

KARTA KURSU

Nazwa	Programowanie obiektowe
Nazwa w j. ang.	Object oriented programming

Koordynator	Prof. UP dr hab. Piotr Czerski	Zespół dydaktyczny
		Prof. UP dr hab. Piotr Czerski dr Łukasz T. Stępień
Punktacja ECTS*	st. stacjonarne: 2	

Opis kursu (cele kształcenia)

Celem kursu jest zapoznanie studentów z podstawami analizy, projektowania i programowania obiektowego oraz nauczenie postaw programowania w języku C++.
Kurs prowadzony jest w języku polskim.

Warunki wstępne

Wiedza	Student zna podstawowe zagadnienia z algorytmiki (struktury danych i proste algorytmy) oraz składnię języka C.
Umiejętności	Potrafi zapisywać podstawowe algorytmy i definiować struktury danych za pomocą języka C.
Kursy	Podstawy oprogramowania, Matematyka

Efekty kształcenia

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Wiedza	W01: wymienia i omawia cechy obiektowego podejścia do programowania.	K_W10
	W02: ma wiedzę na temat mechanizmów pozwalających na programowanie obiektowe z zastosowaniem języka C++.	K_W10
	W03: orientuje się na poziomie podstawowym w zagadnieniach programowania generycznego w języku C++ (zna szablony klas i funkcji).	K_W10
	W04: zna składnię języka C++ i potrafi wskazać różnice pomiędzy językami C i C++	K_W10

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Umiejętności		

	U01: potrafi zapisywać podstawowe algorytmy i struktury danych w języku C++.	K_U13
	U02: projektuje i tworzy z wykorzystaniem podstaw metodologii obiektowej proste programy w języku C++.	K_U13
	U03: kompiluje, uruchamia i znajduje błędy w napisanych przez siebie programach w języku C++.	K_U13
	U04: potrafi korzystać z wybranych funkcji, klas i szablonów z biblioteki standardowej i używać ich w pisanych przez siebie programach.	K_U13

Kompetencje społeczne	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
	K01: potrafi korzystać z różnych źródeł informacji (m.in. zasobów sieciowych) w celu poszerzania własnej wiedzy i zdobywania nowych umiejętności. K02: wykazuje umiejętność praktycznego stosowania zdobytej wiedzy przedmiotowej, a także potrafi działać kreatywnie w celu rozwiązywania napotkanych problemów.	K_K01, K_K02 K_K02, K_K03

Studia stacjonarne

Organizacja												
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin	15					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 kształcenia

	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	X
W02	X				X			X				X	X
W03	X				X			X				X	X
W04	X				X			X					X
U01	X				X			X				X	X
U02	X				X			X				X	X
U03	X				X			X					X
U04	X				X			X				X	X
K01	X				X			X				X	X
K02	X				X			X				X	X

Kryteria oceny	<p>Ocenę dobrą lub bardzo dobrą może uzyskać student, który:</p> <ul style="list-style-type: none"> • implementuje różne operatory dla zaprojektowanych klas, • zna i stosuje w praktyce problematykę dynamicznego zarządzania pamięcią (w tym tworzy własne operatory przypisania, konstruktory i destruktory dla klas korzystających z dynamicznego zarządzania pamięcią), • potrafi implementować i wykorzystywać szablony funkcji, • poprawnie korzysta z funkcji wirtualnych,
----------------	--

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. Tworzenie i niszczenie obiektów – konstruktory i destruktory.
11. Wskaźnik „this” – jego znaczenie i sposób użycia.
12. Funkcje i zmienne statyczne.
13. Szablony klas i funkcji.

Wykaz literatury podstawowej

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

Wykaz literatury uzupełniającej

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

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) **studia stacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	5
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	3
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	2
	Przygotowanie do egzaminu/zaliczenia	10
Ogółem bilans czasu pracy		60
Liczba punktów ECTS w zależności od przyjętego przelicznika		2