Zał. nr 4 do ZW 64/2012

FACULTY OF INFORMATICS AND MANAGEMENT / DEPARTMENT OF INFORMATICS

SUBJECT CARD

Name in Polish Projektowanie oprogramowania
Name in English Software System Development
Main field of study (if applicable): Informatics
Specialization (if applicable): ……..
Level and form of studies: 1st~~/ 2nd~~* level, full-time~~/ part-time~~*
Kind of subject: obligatory~~/ optional / university-wide~~*
Subject code INZ0270Wp
Group of courses YES~~/ NO~~*

	Lecture	Classes	Laboratory	Project	Seminar
Number of hours of organized classes in University (ZZU)	30			30	
Number of hours of total student workload (CNPS)	60			90	
Form of crediting	Examination~~/ crediting with grade~~*	Examination / crediting with grade*	Examination / crediting with grade*	~~Examination /~~ crediting with grade*	Examination / crediting with grade*
For group of courses mark (X) final course	X				
Number of ECTS points	2			3	
including number of ECTS points for practical (P) classes	0			2	
including number of ECTS points for direct teacher-student contact (BK) classes	1,2			1,8	

*delete as applicable

PREREQUISITES RELATING TO KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. A student defines features of object-oriented paradigm
2. A student knows and describes basic software life-cycle models
3. A student applies a programming language for solving algorithmic problems; he/she knows how to implement graphical user interface in a selected tool and programming language.

\

SUBJECT OBJECTIVES

C1 To familiarize students with the tasks carried out during the basic software life-cycle processes according to ISO/IEC 12207; To teach students good design practices (including design patterns) and to prepare them for team project course (for programmatic projects)
C2 To develop students' skills that will enable them to develop technical project documentation in the UML with the appropriate tools

SUBJECT EDUCATIONAL EFFECTS

relating to knowledge:

PEK_W01 A student lists and describes basic software life-cycle processes

PEK_W02 A student recognizes and describes elements used in basic UML diagrams applied for software modeling and specification

PEK_W03 A student lists, classifies and selects appropriate design patters in the context of existing problem

PEK_W04 A student lists some tools used for software modeling, specification, implementation and testing

relating to skills:

PEK_U01A student uses UML for requirement specification and business modeling

PEK_U02 A student designs a graphical user interface

PEK_U03 A student adapts basic architectural patterns and design patterns to solved engineering problem

PEK_U04 A student implements selected functional requirements in a high level programming language

PEK_U05 A student plans and specifies tests, and carry out functional testing process

relating to social competences:

PEK_K01 A student has awareness of the importance of non-technical aspects of activity of computer engineer, understands the need to provide high quality information systems, taking into account the needs of different user groups.

PROGRAMME CONTENT

Form of classes - lecture		**Number of hours**
Lec 1	Introduction. Basic definitions.	2
Lec 2	Stakeholder requirements definition process. Techniques of requirements elicitation. Domain model and vocabulary.	2
Lec 3	System requirements analysis process. Requirements classification. Requirements diagrams.	2
Lec 4	System architectural design process. Selected architectural perspectives.	2
Lec 5	Requirements analysis process. How to write effective use-cases.	2
Lec 6	User interface prototype. GUI design guideline.	2
Lec 7	Software architecture design process. Architectural patterns.	2
Lec 8	Data base design. Design of use-case realizations.	2
Lec 9	Architectural mechanisms. Application of state machines to object's life cycle design.	2
Lec 10	Design patterns – examples and discussion.	2
Lec 11	Detail design process.	2
Lec 12	Software construction process.	2
Lec 13	Software testing. Testing techniques.	2
Lec 14	Integration, qualification testing, and installation processes.	2
Lec 15	Repetition.	2

	Total hours	30

Form of classes - project		**Number of hours**
Proj 1	Introduction to the project.	2
Proj 2	Elaboration of application concept (vision, glossary, business rules, domain model)	6
Proj 3	Requirement specification (functional, and non-functional requirements, use-case model, prototype of GUI)	8
Proj 4	Design (architectural design, use-case realization, class details)	6
Proj 5	Construction and testing	8
	Total hours	30

