

DESCRIPTION OF A STUDY COURSE – SYLLABUS

| | | | | | |
|---|---|-----------------|--|---------------------|---|
| Title of a course | Object-oriented Technologies I | | | | |
| Head of course | Vlatka Davidović, Senior Lecturer | | | | |
| Study programme | Professional undergraduate study Information Science | | | | |
| Status of a course | Obligatory | | | | |
| Year of study | 2 | Semester | IV | ECTS credits | 6 |
| Teaching plan (L + E + S+ Pr) | 2 + 3 + 0 + 0 | | | | |
| Goals of a course | | | | | |
| Acquiring knowledge of the basic concepts of object-oriented programming and application design. | | | | | |
| Conditions for enrolling course | | | | | |
| No conditions | | | | | |
| Learning outcomes on a level of a study programme which includes course | | | | | |
| Outcome 2: Apply business information system design methods. Outcome 4: Develop an application solution for the Internet and desktop environment. Outcome 12: Apply engineering methods and principles in information science. Outcome 15: Independently present professional content in written and spoken form in Croatian and English. | | | | | |
| Expected learning outcomes on a level of a course | | | | | |
| <ol style="list-style-type: none"> 1. Describe the process of planning and building software from the object-oriented approach aspect. 2. Apply basic concepts of object-oriented programming 3. Design an application using the object-oriented approach 4. Implement the application in object programming language based on the assigned task 5. Create an application with data entry and display, using the object-oriented approach | | | | | |
| Content of a course | | | | | |
| <p>Software products and their development. Procedural vs. object-oriented programming. Introduction to object-oriented analysis. Concepts of object-oriented approach. Basic phases of object-oriented design. Process of collecting demands. Classes and objects as basic units of abstraction. Classes and objects model. Links between classes. Polymorphism. Encapsulation. Attribute class design, operation/method and relation. Pattern design. Inheritance.</p> <p>Incremental and iterative processes in the development of applications. Static and dynamic modelling in object-oriented analysis. Identification and analysis of dynamic models. Use of CASE tools. Selection of tools for modelling. UML (Uniform Modelling Language). Oracle Jdeveloper 9i and 10g tool. Object-oriented languages: CV++, Pearl, Java, Script languages. Application on the client server and the network. Security and control. Standards for application expansion.</p> | | | | | |
| Teaching modes | <input checked="" type="checkbox"/> lectures <input type="checkbox"/> auditory exercises <input checked="" type="checkbox"/> seminars and workshops <input type="checkbox"/> distance learning <input type="checkbox"/> field classes | | <input checked="" type="checkbox"/> individual assignments <input type="checkbox"/> multimedia and network <input type="checkbox"/> laboratory <input type="checkbox"/> supervisor's work <input type="checkbox"/> other _____ | | |
| Comments | | | | | |
| Students' obligations | | | | | |
| If a student passes the full exam, he / she is obliged to do the project assignment in the prescribed range beforehand. | | | | | |
| Grading, evaluation and monitoring of students' work continuously during lectures and exams | | | | | |
| Grading is based upon evaluation of course's learning outcomes' adoption. Grading is performed continuously during lectures and/or during exam, in compliance with the provisions of Regulation on the assessment of | | | | | |

students.

Continuous check-up:

| Outcomes | Written exam | Practical tasks | Project | Threshold | Max |
|--------------------|--------------|-----------------|---------|-----------|-------|
| Outcome 1 | 10% | | | 5% | 10% |
| Outcome 2 | 20% | | | 10% | 20% |
| Outcome 3 | 10% | | | 5% | 10% |
| Outcome 4 | | 30% | | 15% | 30% |
| Outcome 5 | | | 30% | 15% | 30% |
| Percentage of ECTS | 2,4 | 1,8 | 1,8 | - | - |
| Total | 40% | 30% | 30% | 50 % | 100 % |

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Exam term:

| Outcomes | Written exam | Project | Oral exam | Max |
|--------------------|--------------|---------|-----------|-------|
| Outcome 1 | 10% | | | 10% |
| Outcome 2 | 10% | | 10% | 20% |
| Outcome 3 | 10% | | | 10% |
| Outcome 4 | 20% | | 10% | 30% |
| Outcome 5 | | 20% | 10% | 30% |
| Percentage of ECTS | 3 | 1,2 | 1,8 | - |
| Total | 50% | 20% | 30% | 100 % |

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Grading:

A student has passed the exam if he has acquired at least 50% of anticipated credits of a specific learning outcome.

If a student has passed learning outcomes of all courses, the accomplished credits (percentages) of all passed learning outcomes are being added, while the final grade is defined upon following table:

| Range of credits (percentages) | Numerical grade | ECTS grade |
|--------------------------------|------------------|------------|
| 90,00 – 100,00 | Excellent (5) | A |
| 75,00 – 89,99 | Very good (4) | B |
| 60,00 – 74,99 | Good (3) | C |
| 50,00 – 59,99 | Sufficient (2) | D |
| 0,00 – 49,99 | Insufficient (1) | F |

Obligatory literature

1. Davidović, V., Pogarčić, I., Objektno orijentirane tehnologije I vježbe, Veleučilište u Rijeci, 2011.
2. Eckel, B.: Thinking in Java 4th Edition, Prentice Hall, 2006

Additional literature

1. Bloch, J.: Effective Java, Addison-Wesley, 2007.
2. John Hunt, J: The Unified Process for Practitioners: Object Oriented Design, UML and Java, Springer; 1 edition (June 16, 2000)
3. Freeman, E., Freeman, E., Sierra, K., Bates, B.: Head First Design Patterns, O'Reilly, 2004.
4. Sierra, K., Bates, B.: Head First Java, O'Reilly, 2005

