

KARTA KURSU

Nazwa	Języki i Narzędzia Programowania Obiektowego
Nazwa w j. ang.	Object-Oriented Programming Languages and Tools

Koordinator	mgr inż. Agnieszka Smolarek	Zespół dydaktyczny
		dr M. Ciura mgr inż. Agnieszka Smolarek
Punktacja ECTS*	st. stacjonarne: 2 st. niestacjonarne: 2	

Opis kursu (cele kształcenia)

Celem kursu jest zapoznanie studentów z podstawami analizy, projektowania i programowania obiektowego, ze szczególnym uwzględnieniem dobrych praktyk związanych z bezpieczeństwem aplikacji w języku C++.

Studenci poznają techniki programowania obiektowego, mechanizmy zarządzania pamięcią oraz narzędzia do analizy i testowania bezpieczeństwa kodu.

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	Programowanie

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	W01: wymienia i omawia cechy obiektowego podejścia do programowania.	K_W07
	W02: ma wiedzę na temat mechanizmów pozwalających na programowanie obiektowe z zastosowaniem języka C++.	K_W07
	W03: orientuje się na poziomie podstawowym w zagadnieniach programowania generycznego w języku C++ (zna szablony klas i funkcji).	K_W07
	W04: zna składnię języka C++ i potrafi wskazać różnice między językiem C i C++.	K_W07

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	U01: potrafi implementować podstawowe algorytmy i struktury danych w języku C++ w oparciu o zasady bezpieczeństwa.	K_U01
	U02: projektuje i tworzy z wykorzystaniem podstaw metodologii obiektowej proste programy w języku C++.	K_U05
	U03: kompiluje, uruchamia i znajduje błędy w napisanych przez siebie programach w języku C++.	K_U05
	U04: potrafi korzystać z wybranych funkcji, klas i szablonów z biblioteki standardowej i używać ich w pisanych przez siebie programach.	K_U05
	U05: potrafi analizować i testować bezpieczeństwo aplikacji napisanych w C++.	K_U06

	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_K01, K_K02
	K02: wykazuje umiejętność stosowania w praktyce zdobytej wiedzy przedmiotowej i potrafi działać kreatywnie w celu rozwiązywania napotkanych problemów.	K_K02, K_K04
	K03: Zna zasady etycznego programowania i bezpieczeństwa kodu.	K_K02

Studia stacjonarne

Organizacja												
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin						30						

Studia niestacjonarne

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

Opis metod prowadzenia zajęć

Kurs składa się z ćwiczeń prowadzonych w formie laboratoriów. W ramach zajęć studenci projektują i tworzą zadane programy w języku C++, które następnie są omawiane.
Metody oceny: Kolokwia i testy teoretyczne, Zadania programistyczne na zajęciach.

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					
W02					X			X					
W03					X			X					
W04					X			X					
W05					X			X					
U01					X			X					
U02					X			X					
U03					X			X					
U04					X			X					
U05					X			X					
K01					X			X					
K02					X			X					
K03					X			X					

Kryteria oceny

Ocenę dobrą lub bardzo dobrą może uzyskać student, który:

- potrafi projektować i implementować w języku C++ średniozaawansowane klasy
- 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,
- potrafi w praktyce stosować zasady poprawnego definiowania i używania obiektów stałych.
- analizuje i testuje kod pod kątem bezpieczeństwa.

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. Podstawy bezpieczeństwa w programowaniu w C++.
15. Metody testowania i analizy bezpieczeństwa aplikacji C++.

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) **studia stacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	10
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	25
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	
	Przygotowanie do egzaminu/zaliczenia	
Ogółem bilans czasu pracy		65
Liczba punktów ECTS w zależności od przyjętego przelicznika		2

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

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