Nazwa przedmiotu
Programowanie 2

Kod ECTS
3.4.7SX.PRG2

Nazwa jednostki prowadzącej przedmiot
Uniwersytet Opolski, Wydział Matematyki, Fizyki i Informatyki, Instytut matematyki i Informatyki

Studia

Kierunek	stopień	tryb	specjalność	specjalizacja
Informatyka	*Pierwszy*	*Stacjonarne* *Niestacjonarne*[*)]		

Nazwisko osoby prowadzącej (osób prowadzących)
Pracownicy Zakładu Informatyki

Formy zajęć, sposób ich realizacji i przypisana im liczba godzin

A. Formy zajęć
- *wykład (W),*
- *laboratorium (L).*

B. Sposób realizacji
- *zajęcia w sali wykładowej/ dydaktycznej/laboratoryjnej*

C. Liczba godzin

Wykład – 15 godzin
Laboratorium – 45 godzin

*) Studia niestacjonarne:
Wykład – 8 godz. (2T+6Z)
Laboratorium –26 godzin

Liczba punktów ECTS: 5

Bilans nakładu pracy przeciętnego studenta:
- *5 godz. – wstępny przegląd literatury [*) 5]*
- *15×1 godz. = 15 godz. – udział w wykładach [*) 8]*
- *15×3 godz. = 45 godz. – udział w labortatoriach [*) 26]*
- *15×1 godz. = 15 godz. – analiza i przyswojenie treści wykładu [*) 23]*
- *5 × 1 godz. = 5 godz. – udział w konsultacjach do wykładu [*) 5]*
- *15×1 godz. = 15 godz. – przygotowanie do labortatoriów [*) 18]*
- *15 godz. – przygotowanie projektu do wykładu [*) 23]*
- *10 godz. – przygotowanie dosprawdzianów zaliczeniowych na laboratoriach [*) 18]*

Łączny nakład pracy studenta: 125 godzin, co odpowiada 5 pkt. ECTS

w tym
- *nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 15+45+5=**65** godz., co odpowiada **2** pkt. ECTS;*
- *nakład pracy związany z zajęciami o charakterze praktycznym: 45+15+10+15 =**85** godz., co odpowiada **3** pkt. ECTS*

*) na studiach niestacjonarnych:
- *nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 8+26+5=39 godz., co odpowiada **2** pkt. ECTS;*
- *nakład pracy związany z zajęciami o charakterze praktycznym: 26+18+18+23 =**85** godz., co odpowiada **3** pkt. ECTS*

Status przedmiotu
- *obowiązkowy (kanon)*

Język wykładowy
Polski (możliwość realizacji w języku angielskim)

Metody dydaktyczne
- *wykład / wykład problemowy / wykład z prezentacją multimedialną*
- *ćwiczenia laboratoryjne: projekt i implementacja programów komputerowych*

Forma i sposób zaliczenia oraz podst. kryteria oceny lub wymagania egzaminacyjne
Na ogólnych zasadach określonych w programie kształcenia, a w szczególności

A. Sposób zaliczenia
- *zaliczenie z oceną laboratorium*
- *zaliczenie z oceną (wykład)*

B. Formy zaliczenia
- *(W) ustalenie oceny zaliczeniowej na podstawie oceny projektu programistycznego*
- *(L) zaliczenie z oceną: ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru za pisanie programów komputerowych, a także na podstawie ocen z kolokwiów*

C. Podstawowe kryteria
- *(W) (L) uzyskanie pozytywnej oceny zaliczeniowej*

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi
Należy określić:
A. *Wymagania formalne: zaliczenie/ocena pozytywna z przedmiotu Programowanie 1*
B. *Wymagania wstępne: znajomość paradygmatu programowania proceduralnego i praktyczna umiejętność stosowania podstawowych technik programistycznych w tym zakresie*

Cele przedmiotu

Przedmiot stanowi rozszerzenie i uzupełnienie przedmiotu „Programowanie 1" w zakresie podstaw metodologii programowania i jej praktycznego wykorzystania do tworzenia prostych programów z interfejsem graficznym. Pojęcia teoretyczne są ilustrowane w wybranym języku programowania w zintegrowanym środowisku programistycznym. Nabycie przez studenta umiejętności tworzenia aplikacji opartych o graficzny interfejs użytkownika.

