

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Introduction to Information Technology</b>		Kod <b>1010514311010519079</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 1</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>angielski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>16</b> Ćwiczenia: <b>-</b> Laboratoria: <b>16</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>5</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki		Podział ECTS (liczba i %)
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr hab. inż. Jerzy Nawrocki, prof. PP email: jerzy.nawrocki@put.poznan.pl tel. 665-2980 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Zgodnie z podstawą programową kształcenia ogólnego dostępną na stronie: <a href="http://bip.men.gov.pl/men/bip/akty_prawne/rozporzadzenie_20081223_zal_4.pdf">http://bip.men.gov.pl/men/bip/akty_prawne/rozporzadzenie_20081223_zal_4.pdf</a> zakłada się, że rozpoczynając przedmiot student ma podstawową wiedzę: - z matematyki: IV etap edukacyjny, zakres podstawowy poszerzony o rachunek różniczkowy (z zakresu rozszerzonego); - z informatyki: IV etap edukacyjny, zakres podstawowy.
2	<b>Umiejętności:</b>	Zgodnie z podstawą programową kształcenia ogólnego dostępną na stronie: <a href="http://bip.men.gov.pl/men/bip/akty_prawne/rozporzadzenie_20081223_zal_4.pdf">http://bip.men.gov.pl/men/bip/akty_prawne/rozporzadzenie_20081223_zal_4.pdf</a> zakłada się, że rozpoczynając przedmiot student ma podstawowe umiejętności: - z matematyki: IV etap edukacyjny, zakres podstawowy poszerzony o rachunek różniczkowy (z zakresu rozszerzonego); - z informatyki: IV etap edukacyjny, zakres podstawowy.
3	<b>Kompetencje społeczne</b>	W zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
Po zaliczeniu tego modułu student powinien mieć dobrą orientację w podstawowych obszarach informatyki oraz powinien umieć czytać i pisać proste programy dotyczące bardzo różnych zagadnień i napisane z wykorzystaniem różnych paradygmatów programowania. Ponadto przedmiot powinien inspirować studentów do dalszego pogłębiania wiedzy w tym zakresie.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. ma podstawową wiedzę w zakresie algorytmów i złożoności, budowy komputera, systemów operacyjnych, sieci komputerowych, różnych paradygmatów programowania (programowanie imperatywne i język C, programowanie na poziomie asemblera, programowanie regułowe), sztucznej inteligencji (przetwarzanie języka naturalnego), baz danych, inżynierii oprogramowania, oraz systemów sterowania; - [K_W4] 2. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych, - [K_W6] 3. ma wiedzę nt. kodeksów etycznych dotyczących informatyki, rozumie specyfikę systemów krytycznych ze względu na bezpieczeństwo (ang. mission-critical systems); - [K_W10] 4. ma podstawową wiedzę nt. patentów, ustawy prawo autorskie i prawa pokrewne oraz ustawy o ochronie danych osobowych; - [K_W14]		
<b>Umiejętności:</b>		

1. posługiwać się technikami programistycznymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych - [K\_U6]
2. przy formułowaniu i rozwiązywaniu zadań informatycznych, dostrzegać ich aspekty społeczne i prawne. - [K\_U9]

#### Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, - [K\_K1]
2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życie - [K\_K4]
3. docenia wagę indywidualnego doskonalenia oraz zna podstawowe zasady efektywnej pracy zespołowej. - [K\_K5]

### Sposoby sprawdzenia efektów kształcenia

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę przygotowania studenta do poszczególnych sesji zajęć laboratoryjnych (sprawdzian "wejściowy") oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,
- ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych poprzez 2 kolokwia w semestrze,
- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym (od 5 do 10 zadań, każde zadanie oceniane w skali od 0 do 10 punktów; na ocenę dostateczną trzeba zdobyć przynajmniej 55% punktów).

### Treści programowe

Przedmiot obejmuje następujące zagadnienia:

Zasady skutecznego działania (proaktywność; zaczynaj mając koniec na względzie; najważniejsze rzeczy najpierw; myślenie o obopólnej wygranej; najpierw staraj się zrozumieć, a dopiero potem oczekuj, że będziesz rozumiany; dbaj o synergię; ?ostrz piłę?)

