| Nazwa w języku angielskim:  |  Programming fundamentals  |
| --- | --- |
| **Język wykładowy:**  | polski  |
| **Kierunek studiów, dla którego przedmiot jest oferowany**:  |  informatyka  |
| **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:**  |  **5**  |
| **Imię i nazwisko koordynatora przedmiotu:**  | Mirosław Barański  |
| **Imię i nazwisko prowadzących zajęcia:**  | Mirosław Barański Michał Seredyński  |
| **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 pozna wybrane środowisko programistyczne, opanuje podstawowe (elementarne) algorytmy oraz podstawowe konstrukcje programistyczne związane z programowaniem imperatywnym, strukturalnym i proceduralnym oraz będzie umiał 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**  | Zna i rozumie zagadnienia związane z algorytmami, ich własności oraz zna etapy rozwiązywania zadań  | **K\_W06**  |
| **W\_02**  | Zna i rozumie zagadnienia z zakresu podstawowych konstrukcji języka C/C++  | **K\_W06**  |
| **W\_03**  | Zna i rozumie zagadnienia związane z typami standardowymi oraz złożonymi (tablice, struktury, pliki) w języku C/C++ oraz wybranych standardów zapisu tych liczb  | **K\_W06, K\_W09**  |
| **W\_04**  | Zna i rozumie zagadnienia z zakresu wykorzystania funkcji w języku C/C++  | **K\_W06**  |
| **W\_05**  | Zna i rozumie zagadnienia z zakresu rozwiązywania problemów za pomocą metody zstępującej i wstępującej  | **K\_W06**  |
| **W\_06**  | Zna i rozumie zagadnienia z zakresu rekurencji i jej implementacji w języku C/C++  | **K\_W06**  |
| **Symbol efektu**  | **Efekt uczenia się: UMIEJĘTNOŚCI**  | **Symbol efektu kierunkowego**  |
| **U\_01**  | Potrafi, na podstawie literatury, formułować wnioski dotyczące najnowszych rozwiązań problemów obejmujących zagadnienia programowania.  | **K\_U01**  |
| **U\_02**  | Potrafi samodzielnie planować i realizować uczenie się przez całe życie w celu poznawania nowych trendów w programowaniu  | **K\_U06**  |
| **U\_03**  | Potrafi implementować proste algorytmy w języku C/C++ oraz dobrać odpowiednie struktury danych do rozwiązywanego problemu  | **K\_U22**  |
| **U\_04**  | Potrafi weryfikować poprawność napisanego programu, potrafi dobrać odpowiednie dane testowe  | **K\_U12**  |
| **U\_05**  | Potrafi rozwiązywać proste problemy algorytmiczne za pomocą języka C/C++  | **K\_U22**  |
| **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\_U10**  |
| **Symbol efektu**  | **Efekt uczenia się: KOMPETENCJE SPOŁECZNE**  | **Symbol efektu kierunkowego**  |
| **K\_01**  | Jest gotów do krytycznej oceny własnych rozwiązań w analizie zadań programistycznych  | **K\_K01**  |
| **K\_02**  | Jest gotów do uznania znaczenia wiedzy w rozwiązywaniu problemów programistycznych oraz do jej krytycznej oceny  | **K\_K01**  |
| **Forma i typy zajęć:**  | studia stacjonarne: wykłady (30 godz.), ćwiczenia laboratoryjne (30 godz.) studia niestacjonarne: wykłady (15 godz.), ćwiczenia laboratoryjne (18 godz.)  |
| **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). Metody zapisu algorytmów: pseudokod, standard w schematach blokowych.
2. Generacje języków programowania. Języki maszynowe i asemblerowe. Języki wyższego poziomu. Języki 4-ej generacji. Języki sztucznej inteligencji. Krótka charakterystyka wybranych języków programowania. Historia rozwoju języka C++. Jednostki leksykalne. Struktura programu w C++.
3. Typy danych i zmienne I. Standardowe typy danych: całkowito-liczbowe. Wybrane standardy dla liczb całkowitych. Zmienne i ich deklaracje. Własności zmiennych.
4. Typy danych i zmienne II. Typy zmiennoprzecinkowe. Wybrany standard dla liczb rzeczywistych. Typy wyliczeniowe.
5. Instrukcje. Instrukcje decyzyjne. Instrukcje iteracyjne. Instrukcja wyboru. Instrukcje sterujące. Instrukcja grupująca.
