

1. Nazwa przedmiotu: PODSTAWY INFORMATYKI		2. Kod przedmiotu: MK_13		
3. Karta przedmiotu ważna od roku akademickiego: 2012/2013				
4. Forma kształcenia:		studia pierwszego stopnia		
5. Forma studiów:		studia stacjonarne		
6. Kierunek studiów:		INFORMATYKA (RAU)		
7. Profil studiów:		ogólnoakademicki		
8. Specjalność: -				
9. Semestr:		1 i 2		
10. Jednostka prowadząca przedmiot:		Instytut Informatyki		
11. Prowadzący przedmiot:		dr inż. Robert Tutajewicz		
12. Przynależność do grupy przedmiotów:		przedmioty wspólne		
13. Status przedmiotu:		obowiązkowy		
14. Język prowadzenia zajęć:		polski		
15. Przedmioty wprowadzające oraz wymagania wstępne: brak				
16. Cel przedmiotu:				
Celem przedmiotu jest zapoznanie studenta z podstawowymi pojęciami i problemami spotykanymi w informatyce. Z założenia przedmiot ma stanowić podbudowę dla wielu innych wykładów monograficznych, prowadzonych w kolejnych semestrach. Podstawowe zagadnienia prezentowane w ramach przedmiotu to:				
- pojęcie algorytmu				
- maszynowa reprezentacja danych i realizacja operacji arytmetycznych				
- organizacja jednostki centralnej				
- język assemblera				
- semaforey, klasyczne problemy synchronizacji (producent-konsument, problem pięciu filozofów)				
- podstawowe informacje o sieciach komputerowych				
- przegląd metod dostępu do danych				
17. Efekty kształcenia:¹				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
1	Student zna formalne metody zapisu algorytmów	EP, SP	WT, C	K1A_W08
2	Ma podbudowaną teoretycznie wiedzę w zakresie gramatyk formalnych w zapisie Chomsky’ego	Sprawdzian pisemny, egzamin	Wykład i ćwiczenia tablicowe	K1A_W08
3	Potrafi zasymulować proste algorytmy przy użyciu maszyny Turinga	EP, SP	WT, C	K1A_U13
	Zna podstawowe elementy konstrukcyjne komputera i potrafi omówić rolę każdego z nich. Wie jak działa komputer wg modelu von Neumana	EP	WT	K1A_W07

¹ należy wskazać ok. 5 – 8 efektów kształcenia

Potrafi wyjaśnić rolę układu sterującego komputera, umie zdefiniować proste rozkazy dla zadanej architektury komputera wykładowego (tzw. maszyny W)	EP, SP, CL	WT, C, L	K1A_U19
Potrafi pisać proste programy w języku asemblera komputera wykładowego (maszyny W)	EP, SP, CL	WT, C, L	K1A_U20
Potrafi sformułować sposoby ochrony zasobów niepodzielnych oraz wyjaśnić na czym polega gospodarka zasobami i wzajemna blokada oraz wie jak jej uniknąć	EP, SP	WT, C	K1A_U04 K1A_U19
Umie modelować komputer i sieć komputerową jako stanowisko lub sieć stanowisk obsługi (M/M/1) i analizować ten model	EP, SP, CL	WT, C, L	K1A_U05 K1A_U09

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

	W.	Ćw.	L.	P.	Sem.
Semestr 1	30	30	-	-	-
Semestr 2	15	15	30	-	-

19. Treści kształcenia:

(oddzielnie dla każdej z form zajęć dydaktycznych W./Ćw./L./P./Sem.)

Wykłady są prowadzone z wykorzystaniem wcześniej przygotowanych slajdów dostępnych dla studentów w Internecie.

Treść wykładów:

Algorytmy. Pojęcie i przykłady algorytmów, metody zapisu, szeregowo i równoległa realizacja algorytmów. Efektywność algorytmów, efektywność czasowa i pamięciowa i zależności między nimi, przykłady oceny efektywności algorytmów dla szeregowych i równoległych algorytmów sortowania.

