

# PODSTAWY PROGRAMOWANIA

## Opis przedmiotu

### INFORMACJE OGÓLNE

1. Nazwa przedmiotu: Podstawy programowania
2. Kod przedmiotu: DPPR LI0
3. Semestr: zimowy
4. Liczba godzin: 30 – wykłady, 30 – laboratoria
5. ECTS: 6

### INFORMACJE SZCZEGÓLNE

1. Celem przedmiotu jest prezentacja podstaw programowania proceduralnego na przykładzie języków ANSI C i Python. Omawiany jest proces tworzenia programu, znaczenie algorytmu, języka, kompilatora i interpretera. Przedstawia się metody sterowanie przebiegiem programu, znaczenie pojęcia typu danych, podstawowe typy i ich reprezentację w pamięci. Omawiane są złożone typy danych, wyrażenia, zmienne statyczne, wskaźniki, zmienne dynamiczne oraz zagadnia dynamicznego zarządzania pamięcią.

### SYLABUS

#### Wykłady:

1. Tworzenie oprogramowania. Algorytm, procesor, program, język programowania. Alfabet, składnia i semantyka. Język maszynowy oraz języki wyższego rzędu. Kompilacja, interpretacja i konsolidacja programu. Zmienne, typy, pierwszy program.
2. Sterowanie przebiegiem programu. Prawda i fałsz, wstępne uwagi o wyrażeniach logicznych. Instrukcje złożone w instrukcjach sterujących. Zakresy ważności nazw, czas życia obiektów, przesłanianie nazw. Instrukcja warunkowa. Pętle. Przerwa i kontynuacja wykonywania pętli. Instrukcja skoku. Instrukcja wyboru.
3. Podstawowe typy danych. Reprezentacja danych liczbowych i znakowych. Pojęcie typu. Rola typu w procesie tworzenia programu, stałe i zmienne. Znaczenie typu w procesie kompilacji. Typy liczbowe. Typy znakowe. Definiowanie stałych różnych typów. Konwersje typów, rzutowania.
4. Zmienne, operatory i wyrażenia. Definiowanie zmiennych. Operatory. Priorytety operatorów. Kolejność wykonywania działań.
5. Złożone typy danych. Tablice, łańcuchy, rekordy (struktury) i unie. Inicjowanie zmiennych złożonych. Formatowanie wejścia/wyjścia.
6. Procedury i funkcje. Pojęcie funkcji; przekazywanie parametrów i zwracanie wyniku. Prototyp funkcji (typ funkcyjny). Czas życia i zakres ważności nazwy. Ukrywanie informacji. Funkcje biblioteczne.
7. Przeciążanie operatorów i funkcji. Zagadnienie identyczności typów argumentów. Dopasowanie argumentów do funkcji przeciążonych.
8. Zarządzanie pamięcią. Wskaźniki, referencje i dereferencje. Dynamiczna alokacja pamięci, sterta. Wskaźniki a tablice. Wskaźniki w zastosowaniu do przekazywania parametrów. Wskaźniki do funkcji. Wskaźniki bez typu.
9. Operacje wejścia/wyjścia. Operacje wejścia/wyjścia w C (biblioteka studio). Operacje wejścia/wyjścia w C++. Biblioteka iostream; strumień; strumienie predefiniowane. Sterowanie formatem. Operacje wejścia/wyjścia na plikach. Obsługa błędów.
10. Python. Specyficzne własności języka Python. Różnice w składni operacji podstawowych w języku Python w stosunku do języka ANSI C
11. Moduły. Moduł NumPy.

#### Laboratoria:

1. Tworzenie i uruchamianie prostych programów

2. Sterowanie przebiegiem programu (ANSI C) – implementacja
3. Reprezentacja danych liczbowych (ANSI C) – implementacja
4. Operatory (ANSI C) - implementacja
5. Złożone typy danych (ANSI C) – implementacja
6. Procedury i funkcje (ANSI C) – implementacja
7. Wskaźniki (ANSI C) – implementacja
8. Operacje wejścia, wyjścia (ANSI C) – implementacja
9. Tryb interaktywny. Powłoka Idle. Uruchamianie skryptów języka Python
10. Tworzenie i uruchamianie prostych programów (Python)
11. Klasy - implementacja (Python)
12. Tworzenie własnych modułów. Korzystanie z modułu NumPy (Python)