Zał. nr 4 do ZW 64/2012

FACULTY OF INFORMATICS AND MANAGEMENT / DEPARTMENT………………

SUBJECT CARD

Name in Polish: Techniki translacji programów
Name in English: Program Translation Techniques
Main field of study (if applicable): Informatics
Specialization (if applicable): ……………………..
Level and form of studies: 1st level, full-time
Kind of subject: optional
Subject code: INZ0291Wcl
Group of courses YES

	Lecture	Classes	Laboratory	Project	Seminar
Number of hours of organized classes in University (ZZU)	30	15	15		
Number of hours of total student workload (CNPS)	60	30	60		
Form of crediting	crediting with grade	crediting with grade	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	1	2		
including number of ECTS points for practical (P) classes			2		
including number of ECTS points for direct teacher-student contact (BK) classes	1,2	0,6	1,2		

*delete as applicable

PREREQUISITES RELATING TO KNOWLEDGE, SKILLS AND OTHER COMPETENCES

1. Basic skills in programming in C++ or Java
2. Intermediate knowledge in computer architecture and organization
3. Basic knowledge in algorithms and data structures

\

SUBJECT OBJECTIVES

C1 The student should be able to define the syntax of programming languages that can be efficiently analyzed and translated into executable code

C2 After completion of the course the students should be able to develop syntax and semantic analyzers of typical programming languages as well as to create compiler modules responsible for target code generation

C3 Students should be able to efficiently used existing compilers and select high level appropriate programming languages for efficiency-critical applications

SUBJECT EDUCATIONAL EFFECTS

relating to knowledge:

PEK_W01 Knows formal methods for syntax and semantic definition of programming languages
PEK_W02 Knows formal foundations of context-free grammars application in syntax analysis
PEK_W03 Understands principles of attribute grammars and its application to contextual language properties

relating to skills:

PEK_U01 Is able to design the syntax of a programming language using context free grammars
PEK_U02 Is able to create the syntax and semantic analyzer using compiler creation tools
PEK_U03 Is able to create a code generator for the simplified target virtual machine
PEK_U04 Is able to predict the structure and efficiency of the machine code created by a typical compiler

PROGRAMME CONTENT

Form of classes - lecture		Number of hours
Lec 1	Introduction, elements of programming language definition: syntax, semantics and pragmatics; architecture of a programming language compiler, examples of syntax analysis applications.	2
Lec 2	Formal approach to syntax definition; Chomsky grammars, the notion of derivation, derivation trees. Notation for syntax definition: BNF, EBNF, syntax diagrams	2
Lec 3	Lexical analysis; lexical class definition by regular grammars, extraction of the lexical level from the syntax definition	2
Lec 4	lexical analysis: conversion from regular grammars to finite nondeterministic automata; efficient implementation of lexical analyzer based of deterministic finite automaton	2
Lec 5	Methods of descending syntax analysis, LL(1) grammars, recursive descent approach, building syntax analyzers from syntax diagrams	2
Lec 6	Methods of descending syntax analysis: predictive LL(1) analyzers.	2
Lec 7	Bottom-up syntax analysis methods, LARL analyzers	2
Lec 8	Building control tables fro LALR analyzers using SLR method	2
Lec 9	Using off the shelf components in implementation of syntax analyzers	2
Lec 10	Symbol table organization, symbol table contents, efficient search in symbol tables	2
Lec 11	Contextual language property analysis; attribute grammars, examples of application; efficient implementation of analyzers based on attribute grammars	2
Lec 12	Introduction to code generation, abstract machine architectures for intermediate code generation, defining translation rules with attribute grammars	2
Lec 13	Translation of typical syntax elements: control flow structures,	2

	infix expressions, structural data types, Memory organization in run-time. Subroutine calls translation, Call stack organization, activation record contents	
Lec 14	Code optimization	2
Lec 15	Final test	2
	Total hours	30

