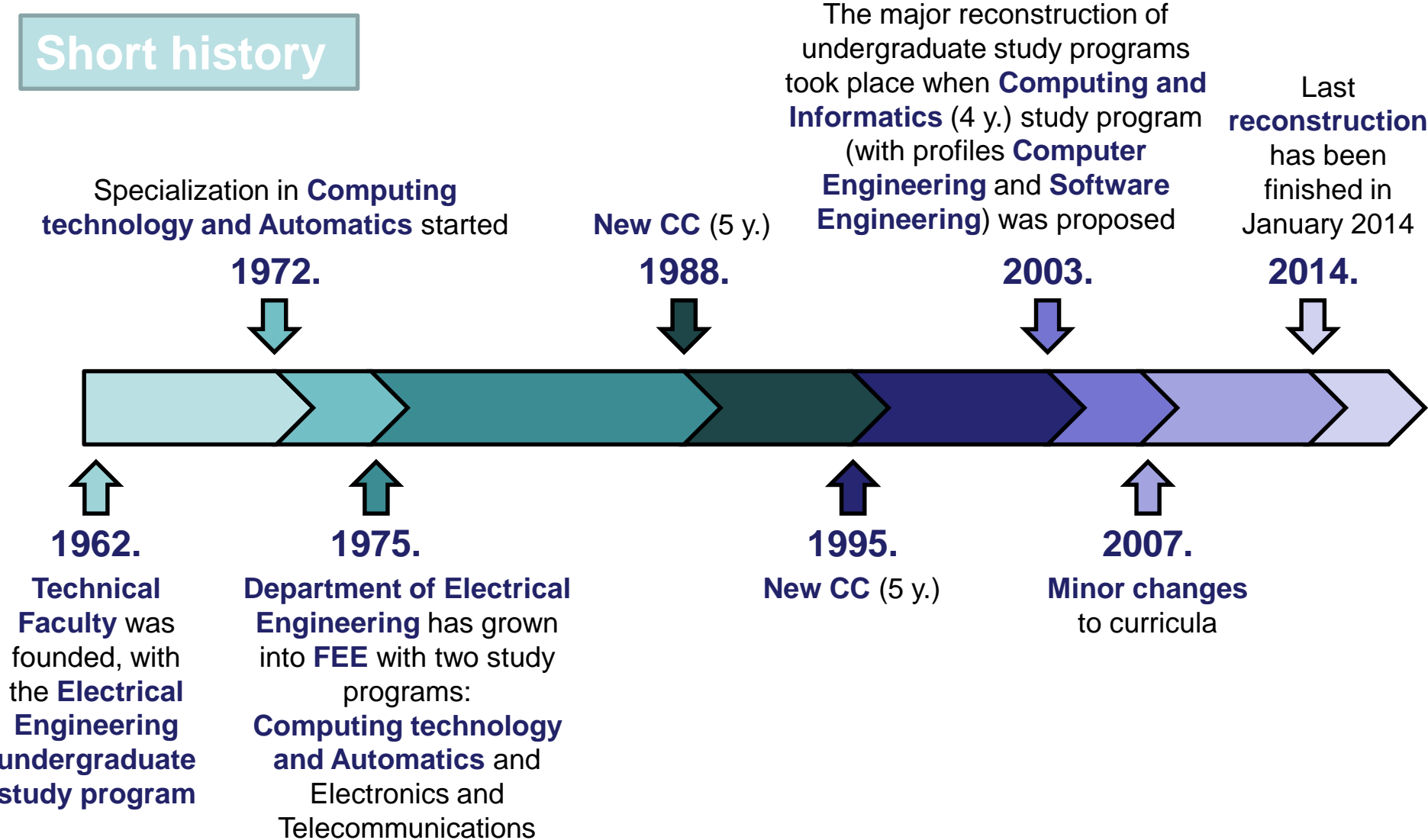
University of Banja Luka, Bosnia & Herzegovina