6. Wskaźniki. Definicja wskaźnika. Zmienne wskaźnikowe i wskazywane. Wskaźniki a referencje. Zmienne dynamiczne. Przydzielanie i zwalnianie pamięci. Zagrożenia wynikające ze stosowania zmiennych dynamicznych. Wprowadzenie do dynamicznych struktur danych
7. Wyrażenia. Operatory. Priorytety. Konwersje. Wyrażenia arytmetyczne i logiczne.
8. Złożone typy danych. Tablice statyczne i dynamiczne. Tablice a wskaźniki. Struktury. Unie. Wskaźniki do struktur (unii).
9. Funkcje Definicje funkcji. Specyfikatory funkcji. Argumenty funkcji. Funkcje biblioteczne. Wskaźniki do funkcji.
10. Zasięg i widoczność zmiennych w programie. Rodzaje zmiennych. Zasięg zmiennych. Przekazywanie
11. danych w funkcjach: przez zmienne globalne, przez wartość, wskaźnik i referencję. Przekazywanie typów prostych i złożonych (liczby, tablice, struktury).
12. Metoda rekurencyjna w programowaniu. Definicja rekurencji. Rozwiązywanie problemów programistycznych metodą rekurencyjną.
13. Metody wstępująca i zstępująca w programowaniu strukturalnym.
14. Wprowadzenie do obiektowych struktur danych. Klasy i obiekty. Elementy programowania obiektowego. Klasy. Obiekty. Funkcje przeciążone.
15. Pliki. Podejście obiektowe do przetwarzania plików. Algorytm przetwarzania operacji wejścia - wyjścia.
16. Biblioteki w programowaniu. Krótki opis wybranych bibliotek. Przykłady – funkcje matematyczne, funkcje operujące na napisach, funkcje konwersji.
 |
| 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. Robert C. Martin, Czysty kod. Podręcznik dobrego programisty. Helion 2014.
2. W.M. Turski, Metodologia programowania, WNT, Warszawa 1982.
3. Bjorne Stroustrup - Język C++ ; WNT 2002.
4. N. Wirth, Wstęp do programowania systematycznego, WNT, Warszawa 1987.
5. A.Alagic, M.A.Arbib, Projektowanie programów poprawnych i dobrze zbudowanych, WNT 1982,
6. A.V. Aho, J.E. Hopcroft, J.D. Ullman, Projektowanie i analiza algorytmów komputerowych, PWN, Warszawa 1983.
7. 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 problemów i zadań laboratoryjnych.  |
| **Sposoby weryfikacji efektów kształcenia 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. Użyj rekurencji.

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\_02 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. Efekt U\_03 – U\_06 będą sprawdzana systematycznie na zajęciach laboratoryjnych, 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, podczas zaliczania zadania indywidualnego, a także będą sprawdzane na egzaminie ustnym. 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 – powyżej 13 pkt., obrona indywidualnego zadania – powyżej 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 i ustnym. Na egzaminie pisemnym można uzyskać do 40 pkt. a na pisemnym do 20pkt.. Egzamin będzie zaliczony w przypadku uzyskania co najmniej 21 pkt. w części pisemnej i 11 pkt. w części ustnej. 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 osobą prowadzącą zajęcia laboratoryjne. 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 96% 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 godz.  |
| Udział w ćwiczeniach laboratoryjnych  | 30 godz.  |
| Samodzielne przygotowanie się do ćwiczeń laboratoryjnych  | 35 godz.  |
| Udział w konsultacjach godz. z przedmiotu  | 8 godz.  |
| Przygotowanie się do egzaminu i obecność na egzaminie  | 22 godz.  |
| **Sumaryczne obciążenie pracą studenta**  | **125 godz.**  |
| **Punkty ECTS za przedmiot**  | **5 ECTS**  |
| **Studia niestacjonarne**  |
| **Aktywność**  | **Obciążenie studenta**  |
| Udział w wykładach  | 15 godz.  |
| Udział w ćwiczeniach laboratoryjnych  | 18 godz.  |
| Samodzielne przygotowanie się do ćwiczeń laboratoryjnych  | 55 godz.  |
| Udział w konsultacjach godz. z przedmiotu  | 7 godz.  |
| Przygotowanie się do egzaminu i obecność na egzaminie  | 30 godz.  |
| **Sumaryczne obciążenie pracą studenta**  | **125 godz.**  |
| **Punkty ECTS za przedmiot**  | **5 ECTS**  |