|  |
| --- |
| Sylabus przedmiotu / modułu kształcenia |
| Nazwa przedmiotu/modułu kształcenia:  | Wstęp do programowania |
| Nazwa w języku angielskim:  | Programming  |
| Język wykładowy:  | polski  |
| Kierunek studiów, dla którego przedmiot jest oferowany:  | Inżynieria procesów technologicznych  |
| Jednostka realizująca:  | Wydział Nauk Ścisłych i Przyrodniczych  |
| Rodzaj przedmiotu/modułu kształcenia (obowiązkowy/fakultatywny):  | obowiązkowy  |
| Poziom modułu kształcenia (np. pierwszego lub drugiego stopnia):  | pierwszego stopnia  |
| Rok studiów:  | pierwszy  |
| Semestr:  | pierwszy |
| Liczba punktów ECTS:  | 4  |
| Imię i nazwisko koordynatora przedmiotu:  | Mirosław Barański  |
| Imię i nazwisko prowadzących zajęcia:  | Mirosław Barański  |
| Założenia i cele przedmiotu:  | Założono, że studenci poznają w podstawowym zakresie zasady programowania i będą potrafić posługiwać się wybranym środowiskiem programistycznym. Celem kursu jest opanowanie przez studentów podstawowej wiedzy z podstaw programowania: student powinien poznać wybrane środowisko programistyczne, opanować podstawowe (elementarne) algorytmy oraz podstawowe konstrukcje programistyczne związane z programowaniem imperatywnym i strukturalnym i obiektowym oraz powinien umieć korzystać z funkcji oraz bibliotek. Celem zajęć jest także nauczenie studenta zaprojektowania i implementacji prostego systemu informatycznego w języku C/C++.  |
| Symbol efektu  | Efekt uczenia się: WIEDZA  | Symbol efektu kierunkowego  |
| W\_01  | Student ma podstawową wiedze na temat algorytmów, ich własności oraz zna etapy rozwiązywania zadań  | K\_W07  |
| W\_02  | Ma uporządkowaną wiedzę z zakresu podstawowych konstrukcji języka C/C++  | K\_W07  |
| W\_03  | Ma uporządkowaną wiedzę z zakresu typów danych w języku C/C++ oraz wybranych standardów zapisu tych liczb  | K\_W07  |
| W\_04  | Ma uporządkowana wiedzę na temat wykorzystania funkcji w językach programowania  | K\_W07  |
| W\_05  | Ma uporządkowaną wiedze z zakresu rozwiązywania problemów za pomocą metody zstępującej i wstępującej  | K\_W07  |
| W\_06  | Ma podstawową wiedzę na temat zasad, technik i metod związanych z programowaniem za pomocą paradygmaty programowania obiektowego.  | K\_W07  |
| Symbol efektu  | Efekt uczenia się: UMIEJĘTNOŚCI  | Symbol efektu kierunkowego  |
| U\_01  | Potrafi na podstawie literatury formułować wnioski dotyczące najnowszych rozwiązań systemów komputerowych i wskazywać sposób ich powiązania z ogólnymi zasadami realizacji pracy systemu.  | K\_U01  |
| U\_02  | Posiada umiejętność samokształcenia się, m.in. w celu podnoszenia kompetencji zawodowych  | K\_U06  |
| U\_03  | Umie implementować proste algorytmy w języku C/C++ oraz dobrać odpowiednie struktury danych do rozwiązywanego problemu  | K\_U19  |
| U\_04  | Umie weryfikować poprawność napisanego programu, potrafi dobrać odpowiednie dane testowe  | K\_U19  |
| U\_05  | Umie rozwiązywać proste problemy algorytmiczne za pomocą języka C/C++  | K\_U19  |
| U\_06  | Potrafi ocenić przydatność rutynowych metod i narzędzi służących do rozwiązywania prostych zadań inżynierskich, typowych dla informatyki oraz wybierać i stosować właściwe metody i narzędzia  | K\_U19  |
| Symbol efektu  | Efekt uczenia się: KOMPETENCJE SPOŁECZNE  | Symbol efektu kierunkowego  |
| K\_01  | Jest gotów do podejmowania decyzji i krytycznej oceny własnych rozwiązań w rozwiązywaniu zadań programistycznych  | K\_K03  |
| K\_02  | Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów programistycznych oraz krytycznie potrafi ocenić swoje działania  | K\_K03  |
| Forma i typy zajęć:  | wykłady (30 godzin), ćwiczenia laboratoryjne (30 godzin)  |
| Wymagania wstępne i dodatkowe:  |
| Znajomość programowania i matematyki na poziomie szkoły średniej |
| Treści modułu kształcenia:  |
| 1. Wstęp do programowania. Pojęcia podstawowe. Fazy powstawania programu (koncepcja, algorytm, kodowanie). Jednostki leksykalne. Struktura programu.
2. Generacje języków programowania. Języki maszynowe i asemblerowe. Języki wyższego poziomu. Języki

4-ej generacji. Języki sztucznej inteligencji. Historia rozwoju języka C++ 1. Typy danych i zmienne. Standardowe typy danych. Zmienne i ich deklaracje. Wyrażenia. Operatory.