About this presentation

- Short history of computing curricula at FEE
- Syllabus of **Fundamentals of Programming** (*then and now*)
- Examination
- Statistics
- Students' feedback
- Conclusion

Undergraduate computing curricula at FEE (1/3)

Short history

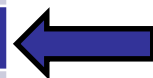


Undergraduate computing curricula at FEE (2/3)

First year courses overview (2003-2013)

- First year courses are common for all Study Programs

#	Course title	Sem.	ECTS	Hours	Type
1	Linear algebra	1	7	3+3+0	A
2	Fundamentals of electrotehnics	1	7	3+2+1	A
3	Physics	1	8	4+2+1	A
4	Fundamentals of computer technics	1	6	2+0+2	A
5	Sociology	1	2	2+0+0	B
6	Philosophy	1	2	2+0+0	B
7	Communication skills	1	2	2+0+0	B
8	Mathematical analysis I	2	8	3+3+0	A
9	Fundamentals of electrotehnics II	2	8	3+3+1	A
10	Programming fundamentals	2	7	3+2+1	A
11	Fundamentals of computer technics II	2	7	3+2+1	A



Legend: A – mandatory, B – elective

Undergraduate computing curricula at FEE (3/3)

First year courses overview (2014-)

- First year courses are common for all study programs

#	Course title	Sem.	ECTS	Hours	Type
1	Mathematics I	1	7	3+3+0	A
2	Fundamentals of electrotehnics I	1	7	3+2+1	A
3	Programming I	1	6	2+2+1	A
4	Fundamentals of computer technics	1	7	3+2+1	A
5	Sociology	1	3	2+0+0	B
6	Philosophy	1	3	2+0+0	B
7	Communication skills	1	3	2+0+0	B
8	Mathematics II	2	7	3+3+0	A
9	Fundamentals of electrotehnics II	2	8	3+3+1	A
10	Programming II	2	8	3+2+1	A
11	Physics	2	7	3+2+1	A

Legend: A – mandatory, B – elective

Course syllabus (1/3)

Programming fundamentals (7 ECTS)

- **Types and levels of the programming languages. Environment for program development: editors, compilers, linkers. Program execution. Programming: basic concepts and principles. Problem solving and algorithms.**
- **C programming language**
 - **Basic structure of C program, and elements of C language.**
 - **Variables, constants, and basic data types. Memory representation.**
 - **Operators and expressions.**
 - **Statements. Conditional statements. Loops, *break*, *continue*, and *goto* statements.**
 - **Functions, definitions and function prototypes. Function parameters and return values.**
 - **Complex data types: one-dimensional and multi-dimensional arrays. Structures and unions.**
 - **Pointers. Pointers as function parameters. Dynamic memory allocation.**
 - **Recursions and recursive functions.**
 - **I/O communication. Files and data streams. I/O function library.**

Course syllabus (2/3)

Programming I (6 ECTS)

- **Types and levels of the programming languages. Environment for program development: editors, compilers, linkers. Program execution. Basic concepts and principles of programming. Problem solving and algorithms.**
- **C programming language**
 - **Basic structure of C program.**
 - **Basic data types. Variables and constants. Memory representation.**
 - **Operators and expressions.**
 - **Statements. Conditional statements. Loops. *Break, continue, and goto* statements.**
 - **Complex data types: arrays, structures, unions.**
 - **Function declaration and function definition. Function parameters and return values.**
 - **Pointers. Pointers as function parameters.**

Course syllabus (3/3)

Programming II (8 ECTS)

- Recursions and recursive functions.
- Functions with variable number of arguments.
- Standard function libraries.
- Pointers to functions.
- I/O communication. Files and data streams. I/O function library.
- Dynamic memory allocation.
- Linear data structures: arrays, lists, stacks, queues.
- Non-linear data structures. Trees: binary trees, tree traversal. Graph basics.
- Basics of algorithm systems. Algorithm representation techniques. Basic sorting and searching algorithms.
- String manipulation. Large number manipulation.
- Standard algorithm library.
- Algorithm complexity.

