



COURSE DESCRIPTION

1. Information about the Study Program

| | |
|-----------------------------------|--|
| 1.1 Higher Learning Institution | “Alexandru Ioan Cuza” University of Iași |
| 1.2 Faculty | Faculty of Computer Science |
| 1.3 Department | Computer Science |
| 1.4 Field of Study | Computer Science |
| 1.5 Type of Degree | Bachelor |
| 1.6 Study Program / Qualification | |

2. Information about the Course

| | | | | | | | |
|----------------------------|-------------------------|--------------|---|---------------------|---|-----------------|----|
| 2.1 Name | Programming Engineering | | | | | | |
| 2.2 Course Coordinator | Iftene Adrian | | | | | | |
| 2.3 Laboratory Coordinator | Iftene Adrian | | | | | | |
| 2.4 Year of study | 2 | 2.5 Semester | 2 | 2.6 Evaluation Type | M | 2.7 Course Type | OB |

* OB – Compulsory / OP – Optional

3. Total Estimated Time (hours per semester and study activities)

| | | | | | |
|---|----|----------------------|----|----------------|-----|
| 3.1 Number of hours per week | 4 | Of which: 3.2 course | 2 | 3.3 laboratory | 2 |
| 3.4 Total hours planned | 56 | Of which: 3.5 course | 28 | 3.6 laboratory | 28 |
| Time distribution | | | | | ore |
| Individual study using manuals, course notes, etc. | | | | | 24 |
| Individual study in the library, using specialized electronic tools, field work | | | | | 24 |
| Preparation of seminars/laboratories, homework, papers, portfolios, essays | | | | | 23 |
| Tutorials | | | | | |
| Exams | | | | | 12 |
| Other activities | | | | | 1 |
| 3.7 Total hours of individual study | | | | | 71 |
| 3.8 Total hour per semester | | | | | 140 |
| 3.9 Number of credits | | | | | 6 |

Obs. $T = C + S$

4. Preconditions (if applicable)

| | |
|------------------------------|---|
| 4.1 Curriculum preconditions | Algorithms and programming. Object oriented programming. |
| 4.2 Competence preconditions | Programming using high level languages. Development and maintenance of applications. Designing and using databases. |

5. Conditions (if applicable)

| | |
|----------------------|--|
| 5.1 For courses | Overhead projector, internet connection, blackboard |
| 5.2 For laboratories | Internet connection, blackboard, computers with appropriate OO language development environment installed (Java, C#, Python, PHP, C++, etc.) |



6. Specific competencies

| | |
|----------------------------------|--|
| Professional Competencies | C1. Programming in high level languages C2. Developping and maintaining IT applications C3. Designing and using databases |
| Transversal competencies | CT1. Working effectively within an interdisciplinary group and the development of interpersonal communication skills |

7. Objectives

| | |
|--------------------------------|---|
| 7.1 General Objective | Building a professional image with regards to the process of developing IT products. Students will learn advanced methods and techniques for creating complex, high quality programs while adapting to clients requirements regarding function, costs and deadline. |
| 7.2 Specific objectives | On successful completion of this course, students will be able to: <ul style="list-style-type: none">• Explain how to use object oriented programming principles in large development projects• Describe the steps required to successfully implement a project for a client and describe the steps necessary for testing the project• Utilize Java technologies to implement and test functional components of a given project• Analyze customer requirements, analyze software errors that occur during the development of the project, analyze the manual tests on a system• Calculation of time, staffing and money needed to develop the project requested by the client |

