



Nazwa modułu: Programowanie w języku Python

Rok akademicki: 2015/2016 Kod: BIT-1-505-s Punkty ECTS: 4

Wydział: Geologii, Geofizyki i Ochrony Środowiska

Kierunek: Informatyka Stosowana Specjalność: —

Poziom studiów: Studia I stopnia Forma i tryb studiów: Stacjonarne

Język wykładowy: Polski Profil kształcenia: Ogólnoakademicki (A) Semestr: 5

Strona www: —

Osoba odpowiedzialna: dr Oleksik Paweł (oleksik@agh.edu.pl)

Osoby prowadzące: dr Oleksik Paweł (oleksik@agh.edu.pl)

Opis efektów kształcenia dla modułu zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Powiązania z EKK	Sposób weryfikacji efektów kształcenia (forma zaliczeń)
Wiedza			
M_W001	Zna podstawową składnię języka i jej semantykę.	IT1A_W07, IT1A_W06, IT1A_W11	Wykonanie projektu
M_W002	Rozumie specyfikę modelu obiektowego realizowanego w Pythonie.	IT1A_W07	Wykonanie projektu
M_W003	Rozumie właściwości dynamicznej typizacji i dynamicznej semantyki.	IT1A_W07, IT1A_W06, IT1A_W04	Wykonanie projektu
M_W004	Zna hierarchię wbudowanych typów danych oraz ich właściwości.	IT1A_W07, IT1A_W04	Wykonanie projektu
Umiejętności			
M_U001	Umie zapisać algorytmy za pomocą składni języka Python.	IT1A_U12, IT1A_U16, IT1A_U13	Wykonanie projektu
M_U002	Potrafi zastosować obsługę wyjątków do sterowania programem.	IT1A_U12, IT1A_U16, IT1A_U13	Wykonanie projektu
M_U003	Umie definiować i używać funkcje z parametrami w dowolnym wariacie (zwykłe, z domyślną wartością, opcjonalne i nazwane).	IT1A_U12, IT1A_U16, IT1A_U13	Wykonanie projektu
M_U004	Umie definiować i wykorzystywać moduły oraz paczki.	IT1A_U12, IT1A_U16, IT1A_U13	Wykonanie projektu

M_U005	Umie definiować klasy oraz stosować podstawowe profilowanie z wykorzystaniem metod specjalnych.	IT1A_U12, IT1A_U16, IT1A_U13	Wykonanie projektu
M_U006	Umie zastosować elementy programowania funkcyjnego.	IT1A_U12, IT1A_U16, IT1A_U13	Wykonanie projektu
Kompetencje społeczne			
M_K001	Student ma świadomość odpowiedzialności za własną pracę	IT1A_K03	Wykonanie projektu

Matryca efektów kształcenia w odniesieniu do form zajęć

Kod EKM	Student, który zaliczył moduł zajęć wie/umie/potrafi	Forma zajęć										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatori um	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W001	Zna podstawową składnię języka i jej semantykę.	+	-	-	-	-	-	+	-	-	-	-
M_W002	Rozumie specyfikę modelu obiektowego realizowanego w Pythonie.	+	-	-	-	-	-	+	-	-	-	-
M_W003	Rozumie właściwości dynamicznej typizacji i dynamicznej semantyki.	+	-	-	-	-	-	+	-	-	-	-
M_W004	Zna hierarchię wbudowanych typów danych oraz ich właściwości.	+	-	-	-	-	-	+	-	-	-	-
Umiejętności												
M_U001	Umie zapisać algorytmy za pomocą składni języka Python.	+	-	-	-	-	-	+	-	-	-	-
M_U002	Potrafi zastosować obsługę wyjątków do sterowania programem.	+	-	-	-	-	-	+	-	-	-	-
M_U003	Umie definiować i używać funkcje z parametrami w dowolnym wariacie (zwykle, z domyślną wartością, opcjonalne i nazwane).	+	-	-	-	-	-	+	-	-	-	-
M_U004	Umie definiować i wykorzystywać moduły oraz paczki.	+	-	-	-	-	-	+	-	-	-	-
M_U005	Umie definiować klasy oraz stosować podstawowe profilowanie z wykorzystaniem metod specjalnych.	+	-	-	-	-	-	+	-	-	-	-

M_U006	Umie zastosować elementy programowania funkcyjnego.	+	-	-	-	-	-	-	+	-	-	-	-
Kompetencje społeczne													
M_K001	Student ma świadomość odpowiedzialności za własną pracę	-	-	-	-	-	-	-	+	-	-	-	-

Treść modułu zajęć (program wykładów i pozostałych zajęć)

Wykład

Wstęp.