Examination

Programming fundamentals

- Practical part:
 - two partial exams during semester (max. 25 points each), or
 - integral exam (max. 50 points)
- Final exam (max. 50 points)

Programming I

- Practical part:
 - two partial exams during semester (max. 30 points each) and online test for additional points (max. 10 points), or
 - integral exam (max. 60 points)
- Final exam (max. 40 points)

First generation of students didn't have test for additional points.

Programming II

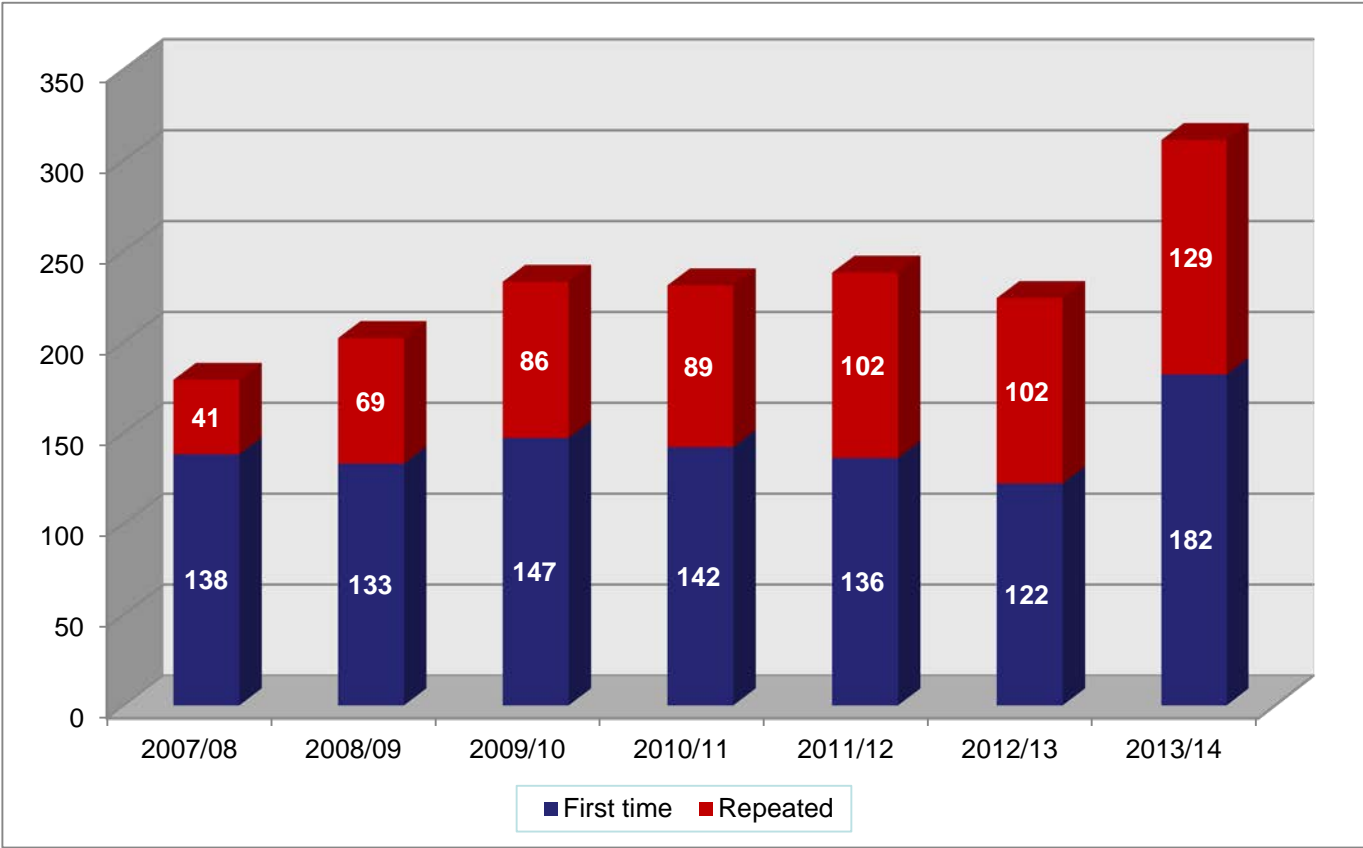
- Practical part:
 - two partial exams during semester (max. 25 points each) and work in a laboratory (max. 10 points), or
 - integral exam (max. 60 points)
- Final exam (max. 40 points)