Form of classes - class		Number of hours
Cl 1	defining syntax of simple languages	2
Cl 2	lexical analysis	2
Cl 3	top-down syntax analysis	2
Cl 4	bottom-up analysis	2
Cl 5	application of attribute grammars to contextual language properties definition	2
Cl 6	building translation schemes for selected syntax constructs in programming languages	2
Cl 7	Translation of expressions	1
Cl 8	Practical skills test	2
	Total hours	15

Form of classes - laboratory		Number of hours
Lab1	Lexical analyzer implementation	2
Lab2	Syntax analyzer implementation	2
Lab3	Checking contextual properties using semantic rules	2
Lab4	Error handling module implementation	2
Lab5	Code generator implementation for compound expressions	2
Lab6	Code generation for control statements and subroutine calls	2
Lab7	Tests and documentation preparation	2
Lab8	Final evaluation and grading	1
	Total hours	15

Form of classes - project		Number of hours
Proj1		
Proj2		
Proj3		
Proj4		
…		
	Total hours	

Form of classes - seminar		Number of hours
Sem1		

Sem2		
Sem3		
…		
	Total hours	

TEACHING TOOLS USED

N1. standard lecture presentation supported by slideshow presentation
N2. open source and free software components supporting compiler construction (lex, yacc, Bison, ANTLR
N3. development environments for C++ and Java (Eclipse, Netbeans, MSVC)

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		Lecture - written test verifying the formal knowledge related to formal programming languages definition methods and their application to building practical algorithms
F2		Laboratory - the evaluation of the final product (simple compiler) created in results of series of lab assignments. The following elements will be evaluated: correctness of the formal definition of the language, ease of practical use, correctness of automatically created target code, code clarity, appropriate application of common software development practices, completeness of final documentation
F3		Classes - the partial classes grade will be given basing on the result of written test carried out at the last class modified with student's activity (presentation of home-solved exercises) (one successful presentation: +0.5 of grade, failure to present declared exercise: -0.5 of grade)
C - The final grade for the whole group of courses will be the equally weighted average of grades F1, F2, F3 obtained from lecture, classes and laboratory. In order to pass all three grades must be positive.		

PRIMARY AND SECONDARY LITERATURE

PRIMARY LITERATURE:

[1] Aho A.V. et al, Compilers, principles and tools, Addison-Wesley, 2007
[2] Andrew W. Appel and Jens Palsberg. Modern compiler implementation in Java, Cambridge University Press, 2002
[3] Hopcroft J., Ullman J.D. Motwani R., Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Addison-Wesley, 2006

SECONDARY LITERATURE:

[1] Wilhelm R., Seidl H., Compiler Design. Virtual Machines, Springer, 2010
[2] Kaspersky K., Code Optimization. Effective Memory Usage, A-LIST LCC, 2003
[3] Kaspersky K., Hacker Disassembling Uncovered, A-LIST LCC, 2003

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

Dr inż. Jerzy Sas

MATRIX OF CORRELATION BETWEEN EDUCATIONAL EFFECTS FOR SUBJECT
Program Translation Techniques
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 (knowledge)	K1INF_W06	C1	Lec1, Lec2, Lec11, Cl1, Cl2	N1
PEK_W02	K1INF_W06	C1	Lec3-Lec8, Cl3, Cl4	N1
PEK_W03	K1INF_W06	C1	Lec11, Cl5	N1
PEK_U01 (skills)	K1INF_U01, K1INF_W06	C2	Lec1, Lec2, Lec6-Lec8, Cl1, Lab1, Lab2	N1, N2
PEK_U02	K1INF_U01	C2	Lec6-Lec9, Cl2, Cl3, Cl4, Lab3, Lab4	N1, N2, N3
PEK_U03	K1INF_U01	C2, C3	Lec12-Lec14, Cl6, Cl7, Lab5, Lab6	N1, N2, N3
PEK_U04	K1INF_U01, K1INF_W06	C2. C3	Lec10, Lec14, Cl6, Cl7, Cl8, Lab7, Lab8	N1, N2, N3

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