

Karta opisu zajęć - Sylabus

Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu

I. INFORMACJE OGÓLNE

| | | | |
|--|---------------|---|---|
| Nazwa zajęć: Podstawy programowania | | Kod zajęć: C11 | |
| Nazwa kierunku studiów, poziom i profil kształcenia: | | Informatyka, studia pierwszego stopnia, profil praktyczny | |
| Język wykładowy: polski | Rodzaj zajęć: | zajęcia kształcenia kierunkowego | |
| Rok studiów: 1 | Semestr: 1 | Liczba punktów ECTS przypisana zajęciom: 7 | Data aktualizacji sylabusu: 01.10.2021 r. |
| Instytut (Zakład) odpowiedzialny za zajęcia: | | Instytut Inżynierii Technicznej | |
| Imię, nazwisko, tytuł/stopień naukowy, adres e-mail wykładowcy (wykładowców)/prowadzących zajęcia: | | Tomasz Lewandowski, dr inż., Tomasz.lewandowski@pwste.edu.pl Przemysław Hawro, mgr inż., przemyslaw.hawro@pwste.edu.pl | |

FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN

Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:

| Studia stacjonarne | | Studia niestacjonarne | |
|---------------------|----|-----------------------|--|
| Wykład: | 30 | Wykład: | |
| Ćwiczenia: | | Ćwiczenia: | |
| Laboratorium: | 45 | Laboratorium: | |
| Lektorat: | | Lektorat: | |
| Projekt: | | Projekt: | |
| Zajęcia praktyczne: | | Zajęcia praktyczne: | |
| Seminarium: | | Seminarium: | |
| Zajęcia terenowe: | | Zajęcia terenowe: | |
| Praktyki: | | Praktyki: | |
| Inna forma (jaka): | | Inna forma (jaka): | |
| RAZEM: | 75 | RAZEM: | |

II. INFORMACJE SZCZEGÓŁOWE

Cel (cele) prowadzenia zajęć:

Celem przedmiotu jest przedstawienie podstaw technik programowania opartych na właściwościach języka programowania C++, tj. podstawowe konstrukcje algorytmiczne, tworzenie i kompilacja programów, proste i złożone typy danych, programowanie proceduralne, operacje wejścia-wyjścia, pamięć operacyjna i wskaźniki.

Wymagania wstępne w zakresie wiedzy, umiejętności oraz kompetencji społecznych (jeśli obowiązują):

Podstawowe umiejętności matematyczne oraz informatyczne na poziomie szkoły średniej

Przypisane do zajęć efekty uczenia się w zakresie wiedzy, umiejętności oraz kompetencji społecznych i odniesienie ich do efektów uczenia się dla określonego kierunku studiów, poziomu i profilu.

UWAGA:

Dzielimy efekty uczenia się przypisane do zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Przypisane do zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii.

| | | | |
|---|---|--|--|
| Symbol efektów uczenia się przypisanego do zajęć* | Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii: | Odniesienie do efektów uczenia się dla określonego kierunku studiów, poziomu i profilu # | |
| Wiedzy - zna i rozumie | | | |
| M_01 | podstawowe metody, techniki i narzędzia stosowane w programowaniu imperatywnym. | K_W01, K_W15 | |
| M_02 | podstawowe pojęcia programistyczne: zadanie algorytmiczne, instrukcja warunkowa, iteracja, rekurencja; budowę programu w języku C++, zakresów dostępności zmiennych, budowy funkcji i sposobów przekazywania parametrów, wykorzystywania i roli wskaźników, struktur i unii w języku C++. | K_W06, K_W20 | |
| Umiejętności - potrafi | | | |
| M_03 | wybrać odpowiednią metodę algorytmiczną do postawionego problemu oraz napisać program w języku C++ realizujący określone wymagania funkcjonalne. | K_U01, K_U02, K_U08 | |
| M_04 | korzystać z gotowych bibliotek dostarczanych wraz z językiem programowania w projektowaniu i implementacji oprogramowania. | K_U08, K_U11 | |
| Kompetencji społecznych - jest gotów do | | | |
| M_05 | pozyskiwania informacji z zasobów internetowych oraz literatury niezbędnych do rozwiązania napotkanych problemów. | K_K01, K_K04 | |
| M_06 | podnoszenia swoich kwalifikacji zawodowych poprzez uczestnictwo w szkoleniach specjalistycznych oraz zdobywanie certyfikatów developerskich. | K_K01, K_K04 | |
| <p>* kod zajęć, # efekty uczenia się dla określonego kierunku studiów, poziomu i profilu (np. K_W01, K_U01, ..) W- wiedza, U- umiejętności, K- kompetencje społeczne 01, 02...- numer efektu uczenia się UWAGA! Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne w zależności od ogólnej liczby godzin zajęć.</p> | | | |
| TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ PRZYPISANYCH DO ZAJĘĆ | | | |
| Symbol treści programowych | Opis treści programowych | Liczba godzin | Odniesienie do efektów uczenia się przypisanych do zajęć |
| wykład | | | |
| TP-01 | Współczesne aspekty programowania. Pojęcie języka strukturalnego, struktura programu w języku C++, pliki źródłowe i nagłówkowe. Pojęcie algorytmu, przykłady podstawowych algorytmów. | 6 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-02 | Wybrane środowiska programistyczne. Strumienie wejścia/wyjścia. Typy i formaty zmiennych, konwersja typów. | 2 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-03 | Podstawowe konstrukcje programistyczne, instrukcje: przypisania, warunkowa, wyboru. Operatory arytmetyczne, relacyjne, logiczne, priorytety operatorów. | 2 | M_01, M_02, M_03, M_04, M_05, M_06 |