Treści programowe

***A.** Problematyka wykładu / **B.** Problematyka laboratorium*

Przegląd paradygmatów programowania. Praca w zintegrowanym środowisku programistycznym. Obsługa zdarzeń. Wykorzystanie obsługi zdarzeń we własnych programach. Komponenty GUI i sposoby ich wykorzystania. Tablice: definicja, inicjacja tablic, tablice wielowymiarowe. Typ: string. Typ wskaźnikowy. Struktury, unie. Obiekty dynamiczne: operator new, gcnew, operator delete. Usuwanie obiektów. Dynamiczne struktury danych (lista jednokierunkowa, lista dwukierunkowa). Kolekcje. Serializacja. Wyjątki. Wyrażenia regularne. Operacje wejścia-wyjścia. Operacje na plikach. Testowanie i debugowanie programów.

Wykaz literatury

A. Literatura wymagana

A.1. wykorzystywana podczas zajęć

1. Gręboszz, Jerzy :Symfonia C++ standard : programowanie w języku C++ orientowane obiektowo. T.1, 2008
2. Gręboszz, Jerzy :Symfonia C++ standard : programowanie w języku C++ orientowane obiektowo. T.2, 2008

A.2. studiowana samodzielnie przez studenta

1. Gręboszz, Jerzy :Symfonia C++ standard : programowanie w języku C++ orientowane obiektowo. T.1, 2008
2. Gręboszz, Jerzy :Symfonia C++ standard : programowanie w języku C++ orientowane obiektowo. T.2, 2008

B. Literatura uzupełniająca

1. Stroustrup, Bjarne.: Język C++ 2004
2. inne podręczniki dostępne on-line poprzez Bibliotekę Główną UO („ibuk")

Efekty kształcenia

Wiedza

Symb.	Efekt	Metoda weryfikacji	Odniesienie
W01	Zna paradygmat programowania proceduralnego i zorientowanego obiektowo.	obserwacja, sprawdzian pisemny	K_W03
W02	Posiada wiedzę z zakresu wskaźników i referencji. Zna metody dynamicznego przydziału pamięci. Zna metody weryfikacji działania programów.	obserwacja, sprawdzian pisemny	K_W04
W03	Zna zasadę działania procesów i wątków.	obserwacja, sprawdzian pisemny	K_W07
W04	Posiada ogólną wiedzę nt. paradygmatów programowania i języków programowania.	obserwacja, sprawdzian pisemny	K_W09

Umiejętności:

Symb.	Efekt	Metoda weryfikacji	Odniesienie
U01	Potrafi ocenić przydatność różnych paradygmatów programowania i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów.	Sprawdzian pisemny	K_U03, K_U28
U02	Potrafi tworzyć i interpretować kod źródłowy w C++.	Konwersacja, Obserwacja	K_U32
U03	Potrafi dobrać narzędzie do tworzenia aplikacji w trybie graficznym	Obserwacja	K_U04, K_U32
U04	Potrafi konstruować i programować algorytmy pod kątem języków programowania.	Sprawdzian pisemny	K_U11
U05	Potrafi stosować podstawowe techniki programistyczne w zakresie programowania	Sprawdzian pisemny	K_U09, K_U10
U06	Potrafi projektować, implementować, testować i debugować proste programy okienkowe	Obserwacja, projekt	K_U04, K_U09, K_U32, K_U07, K_U28, K_U34
U07	Potrafi pisać programy okienkowe w środowisku graficznym.	Obserwacja, projekt	K_U04
U08	Potrafi tworzyc aplikacje wielowątkowe.	Obserwacja, projekt	K_U17

Kompetencje społeczne (postawy)

Symb.	Efekt	Metoda weryfikacji	Odniesienie
K01	Jest świadomy konieczności ciągłej aktualizacji wiedzy dotyczącej języków programowania i narzędzi programistycznych	Konwersacja	K_K01
K02	Rozumie konieczność systematycznej pracy nad złożonym programem komputerowym	Obserwacja	K_K02

Kontakt:

Wykaz numerów telefonicznych i adresów mailowych pracowników znajduje się na stronie Instytutu Matematyki i Informatyki: `www.math.uni.opole.pl`