

(pieczęć wydziału)

**KARTA PRZEDMIOTU**

Z1-PU7

WYDANIE N1

Strona 1 z 3

<b>1. Nazwa przedmiotu:</b> INTRODUCTION TO COMPILERS		<b>2. Kod przedmiotu:</b> ItC		
<b>3. Karta przedmiotu ważna od roku akademickiego:</b> 2012/2013				
<b>4. Forma kształcenia:</b> studia drugiego stopnia				
<b>5. Forma studiów:</b> studia stacjonarne				
<b>6. Kierunek studiów:</b> MAKROKIERUNEK; WYDZIAŁ AEII				
<b>7. Profil studiów:</b> ogólnoakademicki				
<b>8. Specjalność:</b> INFORMATYKA				
<b>9. Semestr:</b> 2				
<b>10. Jednostka prowadząca przedmiot:</b> Instytut Informatyki, RAu2				
<b>11. Prowadzący przedmiot:</b> dr inż. Przemysław Szmal				
<b>12. Przynależność do grupy przedmiotów:</b> przedmioty specjalnościowe				
<b>13. Status przedmiotu:</b> obowiązkowy				
<b>14. Język prowadzenia zajęć:</b> angielski				
<b>15. Przedmioty wprowadzające oraz wymagania wstępne:</b> Introductory course: Computer programming. The student is assumed to be able to use the Visual C programming environment				
<b>16. Cel przedmiotu:</b> The goal of the course is to present selected problems connected to programming language description and compiler construction. The student masters algorithms and methods for lexical and syntactic analysis, as well as ways of extending them for translation purposes. Topics suitable for construction of simple translators met in programmer's practice – command interpreters, macro-generators, linkers and so on – are covered.				
<b>17. Efekty kształcenia:<sup>1</sup></b>				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1	Wiedza z zakresu sposobu opisu języków programowania	Kolokwium, egzamin	wykład, ćwiczenia	K_W03
2	Wiedza z zakresu budowy i działania kompilatorów	Kolokwium, egzamin	wykład, ćwiczenia	K_W03, K_W12
3	Znajomość warstwy leksykalnej języków programowania	Kolokwium, egzamin	wykład, ćwiczenia	K_W03
4	Umiejętność wykonania analizy syntaktycznej tekstu oprogramowania	Kolokwium	Wykład, ćwiczenia, laboratorium	K_U17
5	Umiejętność wykorzystania analizatorów Lex i Yacc	Kolokwium, zadanie laboratoryjne	laboratorium	K_U18

<sup>1</sup> należy wskazać ok. 5 – 8 efektów kształcenia

**18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin)**

W. 30    Ćw. - L. 30    P. - Sem. -

**19. Treści kształcenia:****Wykład**

Essence of programming language machine translation: generating equivalent programs expressed in another language. Lexical, syntactic and semantic layers of programs. Connections with the construction of the language virtual machine. Characteristic stages in translating programs to target form: compilation and its phases, consolidation. Translation schema variants.

Language description methods and using them in the text analysis stage. Formal grammars, Chomsky's classification.

Lexical layer of programming languages – regular grammars. (Stack-less) finite-state automata – nondeterministic, deterministic; building, transforming, optimization, application in the course of lexical analysis. *Lex*, the lexical analyser generator.

The syntactic layer – context-free grammars. Grammar transformations: left recursion elimination, (left) factorisation, disambiguation.

Top-down syntax analysis: deterministic analysers based on the recursive descent principle, non-recursive predictive analysis; LL-grammars. Construction of parse-driving table.

Bottom-up syntax analysis. Simple- and operator-precedence grammars – analysis algorithm, constructing parse-driving table. Evaluating precedence functions. LR-grammars. Analysis algorithm, constructing simple (SLR) canonical (LR), look-ahead (LALR) LR-analysers. Building translators with use of the *yacc* generator: introducing semantic actions through translation schemata.

Error handling: detection and diagnosing, error recovery and correction.

**Zajęcia laboratoryjne**

The aim of laboratory classes is to get students practically acquainted with techniques of building text processors, whose action can be described by means of regular expressions and a subclass of context-free grammars. At the beginning, the students recognize *lex* and *yacc* – how to build simple and nuanced *lex* and *yacc* specifications and how to use and debug resulting lexers and parsers. Then the students are assigned individual projects involving parsing mechanisms; the projects should be completed at home and presented at the end of the semester.

**20. Egzamin:** nie**21. Literatura podstawowa:**

A.V.Aho, M.S. Lam, R. Sethi, J.D. Ullman: "Compilers. Principles, techniques, and tools. Second edition". Addison-Wesley, Reading, MA, 2006 (also available for free-download in a .pdf form at <http://www.pdf-search-engine.com/aho-sethi-ullman-pdf.html>)

**22. Literatura uzupełniająca:**

- Ch. Donnelly, R. Stallman: Bison. The YACC-compatible Parser Generator.
- <http://www.gnu.org/software/flex/manual> ,
- <http://www.gnu.org/software/bison/manual/>
- T. Niemann, "A Compact Guide to Lex & Yacc", <http://www.epaperpress.com/lexandyacc/>
- <http://dinosaur.compilertools.net/>

**23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia**

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30 / 15
2	Ćwiczenia	- / -
3	Laboratorium	30 / 45
4	Projekt	- / -
5	Seminarium	- / -
6	Inne (przygotowanie do egzaminu)	- / -
	Suma godzin	60 / 60

**24. Suma wszystkich godzin: 120****25. Liczba punktów ECTS:<sup>2</sup> 4****26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 2****27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 1****26. Uwagi: -**

Zatwierdzono:

.....  
(data i podpis prowadzącego).....  
(data i podpis dyrektora instytutu/kierownika katedry/  
Dyrektora Kolegium Języków Obcych/kierownika lub  
dyrektora jednostki międzywydziałowej)

---

<sup>2</sup> 1 punkt ECTS – 30 godzin.