| | | | |
|--|--|---|------------------------------------|
| TP-04 | Instrukcje iteracyjne for, while, do-while., Złożone typy danych: tablice jednowymiarowe, wielowymiarowe, znakowe, typy wyliczeniowe. | 4 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TK-05 | Definicja a deklaracja funkcji. Widoczność zmiennych, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje. Funkcje rekurencyjne i biblioteczne. | 2 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TK-06 | Pojęcie wskaźnika, dynamiczne alokowanie i zwolnienie pamięci dla tablic. Poruszanie się po tablicy za pomocą wskaźnika. Arytmetyka wskaźników. Wykorzystanie wskaźników w funkcjach. | 4 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TK-07 | Statyczne wydzielenie pamięci i inicjalizacja. Dynamiczne alokowanie i zwolnienie pamięcią. | 2 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TK-08 | Typ strukturalny łączący dane. Tworzenie struktur oraz tablic struktur. Dynamiczne alokowanie i zwolnienie pamięci. Posługiwanie się składnikami struktur. Zagnieżdżanie struktur. | 4 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TK-09 | Struktury danych: kolejka, stos, lista. Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Zapis i odczyt do/z pliku. | 4 | M_01, M_02, M_03, M_04, M_05, M_06 |
| laboratorium | | | |
| TP-10 | Tworzenie, wczytywanie i zapisywanie projektów. Operacje edycyjne, konfiguracyjne i awaryjne. Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Operacje arytmetyczne, priorytet operacji, konwersja typów. | 5 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-11 | Pisanie prostych programów, programy z rozgałęzieniami z zastosowaniem instrukcji warunkowych, programy iteracyjne wykorzystujące niedeterministyczne pętle typu while oraz do-while. | 4 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-12 | Zastosowanie pętli for do wyprowadzania i obliczania powtarzających się wyrażeń. Tablice jedno, dwu i wielowymiarowe. Tablice dynamiczne. | 6 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-13 | Podział programu na podprogramy, użycie funkcji. Funkcje obsługi zdarzeń, parametry przekazywane przez referencję i przez wartość. Deklaracja i definicja podprogramów, funkcje wbudowane i definiowane. Zmienne lokalne, zmienne globalne. Przekazywanie argumentów do funkcji. | 6 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-14 | Instrukcje iteracyjne w algorytmach sortowania, przeszukiwania i zliczania. Łańcuchy jako przykład typu tablicowego, operacje na łańcuchach, modelowanie tekstów. | 8 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-15 | Zastosowanie wskaźników, adres i zmienna. Operator referencji i dereferencji. Operowanie danymi poprzez wskaźniki. Stos i sarta. Operator new i delete. | 8 | M_01, M_02, M_03, M_04, M_05, M_06 |
| TP-16 | Przykłady wykorzystujące strukturalny jednowymiarowy typ danych – struct – do gromadzenia i przetwarzania bloków danych różnego typu. Standardowe wejście i standardowe wyjście - dostęp do plików. Zapisywanie i odczytywanie struktur danych. | 8 | M_01, M_02, M_03, M_04, M_05, M_06 |
| ZALECANA LITERATURA (w tym pozycje w języku obcym) | | | |

Literatura podstawowa przedmiotu (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. Grębosz J.: Symfonia C++ standard, Edycja 2015
2. Schildt H.: Programowanie C++, Wydawnictwo RM 2002
3. Eckel B.: Thinking in C++ : edycja polska, Helion 2004
4. Prata S.: Język C++: szkoła programowania, Wydawnictwo Robomatic, Wrocław 2002