Only 55% of students took this test for additional points.

Statistics (1/6)

Enrolled students

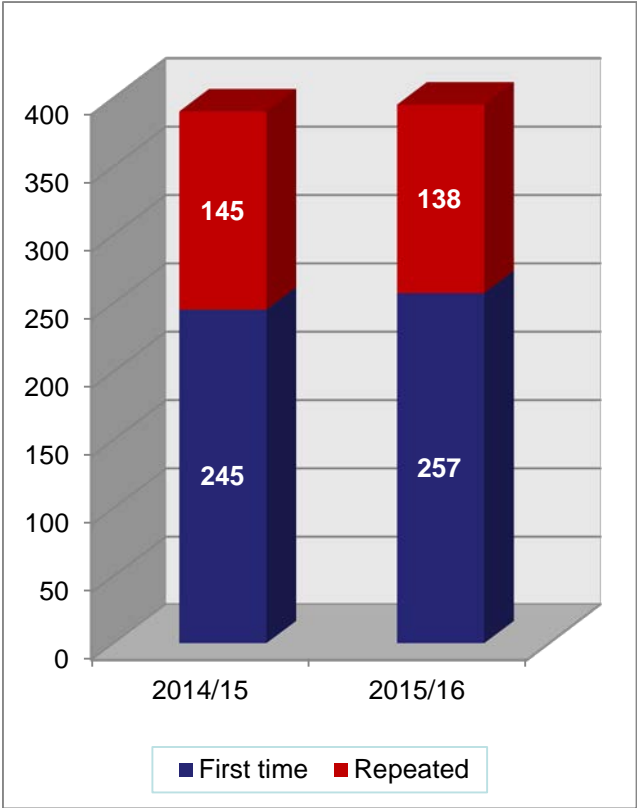
Programming fundamentals



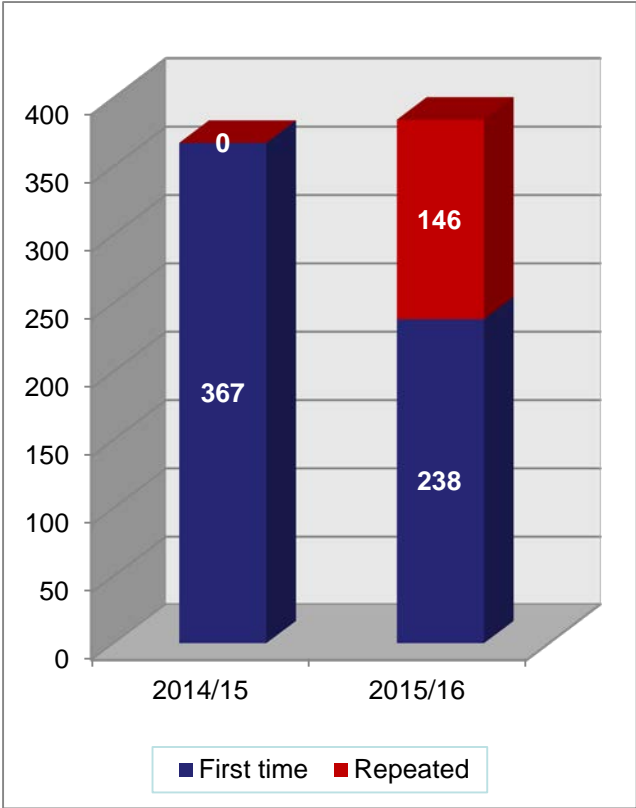
Statistics (2/6)

