

“Ss. Cyril and Methodius” University in Skopje
**FACULTY OF COMPUTER
SCIENCE AND ENGINEERING**

Last release of the Joint course in Software
Engineering in Skopje

Katerina Zdravkova & Bojana Koteska

University Ss. Cyril and Methodius

Faculty of Computer Science and Engineering, Skopje, N. Macedonia

e-mails: {katerina.zdravkova, bojana.koteska}@finki.ukim.mk

Outline of this presentation

- Timeline of JCSE in Skopje
- Course organization
- Student evaluation
- Evolution of student results
- Local software developing industry
- Acknowledgement

Timeline of JCSE in Skopje

- 2001: JCSE – Joint Course in Software Engineering established & new curricula at the former Institute of Informatics
- 2004/05: Parts of JCSE were included in the Theory of Programming course; teacher: Danilo Gligoroski, now professor at NTNU - Norwegian University of Science and Technology
- 2006/07: Complete JCSE delivered to 75 students from 4-year and 46 students from 3-year studies; teachers: Katerina Zdravkova & Boro Jakimovski
- 2011/12: The course was moved to FCSE – Faculty of Computer Science and Engineering
- Since 2014/15, Bojana Koteska replaced Boro Jakimovski

Course organization

- 1 semester: 15 weeks
- 12 weeks for teaching and 2 weeks for knowledge evaluation
- Weekly: 4 lecture hours for the JCSE presentations and 2 lecture hours for assignments

Distribution of presentations

RED	Katerina Zdravkova	BLUE	Bojana Koteska
What is software engineering?	What is software engineering?	What is software engineering?	
Software process models	Software process models	Agile development	Quality software criteria
Basic concepts	Cost estimation	Cost estimation	Modeling and product models
Results of analysis definition phase	Results of analysis definition phase	Basic concepts functional views	Basic concepts functional views
Basic concepts state oriented views	Basic concepts state oriented views	Basic concepts scenario oriented views	
Object oriented views	Object oriented views	Object oriented analysis	Object oriented analysis
Object oriented analysis g-h	Object oriented analysis g-h	Object oriented analysis g-h	
Formal specifications	Formal specifications	Formal specifications	
Overview of design phase	Overview of design phase	Object oriented design	
Implementation	Introduction to testing	Functional testing	Functional testing
Systematic testing	Systematic testing	Software metrics	Software metrics
Maintenance	Maintenance		

Alteration with JCSE

- The whole course is directed towards object-oriented software products
- Agile development lecture according to Roger Pressman's book *Software Engineering: A Practitioner's Approach*
- Extended presentations:
 - Modeling and product models
 - Basic concepts: State-oriented view
 - Results of the „Analysis and Definition“ phase

Practical assignments

- Based on real-life software products
- Recent topics:
 - Facebook Software Specification;
 - Web Based Integrated Development Environment;
 - Personal Investment Management System;
 - Electronic System for Issuing Licenses and Elaborates;
 - Electronic System for Recording and Exchange of Documents.

Student evaluation

- Two mid-term exams covering the theoretical part
 - Mid-term 1: Introduction and analysis
 - Mid-term 2: From formal specification to maintenance
 - The value of both exam is 100 points
- Four complementary tasks based on the same preliminary specifications
 - Task 1: Review of the given software specification
 - Task 2: Use case diagram + cost estimation
 - Task 3: Activity diagrams
 - Task 4: Functional testing - Classification tree method
 - The value of first two tasks is 40 points, of the next two tasks is 60 points

Sample mid-term 1 / part 1

Software Engineering, first midterm exam
13 June 2018

Multiple choice questions can have more than one correct answer. Incorrect answers lead to negative points.