Konwersje. Wyrażenia arytmetyczne i logiczne. Wskaźniki. Definicja wskaźnika. Zmienne wskaźnikowe i wskazywane. Tablice a wskaźniki. Przekazywanie parametrów do funkcji przez wskaźnik. Wskaźniki do funkcji i struktur (unii). Wskaźniki a referencje. 1. Instrukcje. Instrukcje decyzyjne. Instrukcje iteracyjne. Instrukcja wyboru. Instrukcje sterujące. Instrukcja grupująca.
2. Statyczne struktury danych. Tablice, tablice dynamiczne. Struktury.
3. Funkcje Definicje funkcji. Specyfikatory funkcji. Parametry funkcji. Funkcje biblioteczne. Zasięg i widoczność zmiennych w programie. Zmienne statyczne. Zmienne automatyczne.
4. Metoda rekurencyjna w programowaniu. Definicja rekurencji. Rozwiązywanie problemów programistycznych metodą rekurencyjna. Metody wstępujaca i zstępujaca w programowaniu strukturalnym.
5. Wprowadzenie do obiektowych struktur danych. Klasy i obiekty. Elementy analizy obiektowej. Klasy.

Obiekty. 1. Elementy programowania obiektowego, dziedziczenie. Enkapsulacja, konstruktory, destruktory, funkcje przeciążone.
2. Pliki. Podejście proceduralne do przetwarzania plików. Algorytm przetwarzania operacji wejścia - wyjścia.
 |
| Literatura podstawowa:  |
| 1. Jerzy Grebosz - Symfonia C++ standard : programowanie w języku C++ zorientowane obiektowo. T. 1, Wydawnictwo "Edition 2000" : Oficyna Kallimach, rok 2010.
2. Stephen Prata - Język C++ : szkoła programowania, Wydawnictwo Wrocław : "Robomatic", rok 2012.
3. J.Liberty, C++ dla każdego, Helion, 2002.
 |
| Literatura dodatkowa:  |
| 1. W.M. Turski, Metodologia programowania, WNT, Warszawa 1982.
2. Bjorne Stroustrup - Język C++ ; WNT 2002.
3. L.Ullman, A.Signer, Programowanie w języku C++. Szybki start, Helion, 2006 4. N. Wirth, Wstęp do programowania systematycznego, WNT, Warszawa 1987.
4. A.Alagic, M.A.Arbib, Projektowanie programów poprawnych i dobrze zbudowanych, WNT 1982,
5. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Projektowanie i analiza algorytmów komputerowych, PWN, Warszawa 1983.
6. Andrew Koenig, Barbara E. Moo - C++. Potęga języka. Od przykładu do przykładu, Helion 2004.
 |
| Planowane formy/działania/metody dydaktyczne:  |
| Wykład tradycyjny wspomagany technikami multimedialnymi, ćwiczenia laboratoryjne wspomagane technikami multimedialnymi. Zamieszczanie na stronach internetowych zadań laboratoryjnych.  |
| Sposoby weryfikacji efektów uczenia się osiąganych przez studenta:  |
| Efekty W\_01 – W\_06 będą sprawdzane na egzaminie pisemnym i ustnym. Na egzaminie pisemnym zadania będą dotyczyły wybranych problemów algorytmicznych i typów danych, przykładowe zadania: * Dany jest ciąg n-elementowy liczb rzeczywistych. Napisz program, który znajdzie k-tą największą liczbę w tym ciągu. Liczby tworzące ciąg i liczba k są wczytywane z klawiatury,
* Dana jest tablica liczb rzeczywistych. Napisz program wypisujący k-liczb znajdujących się najbliżej mediany.
* Dany jest ciąg liczb całkowitych zapisany w pliku binarnym. Napisz program, który zapisze do pliku tekstowego te liczby, które spełniają warunek: suma cyfr jest równa iloczynowi cyfr.

Na egzaminie ustnym student będzie odpowiadał na pytania dotyczące metod i technik programowania, przykładowe pytania: * Omów metodę zstępująca. Podaj przykłady jej stosowania,
* Omów typ tablicowy. Metody inicjowania tablic,
* Na czym polega przeciążanie funkcji, kiedy je stosujemy. Podaj przykłady.