Maszyna Turinga. Twierdzenie o realizowalności algorytmów w maszynie Turinga, tablica charakterystyczna, przykłady zaprogramowania algorytmów w maszynie Turinga.

Języki formalne. Definicja gramatyki, przykłady gramatyk, notacja BNF, algebra Łukasiewicza, pojęcie translacji i przykład algorytmu translacji wyrażeń arytmetycznych z notacji nawiasowej na odwrotną notację polską.

Elementy konstrukcyjne komputera. Rejestry, liczniki, kodery i dekodery. Jednostka arytmetyczno-logiczna, jej konstrukcja i działanie. Magistrale i problemy związane z ich projektowaniem. Pamięci operacyjne: rodzaje, konstrukcja i działanie.

Realizacja rozkazów na przykładzie komputera W. Architektura maszyny W, sygnały sterujące i ich rola. Cykl rozkazowy i równanie rozkazów komputerowych. Synteza układu sterującego komputerem na podstawie jego równań rozkazów w wersji sztywnej i mikroprogramowanej. Inteligentne dystrybutory faz. Pamięć mikroprogramu, formaty mikroinstrukcji, węzły rozdzielające.

System przerwań. Pojęcie przerwania. Organizacja układu przerwań. Priorytety przerwań. Realizacja rozkazu w wersji przerywalnej na przykładzie rozkazu dodawania. Pojęcie procedury obsługi przerwania. Maskowanie przerwań.

Wymiana informacji między jednostką centralną a otoczeniem. Pojęcia informacji elementarnej i bloku informacji. Wymiana informacji elementarnej. Obwód wymiany jako pośrednik między jednostką centralną a urządzeniami zewnętrznymi. Konstrukcja przykładowego obwodu wymiany. Realizacja rozkazów wejścia/wyjścia na przykładzie rozkazu wyprowadzania znaku na urządzenie zewnętrzne. Układ bezpośredniego dostępu do pamięci (DMA), konstrukcja i działanie.

Komputery „wyższych generacji”. Rozwój dróg obiegu adresów i danych w maszynach wyższych generacji. Metody adresacji. Komputer Super W, realizacja dwuargumentowego rozkazu dodawania, skoku ze śladem, końca pętli, dodawania z podwójną precyzją w komputerze Super W.

Języki programowania. Kod maszynowy, programowanie w języku symbolicznym, składnia programu w języku asemblera, przebieg procesu asemblacji, makroassembler, przykłady użycia makrodefinicji, segmentacja. Języki wyższego poziomu, kompilatory. Przykłady kompilacji fragmentów programu w języku wysokiego poziomu. Algorytm kompilacji wyrażenia arytmetycznego

System operacyjny i jego zadania. Rola i funkcje systemu operacyjnego. Synchronizacja i komunikacja między procesami. Pojęcie zasobów systemu. Zasoby podzielne i niepodzielne. Sposoby ochrony zasobów niepodzielnych.

Procedura TAS, zasuwki, semafor. Zadanie o produkcie i konsumencie. Gospodarka zasobami, pojęcia wzajemnej blokady, obszarów zdrowych i chorych. Metody ochrony przed wzajemną blokadą.

Metody dostępu do danych. Listy, drzewa binarne, struktury sieciowe i ich analiza pod kątem czasu wyszukiwania.

Algorytm rozmieszczania rekordów w strukturze drzew binarnego. Struktury B-drzew i zadanie o optymalnej strukturze. Funkcje mieszające i oparte o nie struktury baz danych, problem kolizji i metody ich rozładowywania.

Komputery równoległe. Zapis algorytmu w postaci kanonicznej, twierdzenie o realizowalności algorytmu, realizacja szeregowo i równoległa, definicja przyspieszenia, model komputera sterowanego przepływem argumentów i przykłady realizacji algorytmów w takich systemach

Sieci komputerowe i ich statystyczne modele. Pojęcie sieci komputerowej, media i ich charakterystyka, topologie sieci, model warstwowy. Statystyczny model komputera jako stanowiska obsługi (M/M/1) i jego analiza. Model sieci komputerowej jako sieci stanowisk M/M/1. Sieci otwarte i zamknięte. Metody obliczania rozkładu obciążeń w sieci.