Name and surname: _____ ID number: _____

1. Define software engineering according to IEEE, or according to ISO. **5 points**

2. Which measures can be undertaken when maintenance of one legacy software is difficult or almost impossible?
 - a) Software products have a relatively short life-time, thus no measures should be undertaken
 - b) Software products can be completely re-implemented
 - c) Software products can be improved by wrapping the existing code into new compatible interfaces
 - d) Software products are not measurable and hardware service requests can't be dealt with
 - e) Software products should be refactored under a continuing operation **3 points**

3. What does the software robustness mean? **3 points**
 - a) capability of the software to continue operating despite abnormalities in input, calculations, operating system
 - b) capability of the software to handle the effects of an erroneous environment
 - c) extensibility with respect to certain parameters

4. Which are the most important notation implemented in the: **8 points**
 - Functional view:
 - Data-oriented view:
 - State-oriented view:
 - Scenario-based view:

Sample mid-term 1 / part 2

5. How is software system incremental-iterative software development carried out? Illustrate it. **8 points**
6. Which are the four steps of the extreme programming software increment? Illustrate your answer. **8 points**
7. What is the prototype? In which product model is it explicitly delivered? **5 points**
8. If the amount of function points is known, can you estimate the execution of the software product measured in man-months and if it is possible, how can it be estimated? **5 points**
9. Which are the essential use case relations in UML and how are they performed? Illustrate your answer. **10 points**
10. What is the state automaton and how is it created? Illustrate it using Mealy's notation. **10 points**
11. Draw the use case diagram according to the story for the newspaper web site use case: A journalist can post articles. Reader can find and read articles. In order to perform any action, both journalist and reader must be authenticated. **8 points**
12. Estimation of functional points starts with the categorization of the requirements. Which are these categories, and how are they determined? **8 points**
13. The activity diagram should be used when: **9 points**
14. Illustrate the object life cycle in the collaboration diagram. **10 points**

Sample mid-term 2 / part 1

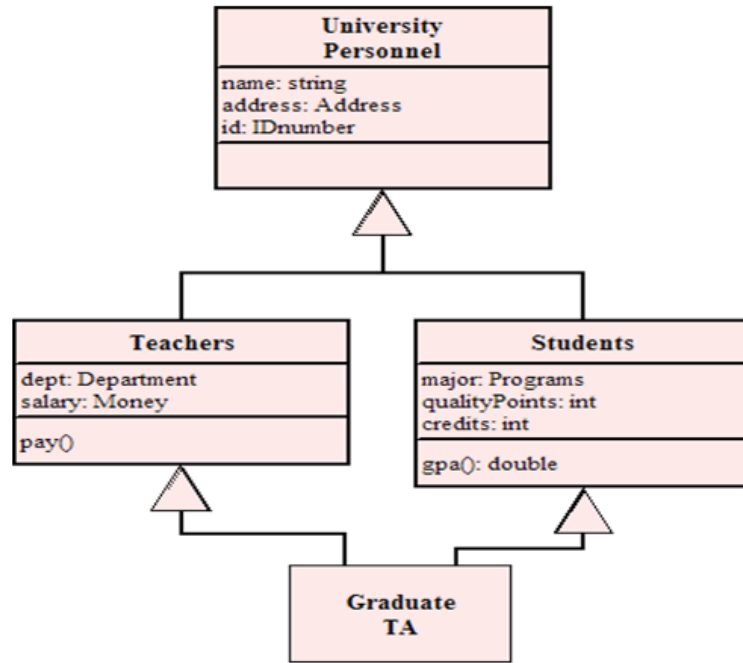
Software Engineering, second midterm exam
13 June 2018

Multiple choice questions can have more than one correct answer. Incorrect answers lead to negative points.

