

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Informatyka</b>		Kod <b>1010801121010810031</b>
Kierunek studiów <b>Elektronika i Telekomunikacja</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 2</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>2</b> Ćwiczenia: <b>-</b> Laboratoria: <b>2</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>7</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>ogólnouczelniany</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>7 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>  prof. dr hab. inż. Jerzy Tyszer email: jerzy.tyszer@put.poznan.pl tel. +48 61 665 3814 Elektroniki i Telekomunikacji ul. Piotrowo 3A 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Podstawowe wiadomości z logiki matematycznej i kombinatoryki. Umiejętność formułowania prostych algorytmów.
2	<b>Umiejętności:</b>	Potrafi pozyskiwać informacje z literatury oraz innych źródeł w języku polskim lub angielskim; potrafi integrować uzyskane informacje, dokonywać ich interpretacji i wyciągać wnioski.
3	<b>Kompetencje społeczne</b>	Zna ograniczenia własnej wiedzy i umiejętności, rozumie konieczność dalszego dokształcania się.
<b>Cel przedmiotu:</b> Zapoznanie z podstawami inżynierii oprogramowania. Przedmiot wprowadza kolejne zagadnienia zarówno praktyki programowania obiektowego komputerów w języku C++ jak i projektowania struktur danych i algorytmów oraz analizy ich złożoności obliczeniowej.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. Podstawowa wiedza teoretyczna i praktyczna w zakresie programowania w językach C i C++, ze szczególnym uwzględnieniem projektowania programów poprawnie zbudowanych, zasad konstruowania oprogramowania obiektowego, wykorzystania szablonów, projektowania złożonych programów oraz wykorzystywania oprogramowania bibliotecznego. - [K1_W09]		
2. Wiedza o podstawowych algorytmach (sortowanie, przeszukiwanie zbiorów danych, metody zachłanne, metody prób i błędów, wybrane algorytmy numeryczne) i strukturach danych (pojemniki, listy jedno i dwukierunkowe, drzewa poszukiwań binarnych, drzewa zrównoważone, grafy i metody ich przeszukiwania, dendryty) wykorzystywanych w codziennej praktyce programisty. - [K1_W09]		
3. Przeglądowa wiedza na temat metod stosowanych w projektowaniu złożonego instrumentarium informatycznego oraz obiektowym projektowaniu specjalizowanych struktur danych. - [K1_W09]		
<b>Umiejętności:</b>		
1. Przy projektowaniu oprogramowania student potrafi przeprowadzić analizę problemu z punktu widzenia postępowania algorytmicznego stosując kryteria złożoności obliczeniowej, szybkości działania programu, skalowalności zastosowanych rozwiązań, oraz adekwatności przyjętych metod. - [K1_U13]		
2. Przy projektowaniu oprogramowania student potrafi dokonać właściwej dekompozycji problemu, dokonać wyboru adekwatnych struktur danych, wyodrębnić hierarchię obiektów, zidentyfikować relacje między obiektami i ich reprezentantami w programie. - [K1_U13]		
3. Student potrafi krytycznie zanalizować dostępne oprogramowanie biblioteczne pod kątem zastosowania w realizowanym projekcie oraz zaproponować zasady współpracy w konfiguracji zbiorowego programisty. - [K1_U13]		
<b>Kompetencje społeczne:</b>		

1. Zrozumienie potrzeby szerszej popularyzacji wiedzy z zakresu nowoczesnych technik informatycznych. - [K1_K02]
2. Świadomość możliwości i ograniczeń współczesnej informatyki przy jednoczesnym otwarciu na możliwość zastosowań w nowych dziedzinach życia codziennego, gospodarki, techniki i nauki. - [K1_K01]
3. Umiejętność formułowania własnych opinii na temat aktualnie stosowanych i dostępnych technologii i rozwiązań w projektowaniu nowoczesnych systemów informatycznych. - [K1_K03]

<b>Sposoby sprawdzenia efektów kształcenia</b>		
-Pisemny egzamin z zakresu treści wykładowych (pytania problemowe lub test wyboru)		
- Kolokwia pisemne obejmujące zadania wykonywane w ramach ćwiczeń laboratoryjnych		
<b>Treści programowe</b>		
-Historia automatyzacji obliczeń, architektura komputerów, struktura programu w języku C++, podstawowe typy danych, reprezentacja uzupełnieniowa liczb binarnych, operatory i wyrażenia, instrukcje sterujące, tablice, funkcje, przekazywanie argumentów, wzorce funkcji, teoria obliczeń, algorytmy rekurencyjne i zachłanne, metody numeryczne, szybkie algorytmy sortowania i przeszukiwania, złożoność obliczeniowa, sztuczna inteligencja, wskaźniki i dynamiczne przydzielanie pamięci, podstawowe struktury danych (wektory, listy, stopy, kolejki, drzewa, grafy), klasy i obiekty klas, dziedziczenie, polimorfizm, wzorce klas, programowanie zorientowane obiektowo, nowoczesna inżynieria oprogramowania, systemy transakcyjne, oprogramowanie telekomunikacyjne.		
<b>Literatura podstawowa:</b>		
1. D.E. Knuth, The art of computer programming, Addison-Wesley Publishing Company, Reading, MA, 1968, 1973.		
2. N. Wirth, Algorytmy + struktury danych = programy, WNT, Warszawa, 1980		
3. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Wprowadzenie do algorytmów, WNT, Warszawa, 2004.		
4. E.W. Dijkstra, Umiejętność programowania, WNT, Warszawa, 1985.		
5. J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków 2008.		
<b>Literatura uzupełniająca:</b>		
1. W. Lipski, Kombinatoryka dla programistów, WNT, Warszawa, 1982.		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. Uczestnictwo w wykładach i ćwiczeniach laboratoryjnych	60	
2. Przygotowanie do egzaminu	30	
3. Przygotowanie do ćwiczeń laboratoryjnych	60	
4. Prace studialne	15	
5. Konsultacje z osobami prowadzącymi ćwiczenia i wykład	5	
6. Uczestnictwo w egzaminie	3	
7. Zaliczenie laboratorium	2	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	175	7
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	70	2
Zajęcia o charakterze praktycznym	95	4