



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez  
Unię Europejską w ramach  
Europejskiego Funduszu  
Społecznego

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>						
Programowanie		13.2.0158						
<b>Nazwa jednostki prowadzącej przedmiot</b>								
Instytut Fizyki Doświadczalnej								
<b>Studia</b>								
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>					
Wydział Matematyki, Fizyki i Informatyki	Fizyka	<b>forma</b>	stacjonarne					
		<b>moduł</b>	fizyka					
		<b>specjalnościowy</b>	Podstawowa					
<b>specjalizacja</b>								
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>								
dr Janusz Młodzianowski; dr Sławomir Werbowy								
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>					<b>Liczba punktów ECTS</b>			
<b>Formy zajęć</b>					4 Przedmiot w wymiarze 30h wykładu i 30h ćwiczeń w laboratorium komputerowym			
Wykład, Ćw. laboratoryjne								
<b>Sposób realizacji zajęć</b>								
zajęcia w sali dydaktycznej								
<b>Liczba godzin</b>								
Ćw. laboratoryjne: 30 godz., Wykład: 30 godz.								
<b>Cykl dydaktyczny</b>								
2015/2016 letni								
<b>Status przedmiotu</b>				<b>Język wykładowy</b>				
obowiązkowy				polski				
<b>Metody dydaktyczne</b>				<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>				
- praca własna - wykład - ćwiczenia w laboratorium komputerowym, praca własna				<b>Sposób zaliczenia</b>				
				- Zaliczenie na ocenę - Zaliczenie (zał)				
				<b>Formy zaliczenia</b>				
				kolokwium				
				<b>Podstawowe kryteria oceny</b>				
				Umiejętność samodzielnego, poprawnego napisania prostego programu w dowolnym kompilatorze języka C/C++, Umiejętność wyjaśnienia celowości zastosowania, w pisamym programie, konkretnych konstrukcji języka C/C++, Umiejętność posługiwania się, przynajmniej w podstawowym zakresie, wierszem poleceń systemu Windows, Obecność na zajęciach.				
<b>Sposób weryfikacji założonych efektów kształcenia</b>								
<b>zakładany efekt kształcenia</b>	<b>Kolokwium</b>	<b>Ocena pracy z komputerem i oprogramowaniem</b>	<b>mtd. dydakt 3</b>	<b>mtd. dydakt 4</b>	<b>mtd. dydakt 5</b>	<b>mtd. dydakt 6</b>	<b>mtd. dydakt 7</b>	<b>mtd. dydakt 8</b>
<b>Wiedza</b>								
K_W12	+	+						
<b>Umiejętności</b>								
K_U12	+	+						

