Faculty of Mathematics, Belgrade
Computer Science Department
Software Development 2 (Razvoj softvera 2)
Fall 2009

Class Time:

Lecture	Monday 14-17	at Studentski trg 16, dlab
Instructor supervised assigned work from the Paper / Topic List below	Wednesday 18-...	at Simina 2

Office Hours :

Monday	**Wednesday**	**Thursday**
14-20 , DLAB	14-18, 718	16-18, 706
	18-… (by appointment), Simina 2	18-20, BEAM

Software Development 2
Schedule

1. UML (quiz)

2. Design Patterns (quiz)

3. Platform independent software development (quiz)

4. ?Human Computer Interaction (Web Page Development, Human-centered software evaluation / Human-centred software development / Graphical user-interface design / Graphical user-interface programming)

5. ?Software testing

6. ? Software verification

Textbooks:

Course Main Texts

Booch,G., Rumbaugh,J., Jacobson,I.,.	*The Unifiied Modeling Language -User Guide, Addison-Wesley* *UML Vodič za korisnike*, CET, 2005.
Gama,E., Helm,R., Johnson,R., Vlissides,J.,	*Design Patterns-Element of Reusable Object Oriented Software, Addison-Wesley* *Gotova Rešenja*, CET, 2002.

Additional Texts

1. Rumbaugh,J., Jacobson,I., Booch,G.,	*Unified Modeling Language Reference Manual*, 2nd Edition, Addison-Wesley, 2004..
2. Alexander, Christopher, Ishikawa, Sara,	*A Pattern Language*, Oxford University Press, New York, 1977.

For further reading:

1. C. Horstmann and G. Cornell: Core JAVA, Volume I Fundamentals, Sun Microsystems, Inc. 2005.

2. Bernd Bruegge and Allen H. Dutoit, *Object-Oriented Software Engineering Using UML, Patterns and Java*, Second Edition. Prentice Hall, 2004.

3. Software Engineering Body Of Knowledge

4. ACM UML Conference Proceedings (copies from instructor)

5. Will David Mitchell *Debugging Java: Troubleshooting for Programmers,* Osborne Mc Graw Hill, 2002

6. Technical journals:

a. Information Processing & Management, Pergamon Press

b. Pattern Recognition, The Journal of the Pattern Recognition Society

c. Dr Dobb's Journal, www.ddj.com

d. Java Report, www.javareport.com

7. Šampionski biseri (Google cache: www.sampioni.com)

Grading Policy:

Attendance, Class Participation and Effort, Topic Presentation or Software Presentation (Review and discussion)	15 pt
Homework, Quiz	15 pt
Final Paper/Project	15 pt

(i) The course topics will be examined through readings, discussion, hands-on experience using various software systems, and through participation in evaluation of different software development algorithms on various test collections.
(ii) There will be periodic assignments and a final paper/project.
(iii) Final paper will be a technical or research paper on a software development issue. Final Project will include implementation (design pattern), UML diagrams and unit tests. Final paper or project require a preparation that includes finding materials outside of base reading and during the assigned work periods.

List of Papers/Topics

Topic, **Paper and Demonstration Materials**	Presenter	**Presentation Date**	Presentation / Discussion (in progress)
1. Past, Present, and Future of User Interface Software Tools www.google.org			
2. Google's API-AJAX Language API How to Design a Good API and Why it Matters			
3. eXtreme Programming	Igor Ivanović	21.12.09	
4. Rich Internet Applications – some trends	Vladimir Milošević,	14.12.09	

	Marko Kaznovac		
5. Fundamental Practises for Secure Software Development	Miloš Kralj, Miloš Đurić	21.12.09	7+8 7
6. Java Midlet (javax.microedition.midlet)			
7. Is Java Dead?	Boris Radovanović	14.12.09	7
8. Test Case Generation, UML, and Eclipse			
9. The Practice of Formal Methods in Safety Critical System			
10. Software Patterns – GoF design pattern, semantic analysis pattern			
11. Anti-patterns: C, PHP, Java, C#			
12. Formal Method: VDM			
13. Components: COM, ActiveX, JavaBeans			
14. Components: CORBA and SOAP			
15. SCRUM	Stjepanović Miloš	21.12.09	
16. Employment: Open source programmer	Aleksandra Aleksić, Marko Dangubić	21.12.09	7 7
17. Mobile Web Widgets - development	Marko Marković, Petar Radović	14.12.09	7 7
18. How to estimate Software Costs			
19. Extreme programming: Web project and Customer-centric development			
20. Software development: Ruby On Rails			
21. Cloud computing and software development	Ilić Andrijana, Marko Stojićević	21.12.09	7 7+8
22. Software development for bioinformatics	Nikola Ranković	14.12.09	
23. Facebook Application Development (FBML, FQL, FBJS)	Đorđe Kalanj, Danijel Marjanović	21.12.09	7+8 7