Name and surname: _____ ID number: _____

1. Describe the basic concepts of Z model based specification (states, operations, restrictions...). **5 points**
2. What is the basic approach in Hoare's specification and verification technique? Where is Hoare's logic applied and what does it describe? **10 points**
3. During requirement specification the classes of a prospective system are defined. Explain briefly the bottom-up and top-down approach for establishing the sub-systems that will possess a weak coupling between components and a strong cohesion between elements inside the components. **10 points**
4. Layered architectures usually consist of three major levels. Which are they? **6 points**
5. How the efficiency of OOD model can be improved? Circle the correct answers. **3 points**
 - a) Direct access to attributes instead of access operations
 - b) Additional calculations instead of attributes
 - c) Direct access to operations instead of access attributes
 - d) Additional attributes instead of calculations
 - e) Composite class consisting of several classes

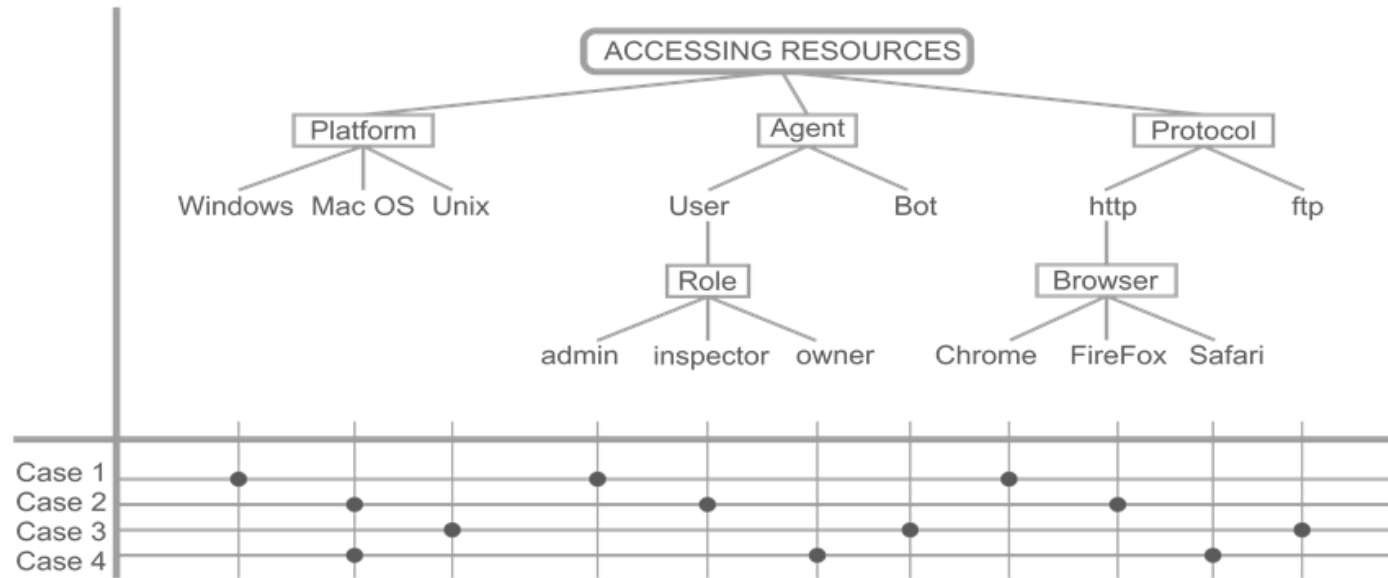
6. Check if the following OOD model is implementable in Java. If not, then transform it to be implementable. Explain your answer. **10 points**



7. Determine the meaning of the following code and improve its readability by implementing the good programming style guidelines and conventions. **15 points**

```
int(main)
{
int a[50],n,i,j,temp;
cin>>n;
for(i=0;i<n;++i)
cin>>a[i];
for(i=1;i<n;++i) {for(j=0;j<(n-i);++j)
if(a[j]>a[j+1])
{ temp=a[j]; a[j]=a[j+1]; a[j+1]=temp; }}
return 0;
```

8. Control-flow graphs are extended with def-, c-use and p-use. Where are they placed in the CFG, what is their meaning and what is the main goal of all-uses criterion? **8 points**
9. Describe all steps of Classification Tree Method. Then, identify each element of the following classification tree and calculate the minimal number of test cases. **10 points**



