Università degli Studi della Basilicata
Scuola di Ingegneria

COURSE: Advanced Programming Techniques			
ACADEMIC YEAR: 2017/2018			
TYPE OF EDUCATIONAL ACTIVITY: Basic			
TEACHER: GIANVITO SUMMA			
e-mail: gianvito.summa@gmail.com		web: http://informatica.unibas.it/moodle/	
phone:		mobile (optional):	
Language: ITALIAN			
ECTS: 6	n. of hours: 48	Campus: Potenza School: Engineer Program: Master Degree in Engineering in Computer Science	Semester: Annual

EDUCATIONAL GOALS AND EXPECTED LEARNING OUTCOMES

This course treats programming techniques in object-oriented programming language (Java).

The educational goals mainly consist in providing skills and abilities to students for designing and building systems in medium/high complexity.

By the end of the course, students will know:

- How to deal with requirements analysis and designing of a system having a medium-high complexity;
- How to use the main programming techniques.

Main topics are listed below:

- Design Patterns;
- Inversion of Control;
- Thread e concurrent programming;
- Clonation and serialization;
- Refactoring techniques;
- Agile development process and AUP;

With respect to Dublin descriptionr goals:

- **Knowledege and understanding**:
  student has to know how to deal with a problem having a medium-high complexity; he has to be able to factor the problem into sub-problems having a lower complexity in order that relevant programming techniques and design patterns could be applied.
- **Applying knowledge and undertanding**:
  student has to know how to design and build an application that solves a problem having a medium-high complexity.
- **Making judgements:**
  student has to prove to be albe to evaluate autonomously all the steps that are needed for defining and building an application, starting from requirements and also by adopting the best useful programming techniques in order to build an efficient, easy to mantain and scalable system.
- **Communication:**
  student must be able to explain how he intends to design and develop the system by using a scientific-technical language.
- **Lifilelong learning skills:**
- *Student must be able to continuosly update his personal skills and abilities by periodically reading books, magazines, papers and web sites in the advanced object-oriented programming field.*

PRE-REQUIREMENTS

It is required that students know the following concepts provided by “Object-Oriented Programming” course, expee:

- Inheritance and polymorphism.

SYLLABUS

Design Patterns:

- Introduction; History; Patterns and Framework; Patterns and Java API; Data and algorithms: Conclusions.

Inversion of Control:

- Introduction; Dependency Injectio; How to…; AOP; Conclusions.

Refactoring:

- Introduction; Techniques; Examples.

Development process:

- Introduction; RUP; XP; AUP.

Clonation and serialization.

Threading e synch.

TEACHING METHODS

Theoretical lessons

EVALUATION METHODS

Written examination, Practical test, Oral examination.

The examination's goal consists in verifying the skills and knowledges acquired by students, as depicted before.

The examination is composed by three parts that take place in three different days. In the order:

- A **written examination** regarding the main course's topics (designing and techniques). Students that do not pass this test (the minimum score is 18/30 points) cannot access to the following test parts. Students cannot use anything else during the test (no PC, no smartphone, no calculators….);
- A **practical test** to be done in the lab, consisting in building an application having a medium-high complexity. This test aims at evaluate if students have acquired skills in requirements analysis and designing. To pass the test students have to reach at least 18/30 points. Students may consult the user guides, Java documentation and slides of the course (all this stuff is available in lab). The test is composed by two exercises: the first one (24 points) is related to the requirements analysis and designing skills and the second one (6 points) is related to building GUI to prove the correctness of the underlying model;
- An **oral examination**. To pass this test students have to reach at least 18/30 points.

Students that pass both the **intermediate tests** (minimum score of 18/30 points) will have an extra bonus of 2 points upon the practical test. Please note that this bonus may be used only within the third (III) examination session, i.e. September. The intermediate tests consist in quizzes, designing problems and practical exercises.

TEXTBOOKS AND ON-LINE EDUCATIONAL MATERIAL

The slides of the course are available online at http://informatica.unibas.it/moodle/

(click "*Tecniche di Avanzate di Programmazione*", authentication required).

Books:

- Cooper -- The Design Patterns Java Companion;
- Eckel -- Thinking in Patterns;
- Martin Fowler -- Refactoring -- Improving the Design of Existing Code -- 1999 – Addison-Wesley

For further details please consult the link provided before and go to the "Riferimenti Utili" section that contains interesting resources (books and web sites) for each topic of the course.

INTERACTION WITH STUDENTS

In the first lesson, the teacher will explain all useful information about the course and how to consult the online stuff (at the Moodle Learning platform). Office hours: On Monday/Wednesday, from 6pm to 7.30pm, at "Docenti a contratto" office room (DiMIE Department, III floor).

Students may also contact teacher by email.

EXAMINATION SESSIONS (FORECAST)[1]

First intermediate verification: 21th Feb 2018

Second intermediate verification: 27th Jun 2018

[1] Subject to possible changes: check the web site of the Teacher or the Department/School for updates.

Sessions:
I: 4/6th Jul 2018
II: 14/16th Jul 2018
III: 12/14th Sep 2018
IV: 12/14th Dec 2018
V: Feb 2019
VI: May 2019

SEMINARS BY EXTERNAL EXPERTS YES □ NO □

FURTHER INFORMATION