Enrolled students

Programming I



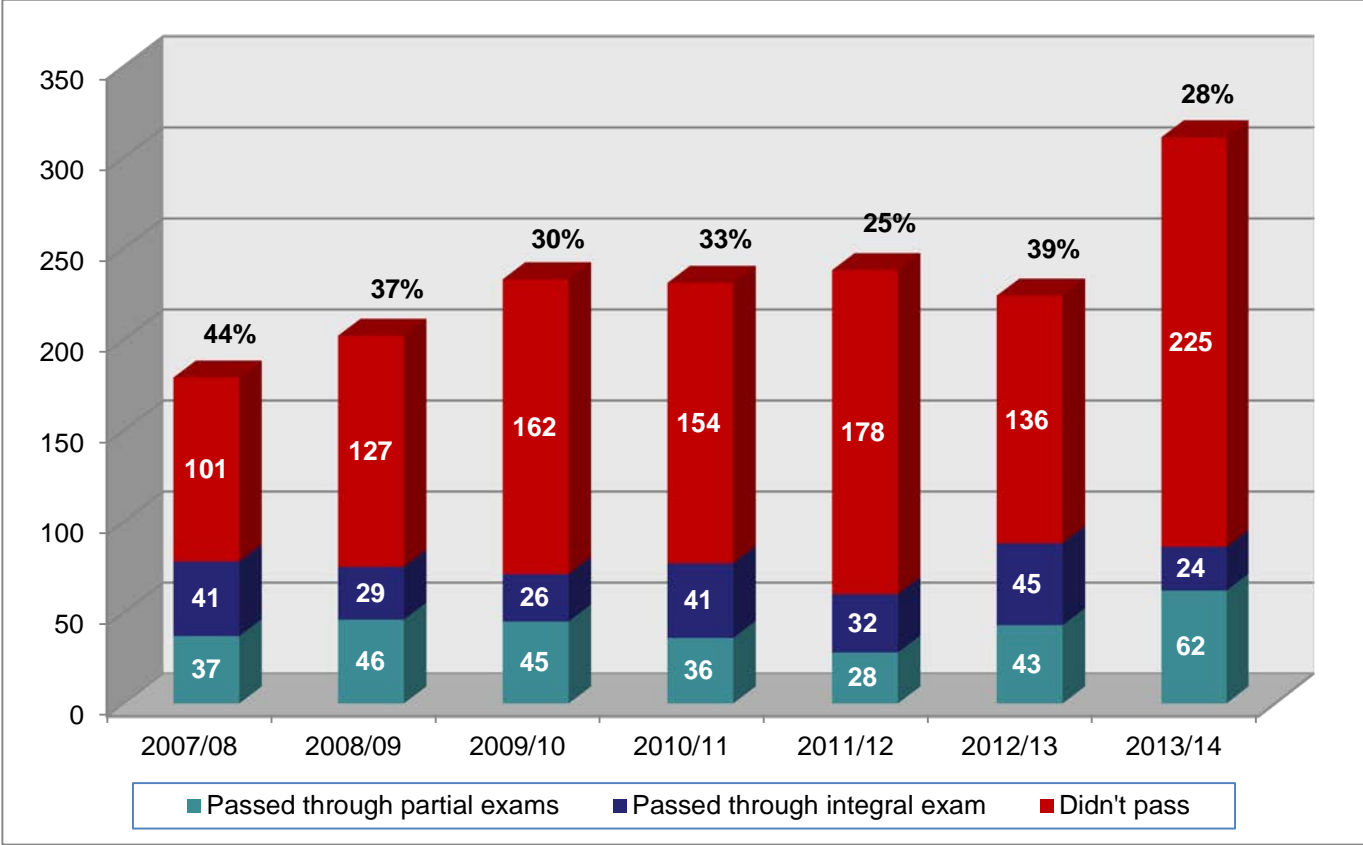
Programming II



Statistics (3/6)