8. Content

| 8.1 | Course | Teaching methods | Notes |
|-----|--|---|---------------------------------------|
| 1. | Course content. Bibliography. Motivation, definitions, famous errors, statistics. Development models (Waterfall, Spiral) | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [1], [2], [3], [4], [5], [6] |
| 2. | Development models (RUP, V-Model, XP, Agile, Lean, Scrum, AMDD, TDD). Requirement analysis. Actor, Use Case | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [1], [7], [8] |
| 3. | Modeling, Modeling languages, UML, Use Case diagrams, class diagrams | Slide presentation. Course notes and tutorials available in electronic format. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [1], [9] |
| 4. | UML Diagrams (Sequence diagrams, Collaboration diagrams, State, Activites, Deployment, Packages). SCRUM | Slide presentation. Course notes and tutorials available in electronic format. Problematization | 2 hours, [4] |



| | | | |
|-----|--|---|---------------------|
| | Methodology | | |
| 5. | Reverse Engineering. Presentation of tools needed for project development: Github, Trello, GoogleDocs. | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [4], [10] |
| 6. | GRASP. SOLID principles | Slide presentation. Course notes and tutorials available in electronic format. Problematization and work on examples. | 2 hours, [2], [4] |
| 7. | Design Patterns: Definitions, components, taxonomy. Creational Patterns. | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [3], [5] |
| 8. | Revision of previous courses. Requirement analysis, modeling, design patterns | Case studies. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [3], [5] |
| 9. | Design Patterns: creational patterns, structural patterns | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [3], [5] |
| 10. | Design Patterns: behavioural patterns | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [3], [5] |
| 11. | Design Patterns: concurrency, testing, distributed. Software testing. | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [3], [5] |
| 12. | Software testing: introduction, methods, processes. Manual vs. automatic testing. | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours, [11], [12] |
| 13. | Program quality, metrics, copyright. | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours |
| 14. | Content for the advanced course: Swebok, SOA, QoS, BPMN, AOP, Refactoring | Slide presentation. Course notes and tutorials available in electronic format. | 2 hours |

Bibliography**Main references:**

- [1] Ian Sommerville: Software Engineering, Addison Wesley, 2001
- [2] Craig Larman: Applying UML and Patterns, Addison Wesley, 2002
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley, 1998

Additional references:

- [4] ArgoUML: <http://argouml-downloads.tigris.org/>
- [5] Preview of Patterns in Java Volume 2: http://www.mindspring.com/~mgrand/pattern_synopses2.htm
- [6] Failure rate: http://www.it-cortex.com/Stat_Failure_Rate.htm
- [7] RUP in the dialogue with Scrum: <http://www.ibm.com/developerworks/rational/library/feb05/krebs>
- [8] Agile Manifesto: <http://agilemanifesto.org/>
- [9] Requirements analysis process: <http://www.outsource2india.com/software/RequirementAnalysis.asp>
- [10] Chikofsky, E.J. and Cross, J.: Reverse Engineering and Design Discovery: A Taxonomy, January 1990
- [11] Junit Test Example: <http://www.cs.unc.edu/~weiss/COMP401/s08-27-JUnitTestExample.doc>
- [12] Bug Life Cycle: <http://www.buzzle.com/editorials/4-6-2005-68177.asp>,
<http://qastation.wordpress.com/2008/06/13/process-for-bug-life-cycle/>

| 8.2 | Laboratory | Teaching methods | Notes |
|-----|---|---|---------------|
| 1. | Revision of OO programming techniques (inheritance, polymorphism, etc.) | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [13] |



| | | | |
|-----|--|--|------------------------------|
| 2. | Requirements analysis, UML – Class diagrams | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [1], [2], [4], [9] |
| 3. | Requirements analysis, UML – Use Case diagrams, Class diagrams | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [4], [9] |
| 4. | Using SCRUM methodology in problem solving | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [4] |
| 5. | Reverse Engineering. Presentation of tools needed for project development: Github, Trello, GoogleDocs. | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [3], [4], [5] |
| 6. | Design Patterns – Crational patterns, JUnit Testing | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [2], [4], [5], [11] |
| 7. | Project work: risk study, requirement analysis | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [4], [9] |
| 8. | Revision | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [3], [5] |
| 9. | Project Work: Requirements Engineering, UML Diagrams | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [4], [9] |
| 10. | Project work: SCRUM sprints (Implementation, testing, integration) | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [3], [4], [5] |
| 11. | Project work: SCRUM sprints (Implementation, testing, integration) | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [3], [4], [5] |
| 12. | Project work: SCRUM sprints (Implementation, testing, integration) | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard. | 2 hours, [3], [4], [5] |
| 13. | Project work: SCRUM sprints (Implementation, testing, integration) | Case study. Discussions. Solutions for a given problem will be drawn on the blackboard.. | 2 hours, [3], [4], [5] |
| 14. | Project work: Evaluation | Video presentation. Case study. Discussion. | 2 hours, [11] |

