

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Podstawy programowania</b>		Kod <b>1010334511010334957</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>1 / 1</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>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>(brak)</b>		(ogólnouczelniany, z innego kierunku) <b>(brak)</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>5 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>  dr Jerzy Bartoszek email: jerzy.bartoszek@put.poznan.pl tel. 61 665-3713, 61 665-2378 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	ma podstawową wiedzę wynikającą z programu szkoły średniej
2	<b>Umiejętności:</b>	potrafi realizować zadania wynikające z programu szkoły średniej
3	<b>Kompetencje społeczne</b>	ma kompetencje społeczne wynikające z programu szkoły średniej
<b>Cel przedmiotu:</b> Prezentacja podstawowych stylów programowania i konstrukcji programistycznych z przykładami programów w językach C/C++.		
<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]		
<b>Umiejętności:</b> 1. potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego - [K_U10] 2. potrafi konstruować algorytmy z wykorzystaniem podstawowych technik algorytmicznych i dokonać analizy ich złożoności - [K_U09]		
<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>		
Wykład: testy pisemne z pytaniami punktowanymi i kryterium zaliczenia od 50,1% punktów. Laboratorium: sprawdziany, ocena wykonanych ćwiczeń i sprawozdań.		
<b>Treści programowe</b>		

<p>Wykłady:                  Algorytm a program. Podstawowe style programowania: imperatywny, deklaratywny, obiektowy. Podstawowe struktury danych w językach C i C++. Podstawowe konstrukcje programistyczne: deklaracje i definicje zmiennych, stałych i ich typów, operatory arytmetyczne i logiczne, wyrażenia, instrukcje przypisania, instrukcje warunkowe, instrukcje pętli, instrukcja goto, instrukcje wejścia-wyjścia, pliki i strumienie. Funkcje i procedury. Parametry. Wskaźniki. Dynamiczny przydział pamięci i implementacja dynamicznych struktur danych. Rekurencja i jej implementacja. Metody weryfikacji poprawności programu.</p> <p>Laboratoria:                  Wprowadzenie do Visual Studio: edytowanie, kompilowanie, wykonywanie i śledzenie programów.                  Deklaracje i definicje zmiennych. Proste instrukcje wejścia i wyjścia.                  Instrukcje przypisania, instrukcje warunkowe.                  Tablice jedno i wielowymiarowe, instrukcje pętli.                  Funkcje i procedury oraz ich parametry.                  Wskaźniki i dynamiczny przydział pamięci. Struktury.                  Dynamiczne struktury danych: listy, kolejki, stosy, drzewa.</p>		
<p><b>Literatura podstawowa:</b>                  1. Grębosz J., Symfonia C++ standard, Wydawnictwo &amp;#38;#34;Edition 2000&amp;#38;#34;, Kraków, 2005.                  2. Schildt H., Programowanie C++, Wydawnictwo RM, Warszawa, 2002                  3. Stroustrup B., Język C++, WNT, Warszawa, 2004</p>		
<p><b>Literatura uzupełniająca:</b>                  1. Banachowski L., Diks K., Rytter W., Algorytmy i struktury danych, WNT, Warszawa, 2006                  2. Powers L., Snell M., Microsoft Visual Studio 2010. Księga eksperta, Wydawnictwo Helion, 2011</p>		
<p><b>Bilans nakładu pracy przeciętnego studenta</b></p>		
<b>Czynność</b>		<b>Czas (godz.)</b>
1. wykłady		16
2. laboratoria		16
3. konsultacje i egzamin		8
4. przygotowanie do ćw. lab., wykonanie sprawozdań		48
5. przygotowanie do sprawdzianów i egzaminu		40
<p><b>Obciążenie pracą studenta</b></p>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	128	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	40	2
Zajęcia o charakterze praktycznym	75	3