Literatura uzupełniająca przedmiotu:

1. Kubiak M.: C++. Zadania z programowania z przykładowymi rozwiązaniami. Wydanie III, Helion 2020
2. Wróblewski P.: Algorytmy, struktury danych i techniki programowania: podstawowy podręcznik do nauki algorytmiki , Wyd. 4., Helion, Gliwice 2010
3. Stabrowski M.: Język C++ w przykładach , Wyższa Szkoła Ekonomiczno-Informatyczna w Warszawie, 2005

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA B-LEARNINGU
 część zajęć może być przeprowadzona w sposób hybrydowy

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA E-LEARNINGU
 brak

III. INFORMACJE DODATKOWE

Odniesienie efektów uczenia się przypisanych do zajęć i treści programowych do form zajęć i metod oceniania

Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć.

| Symbol efektu uczenia się przypisanego do zajęć | Symbol treści programowych realizowanych w trakcie zajęć | Formy zajęć i metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się * | Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć # |
|---|--|---|--|
| Wiedza | | | |
| M_01 | TP_01-TP_16 | Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku | Egzamin, kolokwium, zaliczenie projektu praktycznego |
| M_02 | TP_01-TP_16 | Wykład podający, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku | Egzamin, kolokwium, zaliczenie projektu praktycznego |
| Umiejętności | | | |
| M_03 | TP_01-TP_16 | ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku, | Egzamin, kolokwium, zaliczenie projektu praktycznego |
| M_04 | TP_01-TP_16 | ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku, | Egzamin, kolokwium, zaliczenie projektu praktycznego |
| Kompetencje społeczne | | | |

| | | | |
|---|--|--|---------------------------|
| M_05 | TP_01-TP_16 | ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku, | Obserwacja pracy studenta |
| M_06 | TP_01-TP_16 | ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku, | Obserwacja pracy studenta |
| MIARA ŚREDNIEGO NAKŁADU PRACY STUDENTA NIEZBĘDNA DO UZYSKANIA EFEKTÓW UCZENIA SIĘ (godziny) | | | |
| Forma aktywności | | Średnia liczba godzin na zrealizowanie aktywności * | |
| Godziny zajęć (według harmonogramu) z nauczycielem | | 75 | |
| w tym liczba godzin z praktyk zawodowych realizowanych w uczelni (według harmonogramu) | | | |
| Praca własna studenta # | | 125 (przygotowanie do zajęć, czytanie literatury, przygotowanie do egzaminu, wykonanie projektu) | |
| SUMA GODZIN: | | 200 | |
| MIARA ŚREDNIEGO NAKŁADU PRACY STUDENTA NIEZBĘDNA DO UZYSKANIA EFEKTÓW UCZENIA SIĘ (punkty ECTS) | | | |
| | | Liczba punktów ECTS * | |
| SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ | Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim | 7 | 4 |
| | Liczba punktów ECTS przypisana praktykom zawodowym, jeśli formą zajęć dla tego przedmiotu są praktyki zawodowe | | |
| | Praca własna studenta | | 3 |
| * godziny lekcyjne, czyli 1 godz. oznacza 45 min. | | | |
| # przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu,... | | | |
| KRYTERIA OCENIANIA I WYMAGANIA EGZAMINACYJNE | | | |
| Na ocenę dostateczną student ma podstawową wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C/C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji prostego problemu programistycznego w języku C/C++. Student potrafi samodzielnie rozwiązać proste zadania i problemy programistyczne oraz uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego. | | | |
| Na ocenę dobrą student ma zadowalającą wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C/C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji problemu programistycznego w języku C/C++. Student potrafi samodzielnie rozwiązać większość zadań i problemów programistycznych oraz uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego. | | | |
| Na ocenę bardzo dobrą student ma rozszerzoną wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C/C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji problemu programistycznego w języku C/C++. Student potrafi samodzielnie rozwiązać wszystkie zadania czy problemy programistyczne i uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego. | | | |
| Kryteria różnicowania ocen w powiązaniu ze stopniem realizacji efektów uczenia się, muszą być: precyzyjne i czytelne. | | | |

**Podpis nauczyciela akademickiego lub
osoby odpowiedzialnej za przedmiot:**

.....
(imię i nazwisko)

.....
(podpis i data)

Podpis kierownika zakładu:

.....
(imię i nazwisko)

.....
(podpis i data)

Podpis dyrektora instytutu:

.....
(imię i nazwisko)

.....
(podpis i data)