Najnowsze trendy w informatyce. Informatyka a genetyka, nanosystemy informatyki, kwantowe systemy informatyki.

Ćwiczenia tablicowe

Tematyka ćwiczeń tablicowych jest ściśle związana z wykładami i stanowi poszerzenie oraz ilustrację wybranych problemów poruszonych w trakcie wykładów. Zagadnienia prezentowane na wykładach są poszerzane i ilustrowane konkretnymi przykładami i zadaniami w trakcie ćwiczeń tablicowych. Część zadań jest rozwiązywana samodzielnie przez studentów, w trakcie ich rozwiązywania lub po rozwiązaniu często mamy dyskusję pozwalającą rozważyć alternatywne drogi rozwiązania problemu. Na temat ćwiczeń tablicowych składają się następujące zagadnienia:

Algorytmy. Cechy algorytmów, zapis za pomocą języków programowania i schematów blokowych, przykłady algorytmów, szeregowo i równoległa realizacja algorytmów. Efektywność algorytmów, efektywność czasowa i pamięciowa i zależności między nimi, przykłady oceny efektywności algorytmów. Pojęcia też ilustrowane kolejnymi algorytmami realizującymi zadanie znalezienia indeksu najmniejszego i największego elementu jednowymiarowej tablicy.

Rekurencja i przykłady algorytmów rekurencyjnych. Rozwiązania rekurencyjne a rozwiązania iteracyjne. Kiedy stosować a kiedy nie stosować rekurencji. Przykłady algorytmów w których zastosowanie rekurencji jest uzasadnione.

Maszyna Turinga. Przykłady prostych zadań realizowanych za pomocą maszyny Turinga i jej uproszczonej wersji zwanej maszyną Rabina-Scotta

Języki formalne. Definiowanie gramatyk, notacja BNF, klasyfikacja gramatyk wg Chomsky'ego. Sprawdzanie czy dane zdanie jest poprawne w określonej gramatyce. Translacja wyrażeń arytmetycznych z postaci nawiasowej na odwrotną notację polską z użyciem stosu.

Realizacja rozkazów na przykładzie komputera W i jego rozszerzonych wersji. Architektura maszyny W, sygnały sterujące i ich rola. Przesył między rejestrowe. Równania rozkazów komputerowych. Projektowanie rozkazów i synteza układu sterującego. Układ przerwań, realizacja rozkazów w wersji przerywalnej. Rozkaz powrotu z przerwania. Układ wejścia/wyjścia i przykłady rozkazów związanych z wymianą informacji między jednostką centralną a otoczeniem. Maszyna Super W i projektowanie dla niej rozkazów.

Programowanie w języku assemblera i translacja. Zapis programu w języku symbolicznym. Przykłady programów w języku assemblera (organizacja sekwencyjnego przetwarzania danych, sprawdzanie warunku zakończenia pętli, pętle zagnieżdżone itd.). Użycie makrodefinicji i ich przykłady. Przykłady kompilacji wyrażeń arytmetycznych zapisanych w języku wysokiego poziomu do programu w języku assemblera a następnie kodu binarnego.

Semafor i gospodarka zasobami. Rola i funkcje systemu operacyjnego. Synchronizacja i komunikacja między procesami. Pojęcie zasobów systemu. Zasoby podzielne i niepodzielne. Sposoby ochrony zasobów niepodzielnych. Procedura TAS, zasuwki, semafor. Zadanie o produkcie i konsumencie. Gospodarka zasobami, pojęcia wzajemnej blokady, obszarów zdrowych i chorych. Metody ochrony przed wzajemną blokadą.

Metody dostępu do danych. Listy, drzewa binarne, struktury sieciowe i ich analiza pod kątem czasu wyszukiwania. Algorytm rozmieszczania rekordów w strukturze drzew binarnego. Struktury skorowidzowe i zadanie o optymalnej strukturze skorowidza. Funkcje mieszające i oparte o nie struktury baz danych, problem kolizji i metody ich rozładowywania. B-drzewa i związane z nimi algorytmy.