Bibliography:**Main references:**

- [1] Ian Sommerville: Software Engineering, Addison Wesley, 2001
- [2] Craig Larman: Applying UML and Patterns, Addison Wesley, 2002
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vissides: Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley, 1998

Additional references:

- [4] ArgoUML: <http://argouml-downloads.tigris.org/>
- [5] Preview of Patterns in Java Volume 2: http://www.mindspring.com/~mgrand/pattern_synopses2.htm
- [6] Failure rate: http://www.it-cortex.com/Stat_Failure_Rate.htm
- [7] RUP in the dialogue with Scrum: <http://www.ibm.com/developerworks/rational/library/feb05/krebs>
- [8] Agile Manifesto: <http://agilemanifesto.org/>
- [9] Requirements analysis process: <http://www.outsource2india.com/software/RequirementAnalysis.asp>
- [10] Chikofsky, E.J. and Cross, J.: Reverse Engineering and Design Discovery: A Taxonomy, January 1990
- [11] Junit Test Example: <http://www.cs.unc.edu/~weiss/COMP401/s08-27-JUnitTestExample.doc>
- [12] Bug Life Cycle: <http://www.buzzle.com/editorials/4-6-2005-68177.asp>,



<http://qastation.wordpress.com/2008/06/13/process-for-bug-life-cycle/>

[13] Lucanu D.: Principiile programării orientate-obiect,

<http://thor.info.uaic.ro/~dlucanu/cursuri/poo/resurse/principiiPOO.pps>

9. Adaptation of course content according to the expectations of the representatives of communities, professional associations and main employers from the field of the teaching

IT companies from Iași and elsewhere in the country use software engineering techniques in the development of large scale projects. The steps necessary for developing large applications (requirements engineering, modeling, implementation, integration, testing, deployment) are widely discussed throughout the course, and students learn the basics and actually go through these stages. During the summer students participate in internships in companies where they are involved in real projects where they apply the theoretical concepts learned during the course in real projects. Following discussions with major employers, this course is updated from year to year, adapting to the current needs of the IT market and beyond.

10. Evaluation

| Activity type | 10.1 Evaluation criteria | 10.2 Evaluation methods | 10.3 Percentage in final grade(%) |
|-----------------|--|--|--------------------------------------|
| 10.4 Course | Passing grade: 40% of the maximum grade obtainable at the written test | Written test + bonuses for course activity | 50 % |
| 10.5 Laboratory | Passing grade: 50% of the maximum grade obtainable for the laboratory work (at least 50 % of the points from the activities during the first 6 laboratories and at least 50 % of the maximum project points) | Attendance + solutions for the activities given during the first 6 weeks + project work + bonuses for extra activities related to the course material. | 50 % (of which 50 % project work) |

10.6 Minimum performance standards

Students will be able to identify the main requirements after discussion with the client and be able to provide the solution which he expects. Implementation requires the building of main classes and their attributes without implementing basic methods. Testing of the project will be done in terms of manual testing. Laboratory Minimum score: 50% of the maximum score that can be obtained in the laboratory (minimum 50% of laboratory subjects and minimum 50% project) Minimum score exam: 40% of the maximum score that can be achieved on the exam For students who meet the promotion criteria, the final mark is determined by dividing the score obtained to the maximum points without a bonus, and the result is multiplied by 10. Students participating in the exam will receive a grade, otherwise they will be considered absent. If one of the criteria for promotion is not met, the student will get a grade less than or equal to 4.

Date
01.02.2018

Course coordinators
Conf. Dr. Iftene Adrian

Laboratory coordinators
Conf. Dr. Iftene Adrian

Date of approval in department

Department head
Prof. Dr. Lucanu Dorel