

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Języki programowania		Kod 1010251141010210531
Kierunek studiów Inżynieria biomedyczna	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2 / 4
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 1 Ćwiczenia: - Laboratoria: 1 Projekty/seminaria: -		Liczba punktów 2
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) ogólnouczelniany
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 2 100% 2 100%
Odpowiedzialny za przedmiot / wykładowca: dr hab. inż. Maciej TABASZEWSKI email: Maciej.Tabaszewski@put.poznan.pl tel. 61 665 - 2390 Wydział Budowy Maszyn i Zarządzania ul. Piotrowo 3 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	podstawowa dotycząca elementów technologii informatycznych
2	Umiejętności:	logicznego myślenia, pozyskiwania informacji z biblioteki i Internetu
3	Kompetencje społeczne	rozumienie potrzeby uczenia się i poszerzania swojej wiedzy
Cel przedmiotu: Przekazanie wiedzy pozwalającej na samodzielne opracowywanie i wykorzystywanie programów narzędziowych i baz danych z wykorzystaniem technik programowania proceduralnego i obiektowego.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza: 1. Student powinien być w stanie scharakteryzować cechy programowania proceduralnego, obiektowego, wizualnego. - [K_W04] 2. Student powinien być w stanie opisać podstawowe konstrukcje w języku C i C++ (pętle, instrukcja warunkowa, funkcja, zmienna, wskaźnik, typy zmiennych, stała, struktura, klasa, obiekt, referencja, konstruktor, destruktor, hermetyzacja, dziedziczenie, polimorfizm). - [K_W15] 3. Student powinien znać podstawowe algorytmy (sortowanie) i struktury danych (tablica jedno i wielowymiarowa, stos, kolejka) - [K_W15]		
Umiejętności: 1. Student potrafi tworzyć oprogramowanie narzędziowe - [K_U01; K_U02; K_U05; K_U08; K_U20]		
Kompetencje społeczne: 1. Student rozumie rolę informatyzacji we współczesnej gospodarce. Potrafi w niej twórczo uczestniczyć. - [K_K01 K_K02]		
Sposoby sprawdzenia efektów kształcenia		
Wykład: Zaliczenie na podstawie testu. Laboratorium: Bieżąca ocena na podstawie odpowiedzi pisemnej z zakresu wymaganego do realizacji danego ćwiczenia. Na ostatnich zajęciach sprawdzian praktycznych umiejętności tworzenia prostych programów.		

Treści programowe		
<p>Wykład:</p> <ol style="list-style-type: none"> Ogólne zasady konstruowania programów. Kompilatory i interpretatory. Programowanie w językach niskiego i wysokiego poziomu, przegląd i podział języków: Assembler, Logo, Basic, Pascal, Prolog, C#, Java, C, C++. Wizualne języki programowania. Programowanie strukturalne. Podstawy programowania w języku C. Zmienne, typy danych, wskaźniki, operatory, pętle, instrukcje warunkowe, funkcje wejścia i wyjścia. Programowanie zorientowane obiektowo. Pojęcia hermetyzacji, klas, obiektów, dziedziczenia, polimorfizmu. Podstawy programowania w języku C++. Referencje, przeciążanie operatorów, strumienie, wyjątki, przestrzenie nazw. Budowa i obsługa interfejsu użytkownika. Środowisko RAD. Tworzenie elementów graficznych. Budowa baz danych. <p>Laboratorium:</p> <ol style="list-style-type: none"> Programowanie strukturalne w języku C (proste obliczenia i przetwarzanie danych). Programowanie obiektowe cz. 1 (proste obliczenia i przetwarzanie danych w C++, strumienie). Programowanie obiektowe cz. 2 (wykorzystanie metod zaawansowanych na potrzeby eksploracji danych) Programowanie obiektowe cz. 3 - budowa programu do przetwarzania informacji zapisanej w pliku binarnym uzyskanym z aparatury pomiarowej. Prosta baza danych. 		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> Liberty J., Rao S., Jones B, L, - C++ dla każdego, Helion, Gliwice 2011 Wróblewski P., Algorytmy, struktury danych i techniki programowania, Helion, Gliwice 2009 		
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> Sedgewick R., Algorytmy w C++, READ ME, Łódź 1999 Kliszewski M., Inżynieria oprogramowania obiektowego, WKT, Warszawa 1994 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. Przygotowanie do wykładu	4	
2. Udział w wykładzie	15	
3. Utrwalanie treści wykładu	4	
4. Konsultacje	4	
5. Przygotowanie do zaliczenia	8	
6. Udział w zaliczeniu	2	
7. Udział w zajęciach laboratoryjnych	15	
8. Udział w zaliczeniu zajęć laboratoryjnych	2	
9. Przygotowanie do zajęć laboratoryjnych	6	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	60	2
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	38	1
Zajęcia o charakterze praktycznym	15	1