Examination

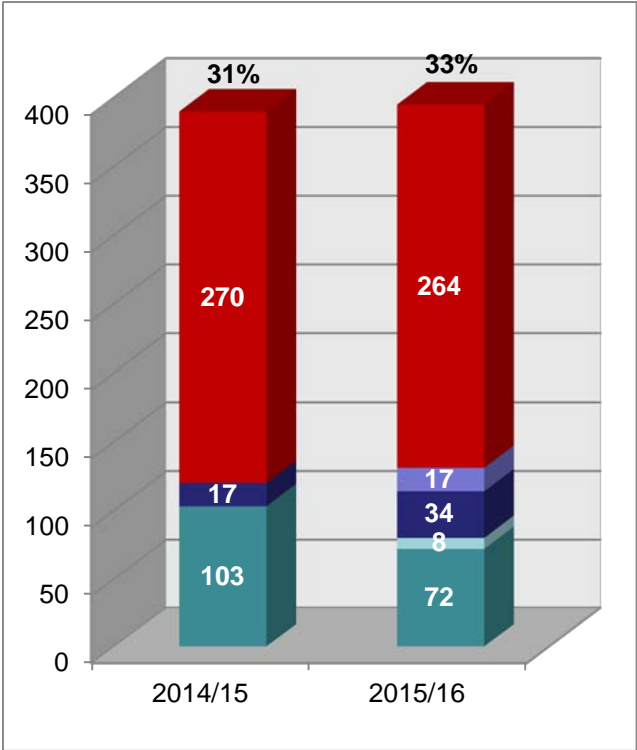
Programming fundamentals



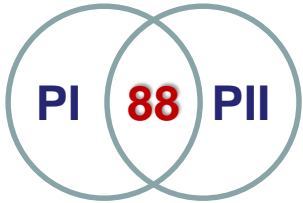
Statistics (4/6)

Examination

Programming I

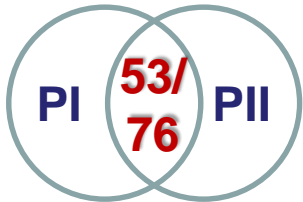


2014/15



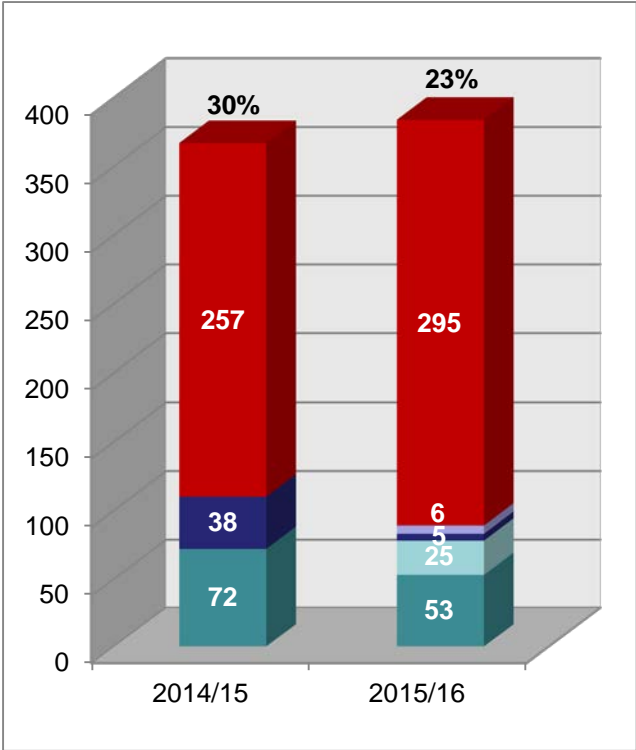
88 students passed both courses

2015/16*



53 students already passed both courses
At least 76 students are expected to pass both courses

