

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Języki formalne i kompilatory</b>		Kod <b>1010331531010330115</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>2 / 3</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>15</b> Ćwiczenia: <b>15</b> Laboratoria: <b>15</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>4</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>(brak)</b>		(ogólnouczelniany, z innego kierunku) <b>(brak)</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki		Podział ECTS (liczba i %)
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Jolanta Cybulka email: jolanta.cybulka@put.poznan.pl tel. 0-61 6653724 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	1) ma podstawową wiedzę w zakresie matematyki, obejmującą algebrę, analizę, logikę, probabilistykę oraz elementy matematyki dyskretnej i stosowanej 2) ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podstawowych algorytmów i ich analizy, technik projektowania algorytmów, abstrakcyjnych struktur danych i ich implementacji, problemów obliczeniowo trudnych
2	<b>Umiejętności:</b>	1) potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować i uzasadniać opinie 2) potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego
3	<b>Kompetencje społeczne</b>	ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania
<b>Cel przedmiotu:</b>		
Zapoznanie słuchaczy z elementami teorii języków formalnych i teorii translacji oraz zaznajomienie z narzędziami translacji sterowanej składnią w celu wykształcenia umiejętności samodzielnego tworzenia prostych systemów przetwarzania języków formalnych.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podst. konstrukcji programistycznych, implementacji algorytmów, paradygmatów i stylów programowania, metod weryfikacji poprawności programów, języków formalnych, kompilatorów, platform - [K_W05]		
2. ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podstawowych algorytmów i ich analizy, technik projektowania algorytmów, abstrakcyjnych struktur danych i ich implementacji, problemów obliczeniowo trudnych - [K_W04]		
<b>Umiejętności:</b>		
1. potrafi konstruować algorytmy z wykorzystaniem podstawowych technik algorytmicznych i dokonać analizy ich złożoności - [K_U09]		
2. potrafi ocenić przydatność rutynowych metod i narzędzi służących do rozwiązywania prostych zadań inżynierskich typowych dla informatyki oraz wybierać i stosować właściwe technologie - [K_U22]		
<b>Kompetencje społeczne:</b>		
1. ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K_K07]		

<b>Sposoby sprawdzenia efektów kształcenia</b>	
<p>Wykład i ćwiczenia audytoryjne: sprawdzian pisemny z punktowanymi zadaniami (sprawdzenie wiedzy z zakresu teorii języków formalnych oraz teorii i technik translacji), kryterium zaliczenia od 50,1% punktów.</p> <p>Laboratoria: 2 testy pisemne z punktowanymi zadaniami punktów (każdy po serii 2 zajęć laboratoryjnych z danego języka) sprawdzające umiejętność tworzenia prostych przetworników tekstów w językach AWK, Lex i YACC; kryterium zaliczenia od 50,1%</p>	
<b>Treści programowe</b>	
<p>Wykład:</p> <p>Pojęcie języka formalnego. Alfabet, składnia i semantyka języka. Podejście generacyjne i akceptorowe do definiowania składni. Klasyfikacja Chomsky'ego języków formalnych. Języki regularne: automaty skończenie stanowe, wyrażenia regularne. Przetwarzanie języków regularnych za pomocą systemów AWK i Lex. Języki bezkontekstowe: automaty ze stosem, gramatyki bezkontekstowe. Języki kontekstowe i obliczalne oraz ich akceptory. Pojęcie translacji, definicja sterowana składnią, schemat translacji. Deterministyczne języki bezkontekstowe (LL i LR) i ich akceptory. Przetwarzanie języków bezkontekstowych w systemie YACC. Wzmianka o formalnych metodach definiowania semantyki języków programowania. Pojęcie translacji. Interpretacja a kompilacja. Fazy i przebiegi kompilacji. Etap analizy w procesie kompilacji wyrażony jako translacja sterowana składnią: analiza leksykalna i składniowa, analiza zależności kontekstowych. Etap syntezy kodu: języki pośrednie i generacja kodu pośredniego. Elementy systemu wykonawczego: zarządzanie pamięcią i realizacja dostępu do nazw nielokalnych.</p> <p>Ćwiczenia audytoryjne:</p> <p>Rozwiązywanie zadań związanych z formalnym definiowaniem języków oraz specyfikowaniem ich przetworników na bazie translacji sterowanych składnią.</p> <ol style="list-style-type: none"> <li>1. Wyrażenia regularne</li> <li>2. Automaty skończenie stanowe</li> <li>3. Gramatyki bezkontekstowe</li> <li>4. Gramatyki bezkontekstowe II, automaty ze stosem</li> <li>5. Schematy translacji</li> <li>6. Sprawdzian</li> <li>7. Podsumowanie, omówienie wyników sprawdzianu, ćwiczenia uzupełniające</li> </ol> <p>Laboratoria:</p> <p>Implementowanie prostych języków formalnych i ich przetworników w środowisku systemów AWK, Lex i YACC na platformie Linux.</p> <ol style="list-style-type: none"> <li>1. Środowisko wykonawcze + AWK</li> <li>2. AWK</li> <li>3. test AWK + Lex</li> <li>4. Lex</li> <li>5. test Lex + YACC</li> <li>6. YACC</li> <li>7. test YACC</li> <li>8. podsumowanie, zaliczenie</li> </ol>	
<p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>1. Cybulka J., Jankowska B., Nawrocki J. R.: Automatyczne przetwarzanie tekstów. AWK, Lex i YACC, Wyd. NAKOM, Poznań, 2002.</li> <li>2. Hopcroft J.E., Ullman J.D.: Wprowadzenie do teorii automatów, języków i obliczeń, PWN, Warszawa, 1994.</li> <li>3. Aho A.V., Sethi R., Ullman J.: Kompilatory. Reguły, metody i narzędzia. WNT, Warszawa 2002.</li> </ol>	
<p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>1. Dembiński P., Małuszyński J.: Matematyczne metody definiowania języków programowania, WNT, Warszawa 1981.</li> <li>2. Kernighan B.W., Ritchie D.M.: Język ANSI C, WNT, 1994.</li> </ol>	
<b>Bilans nakładu pracy przeciętnego studenta</b>	
Czynność	Czas (godz.)

1. wykłady	15	
2. ćwiczenia audytoryjne	15	
3. ćwiczenia laboratoryjne	15	
4. sprawdziany i konsultacje	5	
5. przygotowanie do ćwiczeń audytoryjnych	10	
6. przygotowanie do ćwiczeń laboratoryjnych	10	
7. przygotowanie do sprawdzianu z: wykład+ćwiczenia audyt.	15	
8. przygotowanie do sprawdzianów z ćwiczeń lab.	15	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	50	2
Zajęcia o charakterze praktycznym	50	2