10. Calculate the cyclomatic complexity of the code represented in question 7 by counting its predicates (N.B. The complexity of while and for is equal). **5 points**
11. Describe each type of software maintenance. Then, identify the maintenance type/types in the following use case: Users could check the bank transactions on-line, but the used software often shows errors with transactions. Software errors were corrected according to user requirements. Explain your choices. **10 points**

Sample practical exam

Library Software

An association of libraries in Skopje needs new management software for all libraries in the city. The main functionalities of the system are: (1) borrowing and returning books to libraries; (2) obtaining a list of books by a particular author or topic, (3) requesting a list of currently rented books from a particular member, (4) requesting information about which user has rented a particular copy of a given book. Book borrowings are charged at a weekly rate (maximum 4 weeks), followed by a penalty fee that is calculated daily. Members can return book to any library, after which they will be charged an additional fee for sending the book by post to the appropriate library from which it was borrowed. The system should also enable standard additional administrative activities such as member and employee administration, book fund management, sending notifications, generating statistics.

Task 1. Use case diagram (30p)

Define a use case diagram with all user cases that you think are part of the user requirements.

Task 2. Project Functions (30p)

Define in details (template below) the following user cases:

Use case 1: Borrow a book

Use case 2: Return a book and payment

Use case 3: Search for a book by author / title

Description	
Priority:	
Precondition:	
Postcondition:	
Frequency of use:	
Users:	
Main scenario:	
Alternative scenario:	

Task 3. Activity diagram (20p)

Draw activity diagrams for all 3 user cases.

Task 4. Classification tree (20p)

Build a data tree classification tree for: article, book and library for Use case 2.

NOTE: The exam is passed with 60 points. By taking the exam you can have a maximum of 50% of the points assigned to the homework.

Evolution of student results – number of enrolled students per year

Year	English class	Macedonian class - prof. Zdravkova	Macedonian class - Other prof.	Total
2018/ 2019	52	208	421	681
2017/2018	44	220	410	674
2016/2017	26	138	284	448
2015/2016	23	115	213	351
2014/2015	25	208	176	409
2013/2014	0	79	74	153
2012/2013	0	122	135	257
2011/2012	0	89	103	192

Evolution of student results – number of students that passed the subject

Year	English class	Macedonian class - prof. Zdravkova	Macedonian class - Other prof.	Total	Percentage
2018/ 2019*	20	70	171	261	38%
2017/2018	32	135	287	454	67%
2016/2017	19	106	198	323	72%
2015/2016	22	98	164	284	81%
2014/2015	24	177	116	317	78%
2013/2014	0	55	49	104	68%
2012/2013	0	99	92	191	74%
2011/2012	0	75	73	148	77%

*First exam session only

Evolution of student results – average grade per year

Year	English class	Macedonian class - prof. Zdravkova	Macedonian class - Other prof.
2018/ 2019*	8.80	7.89	7.04
2017/2018	8.16	7.33	7.11
2016/2017	8.58	7.89	6.95
2015/2016	8.50	7.89	7.10
2014/2015	8.25	7.49	6.87
2013/2014	/	8.51	6.65
2012/2013	/	7.54	6.80
2011/2012	/	7.47	6.66

*First exam session only

Local software developing industry

- The course became recognized and appreciated by the local software developing industry
- Major companies have a high demand for quality software engineers
- They recommended to create a specialized direction for software engineering and information systems (SEIS)
- This direction will no longer have the introductory software engineering course, so JCSE goes to history
- In 2018/19, 288 enrolled SEIS in Macedonian, and 80 in English

Acknowledgement

- We thank all the colleagues from the DAAD consortium who have contributed to the development of the Joint Course in Software Engineering, its slides and lecture notes.
- We also thank our colleagues who have unselfishly contributed to the local delivery of the course.
- And finally, we thank you all for your attention.