# **KARTA KURSU**

|  |  |
| --- | --- |
| Nazwa | **Programowanie I** |
| Nazwa w j. ang. | Programming I |

|  |  |  |
| --- | --- | --- |
| Koordynator | dr Łukasz T. Stępień | Zespół dydaktyczny |
| dr Łukasz T. Stępień |
|  |  |
| Punktacja ECTS\* | st. stacjonarne: 2 |

Opis kursu (cele kształcenia)

|  |
| --- |
| Celem kursu jest zapoznanie studentów z elementami algorytmiki i struktur danych, podstawami programowania proceduralnego i programowania obiektowego oraz nauczenie postaw programowania w języku C++. Kurs jest prowadzony w języku polskim. |

Warunki wstępne

|  |  |
| --- | --- |
| Wiedza | Student zna podstawowe pojęcia związane z programowaniem (zmienna, polecenie, funkcja) na poziomie szkoły średniej. Student posiada pewną kulturę logiczną oraz matematyczną. |
| Umiejętności | Student posiada umiejętność opisania prostych algorytmów np. za pomocą schematu blokowego lub listy kroków. |
| Kursy | Logika I, Wprowadzenie do programowania |

Efekty kształcenia

|  |  |  |
| --- | --- | --- |
| Wiedza | Efekt kształcenia dla kursu | Odniesienie do efektów kierunkowych |
| W01: wymienia i omawia cechy obiektowego podejścia do programowania.  W02: zna składnię języka C++ oraz wybrane funkcje z biblioteki standardowej  W03: zna instrukcje sterujące przepływem programu  W04: zna mechanizm wskaźników i referencji  W05: zna paradygmat programowania proceduralnego  W06: zna wybrane proste algorytmy oraz wybrane proste struktury danych, a także sposób ich zapisu w języku C++  W07: ma wiedzę na temat mechanizmów pozwalających na programowanie obiektowe z zastosowaniem języka C++.  W08: orientuje się na poziomie podstawowym w zagadnieniach programowania generycznego w języku C++ (zna szablony klas i funkcji). | K\_W03  K\_W03  K\_W03  K\_W03  K\_W03  K\_W03  K\_W03  K\_W03 |

|  |  |  |
| --- | --- | --- |
| Umiejętności | Efekt kształcenia dla kursu | Odniesienie do efektów kierunkowych |
| U01: potrafi poprawnie używać składni języka C++ oraz wybranych funkcji z biblioteki standardowej  U02: potrafi stosować instrukcje sterujące przepływem programu  U03: potrafi poprawnie używać wskaźników i referencji  U04: potrafi tworzyć własne funkcje w języku C++  i stosować paradygmat programowania proceduralnego  U05: potrafi zapisywać podstawowe algorytmy i struktury danych w języku C++.  U06: tworzy z wykorzystaniem podstaw metodologii obiektowej, proste programy w języku C++.  U07: kompiluje, uruchamia i znajduje błędy w napisanych przez siebie programach w języku C++.  U08: potrafi definiować i używać szablony funkcji i klas w języku C++ | K\_U03  K\_U03  K\_U03  K\_U03  K\_U03  K\_U03  K\_U03 |

|  |  |  |
| --- | --- | --- |
| Kompetencje społeczne | Efekt kształcenia dla kursu | Odniesienie do efektów kierunkowych |
| K01: potrafi korzystać z różnych źródeł informacji (w tym zasobów sieciowych) do poszerzania własnej wiedzy i zdobywania nowych umiejętności.  K02: wykazuje umiejętność stosowania w praktyce zdobytej wiedzy przedmiotowej i potrafi działać kreatywnie w celu rozwiązywania napotkanych problemów. | K\_K02  K\_K01, K\_K02 |

**Studia stacjonarne**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 formie stacjonarnej lub zdalnej. W ramach laboratoriów studenci projektują i tworzą zadane programy w języku C++, które następnie są omawiane. |

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 |  |  | X |
| U01 | X |  |  |  | X |  |  | X |  | X |  |  | X |
| U02 | X |  |  |  | X |  |  | X |  | X |  |  | X |
| U03 | X |  |  |  | X |  |  | X |  | X |  |  | X |
| U04 | X |  |  |  | X |  |  | X |  | X |  |  | X |
| K01 | X |  |  |  | X |  |  | X |  | X |  |  | X |
| K02 | X |  |  |  | X |  |  | X |  | X |  |  | X |
| ... |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| Kryteria oceny | Ocenę **dostateczną** może uzyskać student, który:  - potrafi napisać w języku C++ proste programy z użyciem instrukcji sterujących oraz mechanizmu wskaźników i referencji  - potrafi implementować w języku C++ wybrane proste algorytmy i wybrane proste struktury danych.  Ocenę **dobrą** lub **bardzo dobrą** może uzyskać student, który:  - potrafi zdefiniować własną klasę z różnymi modyfikatorami dostępu  - potrafi poprawnie użyć mechanizmu dziedziczenia  - 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, m.in. konstruktory kopiujące),  - potrafi implementować i wykorzystywać szablony funkcji i klas,  - poprawnie korzysta z funkcji wirtualnych. |

|  |  |
| --- | --- |
| Uwagi | Niezbędna jest **samodzielna** praca studentów z materiałem uprzednio przerabianym na zajęciach. |