24. Software Protection and Application Security (Protection against automated disassemblers)	Marko Trajkov	21.12.09	7
25. NoSQL database	Nikola Milenković	14.12.09	7
26. Softver Development for Web Application - Some Trends	Ana Mijalković, Marija Jeličić	21.12.09	7 7
27. Code Clone Detection			
28. Copy and Paste programming practice, Cargo Cult programming			
29. Software Metrics - Object-Oriented Design Metrics, Improvement of OO Design Quality			
30. Google AJAX search API			
31. What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software	Aleksandar Smiljković	14.12.09	7
32. Software development for statistics	Jelena Tasić, Aleksandra Zafirovska	21.12.09	7 7
33. Widgets and Rich Internet Applications			
34. Wolfram Alpha API, computational knowledge engine			
35. Future Internet, Web 3.0	Nikola Samardžija, Ivan Milosavljević	14.12.09	7 7

Paper Presentation (Review/Critique) Guidelines

Each critique should be no less than ten pages long. The purpose of a presentation is not to summarize the paper; rather you should choose one or two points about the work that you found interesting.

Examples of questions that you might address are:

- What problem does this paper solve, and what are the strengths and limitations of its approach?
- Is the evaluation fair? Does it achieve to support the stated goals of the paper?

- Does the method described seem mature enough to use in real applications? Why or why not? What applications seem particularly amenable to this approach?
- What good ideas does the problem formulation, the solution, the approach or the research method contain that could be applied elsewhere?
- What would be good follow-on projects and why?
- Are the paper's underlying assumptions valid?
- Which important issues in the field does this paper illuminate and how?
- Did the paper provide a clear enough and detailed enough description of the proposed methods for you to be able to implement them? If not, where is additional clarification or detail needed?

Your work should be .PDF or .HTML file. It should list the title of the paper and its authors at the top, along with your name.

Avoid **unsupported** value judgments, like ``I liked..." or ``I disagreed with..." If you make judgments of this sort, explain why you liked or disagreed with the point you describe.

Be sure to distinguish comments about the writing of the paper from comment about the technical content of the work.

Example: Maljković Mirjana, Perić Nikola

Paper Presentation Guidelines

Length : class period (talk of up to 15 minutes to be followed by an up to 10 minutes discussion mediated by the presenter)

Medium : HTML slides, PDF slides.

Note about how the preparedness for other students presentations affects the grade

(i) All listed papers must be read by every student in class.

(ii) The discussion following the paper presentation demonstrates that the student has read the paper.

(iii) Student's involvement and competence in the discussion will directly affect the "Attendance, Class Participation and Effort"'s 33% of student's total grade for the entire course.

Final Paper/Project Suggestions

I Topics for a programming project include :

- real time software, data processing software, data mining software,…

- student projects from previous undergraduate course Software Development

Tasks for a programming project include:

1. Software design

2. Risks and liabilities of computer-based systems

3. Software requirements and specifications

4. Software validation

5. Software evolution

6. Software project management

Software construction should include: Object-oriented programming / Event-driven programming / Using APIs / Fundamental data structures

HCI part of project should discuss: (**Human-centered software evaluation / Human-centred software development)**

II. Design and Implementation of Project Application:

Real time: traffic control,…
Embedded systems: GPS,…
Data processing: telephone billing, pensions
Information systems: web sites, digital libraries
Sensors: weather data
Offices: text processing,…
Scientific: simulations, weather forecasting
…

1. Selection of a document collection for project.
2. Marking up or unifying mark ups of collection documents.
3. Environment preparation for document collection.
4. Creating system specification files.
5. Processing document collection using Your system and specs from 4.
6. Executing different test examples using Your system.
7. Reporting on phases 1.-6. and evaluating the system built

III. Final Paper (Techical or Research Projects)

These projects will investigate an open issue in software development and prepare a research paper outlining existing approaches, their strengths and weaknesses, and offer a new approach to be investigated. Each paper should be no less than eleven pages long (including introduction, bibliography).

1. Software reusability

2. Object oriented development – some trends

3. Software testing

4. Formal method

5. Refactoring

6. Multi-media retrieval (speech, video, web pages), question answering, automatic summarization methods, data mining, bioinformatics

7. “Code Complete” - some lectures