- Używanie standardowego interpretera.
- Podstawowe instrukcje i typy danych.
- Biblioteka standardowa.

Wersje języka i interpreterów.

- Schemat działania standardowego interpretera.

Model danych.

- Pełna obiektowość, możliwości introspekcji.
- Przegląd standardowych typów danych.

Model obliczeń.

- Zmienne, semantyka instrukcji podstawienia, garbage collector.
- Dynamiczna, silna typizacja; duck typing.
- Zakresy widoczności symboli.
- Dynamiczna semantyka.

Podstawy programowania obiektowego.

- Model obiektowy: klasy, instancje, modyfikowanie instancji w locie.
- Metody instancji, metody statyczne i metody klasy.
- Metody specjalne: realizowanie protokołów.
- Dziedziczenie jednobazowe.

Zaawansowane konstrukcje.

- Obsługa wyjątków.
- Wyrażenia generujące listy i słowniki.
- Proste generatory.

Rozszerzone możliwości programowania obiektowego.

- Kontrolowanie dostępu do atrybutów.
- Dekoratory.
- Dziedziczenie wielobazowe.
- Elementy metaprogramowania (metaklasy).

Elementy programowania funkcyjnego.

Tworzenie bibliotek.

Zajęcia praktyczne

Praktyczne wprowadzenie.

- Standardowy interpreter, środowisko programistyczne i programy pomocnicze.
- Zastosowanie podstawowych konstrukcji, typów danych oraz wybranych modułów z

biblioteki standardowej.

Wykorzystanie zaawansowanych konstrukcji.

- Wyjątki.
- Proste generatory i wyrażenia generatorowe.
- Sekcje dozorowane.

Podstawy programowania obiektowego.

- Właściwości klas i instancji; zastosowanie introspekcji.
- Metody związane i niezwiązane; metody statyczne.
- Dziedziczenie jednobazowe.

Metody specjalne klas.

- Profilowanie działania instancji; implementacja wybranych protokołów.
- Metody klas.

Metody specjalne klas - zaawansowane.

- Deskryptory.
- Kontrola dostępu do atrybutów.

Programowanie obiektowe - zaawansowane.

- Dziedziczenie wielobazowe w modelach 'domieszkiwania' i 'kooperacji'.
- Tworzenie dekoratorów.
- Wstęp do metaklas.

Tworzenie bibliotek.

- Moduły i paczki w języku Python.
- Moduły binarne (w języku maszynowym).

Programowanie funkcyjne i dynamiczna semantyka.

Tworzenie prostego GUI (Tkinter).

Sposób obliczania oceny końcowej

Ocena końcowa odpowiada ocenie z zaliczenia.

Wymagania wstępne i dodatkowe

Umiejętność programowania proceduralnego i obiektowego.

Zalecana literatura i pomoce naukowe

Nie podano zalecanej literatury lub pomocy naukowych.

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Nie podano dodatkowych publikacji

Informacje dodatkowe

Podstawowym terminem uzyskania zaliczenia jest koniec zajęć w danym semestrze. Student może dwukrotnie przystąpić do poprawkowego zaliczenia.

udział „praktycznych” punktów ECTS: 3

udział „teoretycznych” punktów ECTS: 1

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w wykładach	15 godz
Samodzielne studiowanie tematyki zajęć	28 godz
Udział w zajęciach praktycznych	30 godz
Przygotowanie do zajęć	40 godz
Sumaryczne obciążenie pracą studenta	113 godz
Punkty ECTS za moduł	4 ECTS