Programming II



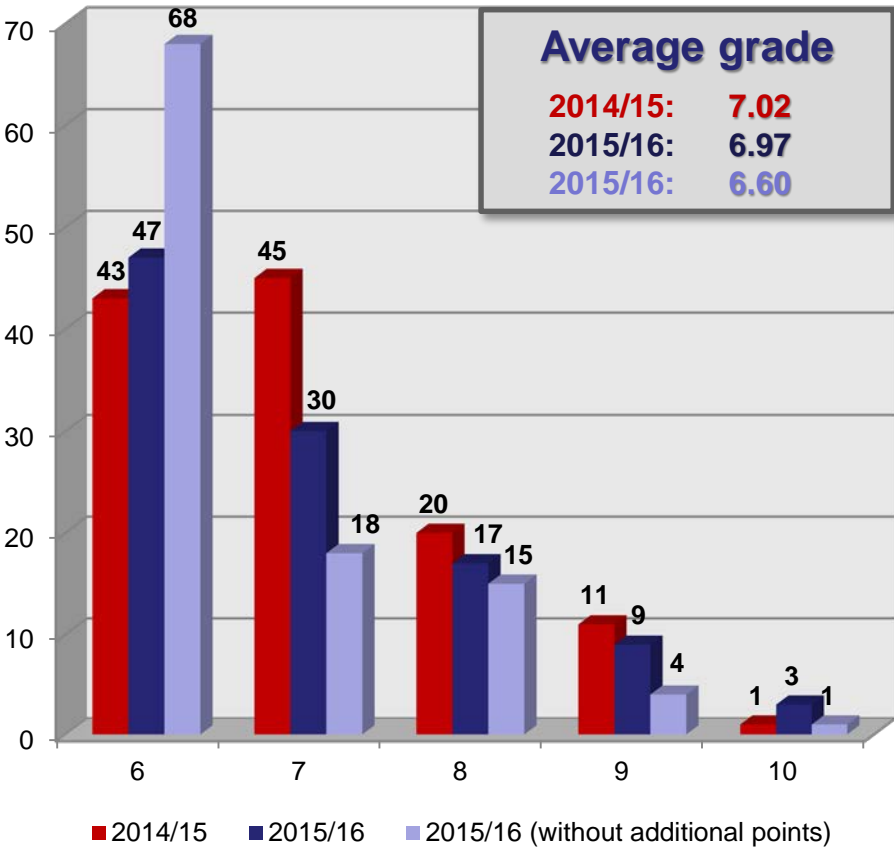
- Passed through partial exams
- Passed through integral exam
- Didn't pass
- Passed practical part through partial exams but didn't pass final exam yet
- Passed practical part through integral exam but didn't pass final exam yet

