

KARTA KURSU

Nazwa	Programowanie II	
Nazwa w j. ang.	Programming II	
Koordinator	dr hab. Piotr Czerski, prof. UKEN	Zespół dydaktyczny
		dr hab. Piotr Czerski, prof. UKEN
Punktacja ECTS*	2	

Opis kursu (cele kształcenia)

Celem kursu jest nauka podstaw programowania obiektowego oraz poznanie technik średnio zaawansowanego programowania z zastosowaniem języka C++. Przedmiot prowadzony jest w języku polskim z użyciem terminologii angielskiej.

Warunki wstępne

Wiedza	Student zna podstawy składni, podstawowe typy danych oraz struktury kontrolne i funkcje języka C, C++
Umiejętności	Student potrafi pisać proste programy użytkowe w języku C, C++ stosując paradygmat strukturalny i proceduralny.
Kursy	Wprowadzenie do Programowania, Programowanie I

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	W01: zna podstawowe pojęcia związane z programowaniem obiekowym: klasa, obiekt, dziedziczenie, polimorfizm, itp.	K_W04
	W02: zna techniki programowania zorientowanego obiektowego;	K_W04
	W03: posiada wiedzę na temat wybranych zaawansowanych elementów programowania w języku C++;	K_W04
	W04: zna wybrane biblioteki pozwalające tworzyć interfejsy graficzne.	K_W04

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	U01: potrafi projektować oraz implementować własne klasy różnorodnych obiektów i tworzyć ich instancje;	K_U03
	U02: projektuje i tworzy kompletne programy z wykorzystaniem metodologii obiektowej;	K_U03
	U03: wykorzystuje zaawansowane elementy języka C++ w swoich programach;	K_U03
	U04: potrafi wykorzystać zdobyte umiejętności do projektowania interfejsów graficznych z wykorzystaniem odpowiednich bibliotek	K_U03

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Kompetencje społeczne	K01: potrafi korzystać z różnych źródeł informacji (w tym zasobów sieciowych) do poszerzania własnej wiedzy i zdobywania nowych umiejętności;	K_K02
	K02: wykazuje umiejętność stosowania w praktyce zdobytej wiedzy przedmiotowej i potrafi działać kreatywnie w celu rozwiązywania postawionych mu zadań;	K_K02
	K03: potrafi przekazywać wiedzę informatyczną w sposób zrozumiały dla innych i współpracować w zespole.	K_K02

Organizacja												
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin	10					20						

Opis metod prowadzenia zajęć

Kurs składa się z wykładu i ćwiczeń prowadzonych w formie laboratoriów. W ramach laboratoriów studenci projektują i tworzą zadane programy w języku C++, które następnie są omawiane. Poza zajęciami w formie tradycyjnej studenci biorą udział w zajęciach z wykorzystaniem platformy e-learningowej.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01	x				x			x				x	
W02	x				x			x				x	
W03	x				x			x				x	
W04	x				x			x				x	
U01	x				x			x					
U02	x				x			x				x	
U03	x				x			x				x	
U04	x				x			x					
K01					x								
K02					x			x					
K03					x			x					

Kryteria oceny	Ocena jest wystawiana na podstawie wykonanych przez studentów projektów programistycznych i/lub testu sprawdzającego wiedzę z zakresu omawianych treści merytorycznych. Ocenę dobrą i bardzo dobrą może uzyskać student, który: wykazuje się dobrą lub bardzo dobrą znajomością składni i narzędzi języka C++ pozwalających programować z wykorzystaniem metodologii obiektowej; potrafi wykorzystywać techniki programowania obiektowego do tworzenia wyspecjalizowanych klas i budowy odpowiednich ich hierarchii; potrafi dbać o jakość pisanych programów i wykorzystywać zaawansowane elementy języka C++;
----------------	---

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

1. Podstawy analizy, projektowania i programowania obiektowego.
2. Pojęcie klasy, hermetyzacja, dziedziczenie, polimorfizm.
3. Podstawy programowania w języku C++.
4. Dynamiczne zarządzanie pamięcią w C++ – operatory new i delete.
5. Dostęp: publiczny, chroniony i prywatny do pól i metod.
6. Wybrane elementy biblioteki standardowej języka C++.
7. Przeciążanie funkcji, funkcje zaprzyjaźnione z klasą.
8. Przeciążanie operatorów.
9. Referencje.
10. Stałe wartości, stałe obiekty oraz metody obiektów stałych.
11. Tworzenie i niszczenie obiektów – konstruktory i destruktory.
12. Wskaźnik „this” – jego znaczenie i sposób użycia.
13. Funkcje i zmienne statyczne.
14. Szablony funkcji
15. Szablony klas

Wykaz literatury podstawowej

1. Eckel B.: Thinking in C++, t.1, Helion 2002
2. Eckel B.: Thinking in C++, t. 2, Helion 2004
3. Grębosz J.: Opus Magnum C++, Helion 2020
4. Prata S.: Język C++. Szkoła programowania, Helion 2013
5. Stroustrup B.: Programowanie. Teoria i praktyka z wykorzystaniem C++, Helion 2013
6. Stroustrup B.: Język C++. Kompendium wiedzy, Helion 2014

Wykaz literatury uzupełniającej

1. Dattatri K.: Język C++. Efektywne programowanie obiektowe, Helion 2005
2. Josuttis N. M.: C++. Programowanie zorientowane obiektowo. Vademecum profesjonalisty, Helion 2003
3. Josuttis N. M.: C++. Biblioteka standardowa, Helion 2014
4. Lippman S., Lajoie J.: Podstawy języka C++ , WNT 2003
5. Schildt H., C++. Sztuka programowania, Helion 2004
6. Shtern V.: „Core C++. Inżynieria programowania”, Helion 2003

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta)

liczba godzin w kontakcie z prowadzącymi	Wykład	10
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	0
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	5
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	5
	Przygotowanie do egzaminu/zaliczenia	10
Ogółem bilans czasu pracy		50
Liczba punktów ECTS w zależności od przyjętego przelicznika		2