Treści merytoryczne (wykaz tematów)

|  |
| --- |
| 1. Składnia języka C++ i środowisko programistyczne 2. Podstawy programowania proceduralnego 3. Podstawy programowania obiektowego. 4. Wybrane proste algorytmy sortowania i wyszukiwania, rekurencja 5. Pojęcie klasy, hermetyzacja, dziedziczenie, polimorfizm. 6. Dynamiczne zarządzanie pamięcią w C++ – operatory new i delete. 7. Dostęp: publiczny, chroniony i prywatny do pól i metod. 8. Wybrane proste struktury danych: stos, kolejka FIFO 9. Wybrane elementy biblioteki standardowej języka C++. 10. Przeciążanie funkcji, funkcje zaprzyjaźnione z klasą. 11. Przeciążanie operatorów (m.in. operatora przypisania). 12. Tworzenie i niszczenie obiektów – konstruktory i destruktory, konstruktor kopiujący 13. Wskaźnik „this” – jego znaczenie i sposób użycia. 14. Funkcje i zmienne statyczne. 15. Szablony klas i funkcji. |

Wykaz literatury podstawowej

|  |
| --- |
| 1. Eckel B.: Thinking in C++, t.1, Helion 2002 (wybrane rozdziały) 2. Eckel B.: Thinking in C++, t. 2, Helion 2004 (wybrane rozdziały) 3. Grębosz J.: Symfonia C++: programowanie w języku C++ zorientowane obiektowo, T. 1, Wydawnictwo "Edition 2000" Oficyna Kallimach, Kraków 2010 (wybrane rozdziały) 4. Grębosz J.: Symfonia C++: programowanie w języku C++ zorientowane obiektowo, T. 2, Wydawnictwo "Edition 2000" Oficyna Kallimach, Kraków 2010 (wybrane rozdziały) 5. Prata S.: Język C++. Szkoła programowania, Helion 2013 (wybrane rozdziały) 6. Stroustrup B.: Programowanie. Teoria i praktyka z wykorzystaniem C++, Helion 2013 (wybrane rozdziały) 7. Stroustrup B.: Język C++. Kompendium wiedzy, Helion 2014 (wybrane rozdziały) 8. Wróblewski P.: Algorytmy. Struktury danych i techniki programowania, Helion 2003 (wybrane rozdziały) |

Wykaz literatury uzupełniającej

|  |
| --- |
| 1. Bhargava, A. Y.: Algorytmy. Ilustrowany przewodnik, Helion, Gliwice 2017 (wybrane rozdziały) 2. Booch G., Rumbaugh J., Jacobson I.: UML. Przewodnik użytkownika, WNT 2002 (wybrane rozdziały) 3. Dattatri K.: Język C++. Efektywne programowanie obiektowe, Helion 2005 (wybrane rozdziały) 4. Josuttis N. M.: C++. Programowanie zorientowane obiektowo. Vademecum profesjonalisty, Helion 2003 (wybrane rozdziały) 5. Josuttis N. M.: C++. Biblioteka standardowa, Helion 2014 (wybrane rozdziały) 6. Lippman S., Lajoie J.: Podstawy języka C++ , WNT 2003 (wybrane rozdziały) 7. Schildt H., C++. Sztuka programowania, Helion 2004 (wybrane rozdziały) 8. Shtern V.: „Core C++. Inżynieria programowania”, Helion 2003 (wybrane rozdziały) 9. Roth S.: Czysty kod w C++17. Oprogramowanie łatwe w utrzymaniu, Helion 2018 (wybrane rozdziały) |

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

|  |  |  |
| --- | --- | --- |
| liczba godzin w kontakcie z prowadzącymi | Wykład | 10 |
| Konwersatorium (ćwiczenia, laboratorium itd.) | 20 |
| Pozostałe godziny kontaktu studenta z prowadzącym | 3 |
| liczba godzin pracy studenta bez kontaktu z prowadzącymi | Lektura w ramach przygotowania do zajęć | 10 |
| Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu | 10 |
| Przygotowanie projektu lub prezentacji na podany temat (praca w grupie) |  |
| Przygotowanie do egzaminu/zaliczenia | 7 |
| Ogółem bilans czasu pracy | | 60 |
| Liczba punktów ECTS w zależności od przyjętego przelicznika | | 2 |