* Current data

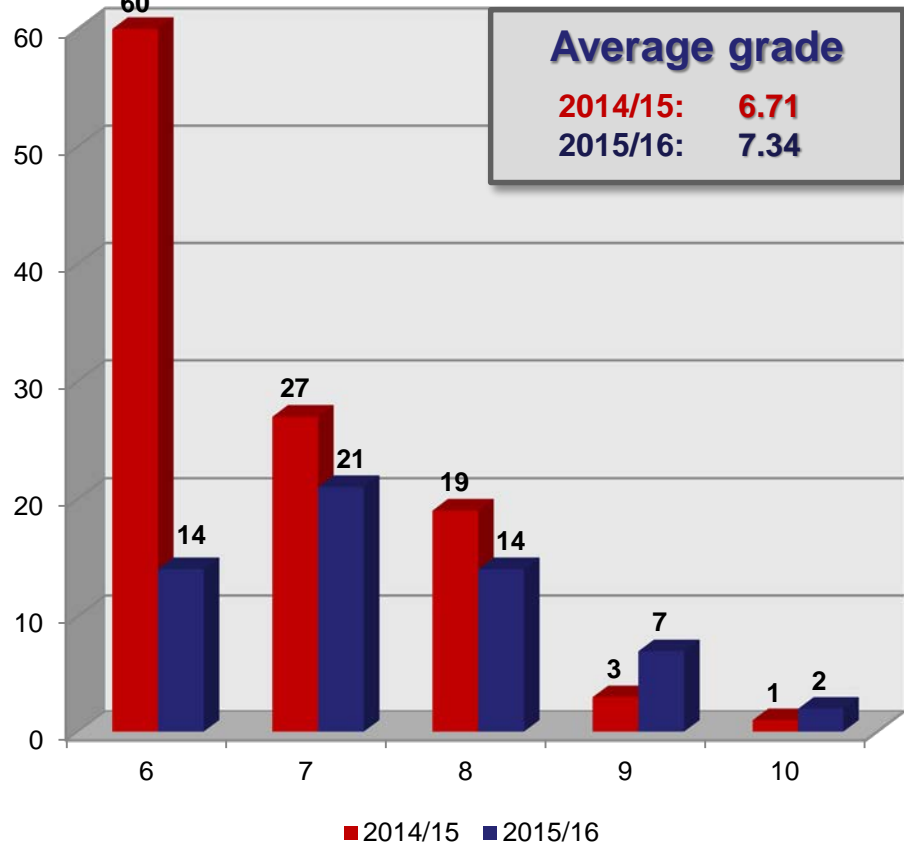
Statistics (5/6)

Grades

Programming I



Programming II



Statistics (6/6)

Comparison with other courses

#	Course title	Sem.	ECTS	Passed (2014/15)	Passed (2015/16)*
1	Mathematics I	1	7	81	80
2	Fundamentals of electrotehnics I	1	7	86	86
3	Programming I	1	6	120	106
4	Fundamentals of computer technics	1	7	131	91
5	Sociology	1	3	198	206
6	Philosophy	1	3		
7	Communication skills	1	3		
8	Mathematics II	2	7	20	22
9	Fundamentals of electrotehnics II	2	8	52	49
10	Programming II	2	8	110	58
11	Physics	2	7	46	14


* Current data

Students' feedback

- ... is, generally, **positive**, but with some suggestions on how to improve courses:
- *“Increase the number of hours”*
- *“Increase the number of hours in a laboratory”*
- *“More homework and tasks in a laboratory”*
- *“Examination should be on computers (instead of paper examination)”*
- *“Partial exams should be on computers”*

Conclusion

- **Last major reconstruction of undergraduate computing curricula at FEE** has been finished in January 2014
- Instead of one, now we have **two semesters of Fundamentals of Programming** in the first year curricula
- Now, **1st year students**:
 - **Learn more** advanced programming concepts (e.g. functions with variable number of arguments, ...) as well as basic concepts of data structures and algorithms
 - **Have more practical experience** / more work in a laboratory
- **Passing rate** for the new courses is **almost the same** as the passing rate for the old course

The logo for DAAD (German Academic Exchange Service) is located in the top left corner of the slide. It consists of the letters 'DAAD' in a bold, blue, sans-serif font.The background of the slide is a composite image. The left side shows a snowy mountain landscape with evergreen trees and a path. The right side shows a large, multi-story building with a snow-covered roof and illuminated windows, likely a resort or hotel in a mountainous region.

16th Workshop “Software Engineering Education and Reverse Engineering”
Jahorina, 22 - 26 August 2016.

G. Banjac and D. Brdjanin
University of Banja Luka, Bosnia & Herzegovina

Transition from One- to Two-Semester(s) of Fundamentals of Programming

Thank You!