Programowanie imperatywne i C (programy z jedną instrukcją; koncepcja zmiennej; drukowanie wartości zmiennej; czytanie liczb; instrukcja warunkowa; uproszczona instrukcja warunkowa; instrukcja powtarzania ?while?; modularyzacja i funkcje)

Od algebry Boole'a do komputera (algebra Boole'a; sumator 4-bitowy; bramki; rejestry; dekodery; operacje boolowskie w C)

Asembler i koncepcja von Neumanna (architektura typowego procesora; prosty program w języku assemblera; dodawanie liczb heksadecymalnych; reprezentacja liczb ujemnych; koncepcja von Neumanna; rozkazy skoku; podprogramy w języku assemblera)

Algorytmy i struktury danych (tablice; rekordy; dynamiczny przydział pamięci; grafy i listy; stosy; kolejki; heapsort; złożoność obliczeniowa; problem stopu)

Metody numeryczne (reprezentacja liczb rzeczywistych; stabilność numeryczna; metoda stycznych i obliczanie pierwiastka; wzór Maclaurina i funkcje  $\exp$ ,  $\cos x$ )

Inżynieria oprogramowania (analiza problemu; wymagania funkcjonalne i przypadki użycia; wybrane diagramy języka UML; wstępny projekt interfejsu użytkownika; wymagania pozafunkcjonalne i architektura; implementacja i testowanie)

Przetwarzanie tekstów i AWK (idea języka AWK; najprostsze programy; wzorce wiersza; wyrażenia regularne; zmienne)

Systemy operacyjne (historia systemów operacyjnych; pojęcie procesu; interferencja obliczeń; metoda ścisłej wymiany; instrukcja TSL; semafony binarne; problem producent-konsument)

Komputerowe systemy sterowania (pojęcie systemów czasu rzeczywistego; projektowanie metodą HRT HOOD; szacowanie maksymalnego czasu wykonania programów; szeregowanie zadań)

Bazy danych (era przed-relacyjna; relacyjne bazy danych; projekcja, selekcja i połączenie w języku SQL; tworzenie, wstawianie i sortowanie danych w języku SQL; wzorce; operacje przydatne do pielęgnacji bazy danych; funkcje agregujące; zagnieżdżanie zapytań)

Sztuczna inteligencja i język naturalny (test Turinga; program ELIZA; części mowy i niejednoznaczność; analiza leksykalna i program Lex, pojęcie gramatyki formalnej; translacja; gramatyki bezkontekstowe)

Sieci komputerowe (stos protokołów internetowych i zagnieżdżanie danych; gniazda TCP; koncepcja `software_as_a_service`; serwisy webowe; WSDL, UDDI, XML i SOAP)

Prawne i etyczne aspekty informatyki (ustawa o ochronie danych osobowych; prawo autorskie; patenty; ryzyko informatyzacji i Therac-25; dobre praktyki dotyczące budowy systemów krytycznych ze względu na bezpieczeństwo; kod etyczno-zawodowy ACM)

Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy,
2. ćwiczenia laboratoryjne: ćwiczenia praktyczne, wykonywanie eksperymentów, dyskusja, praca w zespole, studium przypadków.

#### Literatura podstawowa:

1. Język C ? Programowanie, B.W. Kernighan, D.M. Ritchie, Helion, 2010.
2. Układy cyfrowe, B. Wilkinson, WKiŁ, Warszawa, 2000
3. Programowanie komputerów IBM PC w języku assemblera, J. Nawrocki, WPP, Poznań, 1991
4. Wprowadzenie do przetwarzania tekstów w języku AWK, J. Nawrocki, W. Complak, ProDialog 2, 23-46, Poznań, 1994
5. Sieci komputerowe, J.F. Kurose, K.W. Ross, Helion, 2006

<b>Literatura uzupełniająca:</b> 1. 7 nawyków skutecznego działania, S. Covey, Rebis, 2003		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>		<b>Czas (godz.)</b>
1. 1.	udział w wykładach	16
2. 2.	udział w ćwiczeniach laboratoryjnych	16
3. 3.	przygotowanie do ćwiczeń laboratoryjnych	20
4. 4.	zapoznanie się ze wskazaną literaturą (10 stron tekstu naukowego = 1 godz.), 500 stron	50
5. 5.	przygotowanie do egzaminu i obecność na egzaminie: 18 godz. + 2 godz.	20
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	122	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1
Zajęcia o charakterze praktycznym	36	1