Przed egzaminem studenci będą mieli dostęp do pełnej listy pytań na egzamin ustny oraz do przykładowych zadań na egzamin pisemny. Efekt U\_01 - U\_06 będą systematycznie sprawdzane na zajęciach. Zadania na następne laboratorium muszą być dostępne co najmniej tydzień przed zajęciami. Student, na podstawie literatury lub źródeł internetowych, musi się do nich samodzielnie lub korzystając z konsultacji przygotować, przykładowe zadanie: * Dana jest lista osób o strukturze z poprzedniego zadania zapisana w tablicy. Napisz program obliczający: sumę brutto, osoby o maksymalnym, osoby o minimalnym brutto, osoby mieszkające w Siedlcach oraz osoby palące. Wypisz poszczególne wyniki na standardowym urządzeniu wyjścia, sprawdź działanie programu. Efekt U\_06 będzie sprawdzany na zajęciach, przykładowe zadanie:
* Napisz program obliczający iloczyn dwu liczb całkowitych nie używając operacji mnożenia. Następnie w środowisku CodeBlocks (Dev, Visual C++) wykonaj krok po kroku program dla przykładowych danych. W trakcie wykonania śledź wartość wybranej zmiennej.

Efekty K\_01, K\_02 będą weryfikowane, w oparciu o posiadaną wiedzę i umiejętności w czasie zajęć laboratoryjnych oraz podczas zaliczania zadania indywidualnego. Przykładowe zadanie zależy od tematu zadania indywidualnego: * Wymień funkcjonalności swojego zadania indywidualnego. Uzasadnij wybór.
 |
| Forma i warunki zaliczenia:  |
| Moduł kończy się egzaminem. Do egzaminu mogą przystąpić osoby, które uzyskały zaliczenie laboratorium. Na zaliczenie laboratorium składają się oceny cząstkowe uzyskane na regularnych zajęciach z nauczycielem akademickim oraz z samodzielnie wykonanego i ocenionego zadania indywidualnego według schematu: * Regularne zajęcia – 26 pkt.,
* Obrona zadania indywidualnego – 14 pkt.

Zajęcia laboratoryjne będą zaliczone w wypadku uzyskania co najmniej połowy punktów z poszczególnych form aktywności studenta: regularne zajęcia – co najmniej 13 pkt., obrona indywidualnego zadania – co najmniej 7 pkt. Na tej formie zajęć student może maksymalnie uzyskać 40 pkt. Każde ćwiczenie laboratoryjne musi być zaliczone na co najmniej połowę punktów. Egzamin jest egzaminem pisemnym. Na egzaminie pisemnym można uzyskać do 60 pkt. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 30 pkt. w części pisemnej. Ocena końcowa z modułu (wystawiana po zaliczeniu wszystkich części składowych), w zależności od sumy uzyskanych punktów (maksymalnie 100pkt.) jest następująca (w nawiasach ocena wg skali ECTS):* 0 – 50 pkt: niedostateczna (F),
* 51 – 60 pkt: dostateczna (E),
* 61 – 70 pkt: dostateczna plus (D),
* 71 – 80 pkt: dobra (C),  81 – 90 pkt: dobra plus (B),
* 91 – 100 pkt: bardzo dobra (A).

Poprawy: Jednorazowa poprawa każdego laboratorium w trakcie trwania semestru, przy czym laboratorium można poprawiać w ciągu miesiąca od daty jego odbycia: obecność usprawiedliwiona – maksymalnie 10pkt. nieusprawiedliwiona – maksymalnie 8 pkt. Termin poprawy należy uzgodnić z prowadzącą zajęcia laboratoryjne osobą. Poprawy wybranych laboratoriów w sesji egzaminacyjnej, odpowiednio przed drugim i trzecim terminem egzaminu pisemnego. Uwaga: Istnieje możliwość zwolnienia z egzaminu pisemnego lub ustnego studentów wyróżniających się na zajęciach laboratoryjnych. Warunkiem koniecznym zwolnienia z egzaminu jest uzyskanie co najmniej 95% punktów możliwych do zdobycia w trakcie regularnych zajęć laboratoryjnych łącznie z zadaniem indywidualnym. Decyzję o ewentualnym zwolnieniu podejmuje osoba przeprowadzająca egzamin po zasięgnięciu opinii (poprzez rozmowę) osób prowadzących zajęcia. Decyzję o zwolnieniu prowadzący wykład przekazuje studentom nie później niż 2 tygodnie przed końcem semestru.  |
| Bilans punktów ECTS:  |
| Studia stacjonarne  |
| Aktywność  | Obciążenie studenta  |
| Udział w wykładach  | 30 godzin  |
| Udział w ćwiczeniach laboratoryjnych  | 30 godzin  |
| Udział w konsultacjach z przedmiotu  | 5 godzin  |
| Samodzielne przygotowanie się do ćwiczeń  | 20 godzin  |
| Przygotowanie się do egzaminu i obecność na nim  | 15 godzin  |
| **Sumaryczne obciążenie pracą studenta**  | **100 godzin**  |
| **Punkty ECTS za przedmiot**  | **4 ECTS**  |