

Karta opisu zajęć - Sylabus

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

I. INFORMACJE PODSTAWOWE

| | |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| Nazwa zajęć: Podstawy programowania | Cykl kształcenia rozpoczynający się w roku akademickim 2023/2024 |
| 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: | Koordinator zajęć Tomasz Lewandowski, dr inż., tomasz.lewandowski@pwste.edu.pl |
| Jednostka organizacyjna: Instytut Inżynierii Technicznej | |

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 zawodowe: | | Praktyki zawodowe: | |
| Inna forma (jaka): | | Inna forma (jaka): | |
| RAZEM: | 75 | RAZEM: | |

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe:

Podstawowe umiejętności matematyczne oraz informatyczne na poziomie szkoły ponadpodstawowej

Cel (cele) kształcenia dla zajęć:

Celem przedmiotu jest przedstawienie podstaw technik programowania opartych na właściwościach języka programowania C++.

EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW

Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się

UWAGA:

Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.

| Symbol efektów uczenia się określonego dla zajęć* | Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii: | | Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się) | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Wiedzy - zna i rozumie | | | | |
| M_01 | podstawowe metody, techniki i narzędzia stosowane w programowaniu imperatywnym | | K_W06 | |
| M_02 | podstawowe pojęcia programistyczne: zadanie algorytmiczne, instrukcja warunkowa, iteracja, rekurencja; budowę programu w języku C++, zakresów dostępności zmiennych, budowę funkcji i sposoby przekazywania parametrów, wykorzystywanie i rola 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_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 | |
| 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. | | | | |
| TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA | | | | |
| Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne): | | | | |
| Symbol treści programowych | Opis treści programowych | Forma zajęć | Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się * | Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć # |
| | | wykład | | |
| TP-01 | Współczesne aspekty programowania. Podstawowe pojęcia. Wybrane środowiska programistyczne. | | wykład podający, wykład problemowy | egzamin |
| TP-02 | Strumienie wejścia/wyjścia. Typy i formaty zmiennych, konwersja typów. Podstawowe konstrukcje programistyczne, operatory. | | wykład podający, wykład problemowy | egzamin |

| | | | | |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|----------------------------------------------------------------------|-----------------------|
| TP-03 | Tworzenie funkcji. Widoczność zmiennych, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje. Funkcje rekurencyjne i biblioteczne. | | wykład podający, wykład problemowy | egzamin |
| TP-04 | Pojęcie wskaźnika, dynamiczne alokowanie i zwolnienie pamięci. Wykorzystanie wskaźników w funkcjach. | | wykład podający, wykład problemowy | egzamin |
| TP-05 | Dynamiczne struktury danych: kolejka, stos, lista. Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Zapis i odczyt do/z pliku. | | wykład podający, wykład problemowy | egzamin |
| | | laboratorium | | |
| TP-06 | Tworzenie, wczytywanie i zapisywanie projektów. Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Pisanie prostych programów, programy z rozgałęzzeniami z zastosowaniem instrukcji warunkowych, programy iteracyjne wykorzystujące pętle. | | objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy | sprawdzian praktyczny |
| TP-07 | Tablice jedno, dwu i wielowymiarowe. Tablice dynamiczne. Struktury, pliki. | | objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy | sprawdzian praktyczny |

| | | | | |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|----------------------------------------------------------------------|-----------------------|
| TP-08 | Podział programu na podprogramy, użycie funkcji, funkcje wbudowane i definiowane. Zmienne lokalne, zmienne globalne. Przekazywanie argumentów do funkcji. | | objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy | sprawdzian praktyczny |
| TP-09 | Zastosowanie wskaźników, adres i zmienna. Operator referencji i dereferencji. Operowanie danymi poprzez wskaźniki. Stos i sterta. Operator new i delete. | | objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy | sprawdzian praktyczny |
| TP-10 | Wykonanie projektu aplikacji wykorzystującej: strumienie wejścia/wyjścia, różne typy i formaty zmiennych, złożone typy danych, konstrukcje programistyczne, operatory, funkcje oraz inne możliwości i techniki poznane w trakcie wykładu i ćwiczeń laboratoryjnych | | objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy | projekt |

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

Dla wykładu:

* 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 określonych dla zajęć.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (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:

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

III. INFORMACJE DODATKOWE**BILANS PUNKTÓW ECTS****OBCIĄŻENIE PRACĄ STUDENTA (godziny)**

| Forma aktywności | Liczba godzin * |
|--------------------------------------------------------------------------------------------------|-----------------|
| Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia | 75 |
| Praca własna studenta | 75 |
| SUMA GODZIN: | 150 |

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

| | | Liczba punktów ECTS | |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|---------------------|---|
| SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ | Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia | Ogółem: 6 | 3 |
| | Praca własna studenta | | 4 |

* godziny lekcyjne, czyli 1 godz. oznacza 45 min;

OPIS PRACY WŁASNEJ STUDENTA:

Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbole efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej. Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.

| | | |
|-----------|----------------------------------------------------------|----------------------------------------|
| M01, M_02 | czytanie wskazanej literatury, przygotowanie do egzaminu | ćwiczenia praktyczne, projekt, egzamin |
| M_03 | opracowanie zadania, projektu | ćwiczenia praktyczne, projekt |
| M_04 | przygotowanie do zajęć | ćwiczenia praktyczne, projekt |
| M_05 | czytanie wskazanej literatury | ćwiczenia praktyczne, projekt |

KRYTERIA OCENIANIA

Ocena kształtująca:

1. Zrozumienie celu zajęć przez studenta.
2. Raportowanie postępu realizacji ćwiczenia
3. Zaangażowanie studenta w wykonywane ćwiczenie
4. Samoocena i ocena koleżeńska.

Ocena podsumowująca:

Na ocenę dostateczną student ma podstawową wiedzę w zakresie zasad tworzenia algorytmów i technik programowania w języku C++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji prostego problemu programistycznego w języku 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++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji problemu programistycznego w języku 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++. Student nabył umiejętność analizy i wytwarzania kodu oraz potrafi dokonać implementacji problemu programistycznego w języku C++. Student potrafi samodzielnie rozwiązać wszystkie zadania czy problemy programistyczne i uruchomić poprawnie wykonany kod źródłowy z użyciem środowiska programistycznego.

**INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA
ODLEGŁOŚĆ**

istnieje

.....
(data, podpis Koordynatora
odpowiedzialnego za zajęcia)

.....
(data, podpis Kierownika Zakładu/
Kierownika Jednostki Międzyinstytutowej)

Uwaga:
Karta opisu zajęć (syllabus) musi być dostępna dla studenta.