<p><b>Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi</b></p> <p><b>A. Wymagania formalne</b>  <b>B. Wymagania wstępne</b>                  Znajomość matematyki w zakresie 1 semestru studiów.</p>	
<p><b>Cele kształcenia</b></p> <p>Celem zajęć jest przeprowadzenie podstawowego kursu programowania w języku C/C++.</p>	
<p><b>Treści programowe</b></p> <ol style="list-style-type: none"> <li>Przechowywanie i kodowanie informacji w komputerze. Kodowanie ISO646, ISO8859-*, CP***, CP****, Unicode, U2, IEEE754-1985. Techniczne fazy tworzenia programu. Edycja, kompilacja, scalanie i diagnostyka programu.</li> <li>Projektowanie i programowanie strukturalne. Cykl życia projektu, podział na fazy i ich charakterystyka. Projektowanie metodą krokowego udoskonalania. Techniki wspomagające projektowanie. Schematy blokowe, diagramy strukturalne, Warnier-Orr'a, diagramy N-S. Elementy notacji BNF. Dokumentacja. Pojęcie i przykłady algorytmów.</li> <li>Historia rozwoju i specyfika języków C/C++. Zintegrowane pakiety: Borland, Microsoft, GNU, Preprocesor, Biblioteki standardowe. Struktura programu w języku C. Konwencje nazewnictwa, komentarze, słowa kluczowe dyrektywy.</li> <li>Deklaracje i podstawowe typy zmiennych. Definiowanie własnych typów. Konwersje typów.</li> <li>Zmienne wskaźnikowe. Pojęcie wskaźnika. Operator adresu i zawartości. Arytmetyka na wskaźnikach.</li> <li>Instrukcje wyboru i powtarzania. Omówienie i przykłady zastosowania.</li> <li>Funkcja jako podstawowa konstrukcja języka C/C++. Pojęcie funkcji i prototypu funkcji, wywołanie funkcji. Parametry formalne i aktualne. Przesyłanie argumentów przez wartość i adres. Zwracanie wyniku przez nazwę funkcji. Parametry wywołania funkcji main().</li> <li>Operatory. Operatory arytmetyczne, logiczne, bitowe, relacyjne, przypisania, operator przecinkowy.</li> <li>Pojęcie lokalności i globalności. Zmienne lokalne, globalne, automatyczne, statyczne, rejestrowe, zewnętrzne. Przesłanie zmiennych. Zmienne dynamiczne, tworzenie i usuwanie zmiennych dynamicznych.</li> <li>Komunikacja z urządzeniami wejścia/wyjścia, operacje plikowe.</li> <li>Elementy projektowania i programowania obiektowego. Pojęcie i przykłady obiektów. Enkapsulacja. Polimorfizm. Dziedziczenie. Omówienie metod projektowania obiektowego.</li> <li>Elementy języka C++. Przeciążanie funkcji i operatorów. Argumenty domniemane funkcji. Konstrukcje obiektowe klasy, struktury i unii. Funkcja i metoda. Konstruktor i destruktor. Szablony.</li> <li>Elementy programowania w systemie Windows. Okno, i pętla komunikatów. WinMain(), Win32 API.</li> </ol>	
<p><b>Wykaz literatury</b></p> <p>B. W. Kernighan, D. M. Ritchie, Język ANSI C, WNT 1988                  J. Bielecki, ANSI C++, Intersoftland 1997                  N. Wirth, Wstęp do programowania systematycznego, WNT 1978                  D. Harel, Rzecz o istocie informatyki. Algorytmika, WNT 1992                  E. Yourdon, Współczesna analiza strukturalna, WNT 1996                  J. Martin, J. J. Odell, Podstawy metod obiektowych, WNT 1997                  C. Petzold, Programowanie Windows, Microsoft</p>	
<p><b>Efekty kształcenia (obszarowe i kierunkowe)</b></p> <p>K_W12 zna podstawy analizy numerycznej, zna na poziomie podstawowym co najmniej jeden pakiet do obliczeń symbolicznych, zna podstawowe pakiety oprogramowania użytkowego do prezentacji wyników i analizy danych; zna podstawy programowania i inżynierii oprogramowania</p> <p>K_U12 potrafi skompilować, uruchomić, testować i udokumentować napisany samodzielnie program komputerowy</p>	<p><b>Wiedza</b></p> <p>Student ma opanowane podstawowe konstrukcje języka C/C++. Zna typy zmiennych (char, int, float, tablica i struktura). Zna składnię instrukcji wyboru (if, if-else, switch), instrukcji iteracji (for, do, while) oraz operatory działań arytmetyczno logicznych. Rozumie czym jest funkcja i prototyp funkcji. Wie na czym polegają mechanizmy przesyłania parametrów do funkcji (lista formalna, przesyłanie przez wartość, adres, referencję). Zna pojęcie wskaźnika. Wie na czym polegają różnice między językami C i C++. Zna pojęcie obiektu.</p> <p>Zna pojęcie algorytmu programu i technikę projektowania metodą schematów blokowych. Wie co to jest złożoność obliczeniowa i poprawność całkowita. Rozumie na czym polega metodologia projektowania i programowania strukturalnego. Zna technikę dekompozycji funkcjonalnej i krokowego udoskonalania.</p>
	<p><b>Umiejętności</b></p> <p>Student potrafi sformułować za pomocą techniki schematów blokowych prosty algorytm numeryczny a na jego podstawie napisać program w języku C. Umie zaprojektować program składający się z kilku funkcji. Potrafi uzasadnić celowość użycia listy formalnej funkcji z przesłaniem argumentów przez wartość i adres. Potrafi zmodyfikować, poprzez zastosowanie innych rodzajów instrukcji wyboru i/lub iteracji, wybrany fragment programu. otrafi napisać program odczytujący z zapisujących dane numeryczne do i z pliku na dysku.</p>

	<b>Kompetencje społeczne (postawy)</b>
<b>Kontakt</b>	
fizjm@univ.gda.pl	