Komputery równoległe. Zapis algorytmu w postaci kanonicznej, twierdzenie o realizowalności algorytmu, realizacja szeregowo i równoległa, definicja przyspieszenia, model komputera sterowanego przepływem argumentów i przykłady realizacji algorytmów w takich systemach

Sieci komputerowe i ich statystyczne modele. Pojęcie sieci komputerowej, media i ich charakterystyka, topologie sieci, model warstwowy. Statystyczny model komputera jako stanowiska obsługi (M/M/1) i jego analiza. Model sieci komputerowej jako sieci stanowisk M/M/1. Sieci otwarte i zamknięte. Metody obliczania rozkładu obciążeń w sieci.

Laboratorium

Ćwiczenia laboratoryjne są prowadzone na drugim semestrze studiów stacjonarnych. W trakcie laboratorium studenci rozwiązują wskazane przez prowadzącego zadania, wykorzystując specjalnie do tego celu przygotowane oprogramowanie. W ramach laboratorium prowadzone są następujące ćwiczenia:

1. Projektowanie rozkazów dla maszyny W																								
2. Programy w języku asemblera maszyny W																								
3. Układ wejścia/wyjścia																								
4. Układ przerwań																								
5. Metody dostępu do danych																								
6. Analiza sieci stanowisk typu M/M/1																								
20. Egzamin: tak (po drugim semestrze)																								
21. Literaturapodstawowa:																								
1. Stefan Węgrzyn: <i>Podstawy informatyki</i> . PWN, Warszawa, 1982																								
2. <i>Wykłady z Podstaw Informatyki profesora Stefana Węgrzyna</i> . Skrypt uczelniany Politechniki Śląskiej nr 1812																								
22. Literaturauzupełniająca:																								
1. <i>Zbiór zadań z Podstaw Informatyki</i> . Praca zbiorowa pod redakcją Stanisława Kozielskiego. Skrypt uczelniany Politechniki Śląskiej nr 1459																								
2. M. Chłopek, R. Tutajewicz: <i>Ćwiczenia laboratoryjne z Podstaw Informatyki - maszyna W</i> . Skrypt uczelniany Politechniki Śląskiej																								
23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia																								
<table border="1"> <thead> <tr> <th>Lp.</th> <th>Forma zajęć</th> <th>Liczba godzin kontaktowych / pracy studenta</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Wykład</td> <td>45/20</td> </tr> <tr> <td>2</td> <td>Ćwiczenia</td> <td>45/20</td> </tr> <tr> <td>3</td> <td>Laboratorium</td> <td>30/20</td> </tr> <tr> <td>4</td> <td>Projekt</td> <td>/</td> </tr> <tr> <td>5</td> <td>Seminarium</td> <td>/</td> </tr> <tr> <td>6</td> <td>Inne (przygotowanie do egzaminu)</td> <td>/20</td> </tr> <tr> <td></td> <td>Suma godzin</td> <td>120/80</td> </tr> </tbody> </table>	Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta	1	Wykład	45/20	2	Ćwiczenia	45/20	3	Laboratorium	30/20	4	Projekt	/	5	Seminarium	/	6	Inne (przygotowanie do egzaminu)	/20		Suma godzin	120/80
Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta																						
1	Wykład	45/20																						
2	Ćwiczenia	45/20																						
3	Laboratorium	30/20																						
4	Projekt	/																						
5	Seminarium	/																						
6	Inne (przygotowanie do egzaminu)	/20																						
	Suma godzin	120/80																						
24. Suma wszystkich godzin: 200																								
25. Liczba punktów ECTS:² 8 (odpowiednio w sem. 1 i 2: 4 + 4)																								
26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego 5																								
27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty) 2																								
26. Uwagi:																								

Zatwierdzono:

.....
(data i podpis prowadzącego)

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

² 1 punkt ECTS – 25 do 30 godzin.