TEACHING TOOLS USED

N1. Informative lecture supported by multimedia presentations
N2. Examples of documents
N3. Case tool, IDE used for programming and testing
N4. E-learning system used for materials publication, gathering and assessing students projects

EVALUATION OF SUBJECT EDUCATIONAL EFFECTS ACHIEVEMENT

Evaluation (F – forming (during semester), P – concluding (at semester end)	Educational effect number	Way of evaluating educational effect achievement
F1 – elaboration of application concept	PEK_U01	Checking of completeness, intra and inter consistency. Up to 15% of maximal number of points for the whole project
F2 – requirements specification	PEK_U02, PEK_K01	Checking of intra-consistency, completeness, correctness, GUI guidelines. Up to 25% of the maximal number of points for the whole project
F3 - design	PEK_U01, PEK_U03	Checking for inter-consistency (with preceding phases, and between different diagrams), completeness. Up to 30% of the maximal number of points for the whole project
F4 – construction and tests	PEK_U04, PEK_U05	As above. Up to 30% of the maximal number of points.
F5 – final grade from project	PEK_U01, …, PEK_U05	The grade calculated basing on the formula: < 40% punktów → ndst. <40%, 50%) → 3.0 <50%, 60%) → 3.5 <60%, 70%) → 4.0 <70%, 80%) → 4.5 <80%, 90%) → 5.0 >90% → 5.5
F6 – final grade from lecture	PEK_W01, PEK_W02, PEK_W03,	Exam – multiple choice test. The grade is calculated basing on the formula: <50%, 60%) → 3.0 <60%, 70%) → 3.5 <70%, 80%) → 4.0

	PEK_W04	<80%, 90%) → 4.5 >90% → 5.0
P – final grade from the course	All	The final grade is calculated as (0.4 * F5 + 0.6 * F6)

PRIMARY AND SECONDARY LITERATURE

PRIMARY LITERATURE:

[1] Bruegge Bernd. Object-oriented software engineering: using UML, Patterns, and Java. Pearson/Prentice Hall, cop. 2004.
[2] Pfleeger Shari Lawrence. Software engineering: theory and practice. Pearson/Prentice Hall, 2006.

SECONDARY LITERATURE:

[1] Sommerville Ian, Software engineering, Addison-Wesley, 2007.
[2] Materials prepared by the lecturer

SUBJECT SUPERVISOR (NAME AND SURNAME, E-MAIL ADDRESS)

Bogumiła Hnatkowska, Bogumila.Hnatkowska@pwr.wroc.pl

MATRIX OF CORRELATION BETWEEN EDUCATIONAL EFFECTS FOR SUBJECT

System Software Development

AND EDUCATIONAL EFFECTS FOR MAIN FIELD OF STUDY

Informatics

AND SPECIALIZATION ……..

Subject educational effect	Correlation between subject educational effect and educational effects defined for main field of study and specialization (if applicable)**	Subject objectives***	Programme content***	Teaching tool number***
PEK_W01	K1INF_W07	C1	Lec2, …, Lec5, Lec7, Lec8, Lec11, …, Lec15	N1, N4
PEK_W02	K1INF_W07	C2	Lec2, Lec3, Lec4, Lec8, Lec9	N1, N2, N4
PEK_W03	K1INF_W05	C1	Wy7, Wy10	N1, N4
PEK_W04	K1INF_W07	C2	Lec1, Lec6, Lec13	N1, N4
PEK_U01	K1INF_U03, K1INF_U13	C2	Proj2, Proj3, Proj4	N2, N3, N4
PEK_U02	K1INF_U03, K1INF_U14	C1	Proj3	N2, N3, N4
PEK_U03	K1INF_U03, K1INF_U04	C1, C2	Proj4	N3, N4
PEK_U04	K1INF_U02, K1INF_U04	C1	Proj5	N3, N4
PEK_U05	K1INF_U04	C1	Proj5	N3, N4

** - enter symbols for main-field-of-study/specialization educational effects

*** - from table above