

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu: Programowanie dla naukowców i inżynierów

Rok akademicki: 2016/2017 Kod: ZIPM-3-004-s Punkty ECTS: 3

Wydział: Zarządzania

Kierunek: Inżynieria Produkcji Metali Nieżelaznych Specjalność: —

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

Język wykładowy: Polski Profil kształcenia: Ogólnoakademicki (A) Semestr: 0

Strona www: —

Osoba odpowiedzialna: dr hab. inż. Kaczmarczyk Waldemar (wkaczmar@zarz.agh.edu.pl)

Osoby prowadzące: dr hab. inż. Kaczmarczyk Waldemar (wkaczmar@zarz.agh.edu.pl)
dr inż. Opiła Janusz (jmo@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 nowoczesne metody i narzędzia programistyczne wykorzystywane przy prowadzeniu badań naukowych.	IPM3A_W01, IPM3A_W03	Egzamin, Projekt inżynierski
Umiejętności			
M_U001	Potrafi samodzielnie odszukać, opanować i wykorzystać zaawansowane narzędzia programistyczne przydatne dla jego pracy badawczej.	IPM3A_U05	Egzamin, Projekt inżynierski
M_U002	Potrafi planować projekty programistyczne na potrzeby badań naukowych.	IPM3A_U01	Egzamin, Projekt inżynierski
Kompetencje społeczne			
M_K001	Rozumie potrzebę tworzenia otwartego oprogramowania, w szczególności oprogramowania wykorzystywanego w badaniach naukowych.	IPM3A_K02, IPM3A_K03	Projekt, Udział w dyskusji

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	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Inne	E-learning
Wiedza												
M_W001	Zna nowoczesne metody i narzędzia programistyczne wykorzystywane przy prowadzeniu badań naukowych.	-	-	-	-	+	-	-	-	-	-	-
Umiejętności												
M_U001	Potrafi samodzielnie odszukać, opanować i wykorzystać zaawansowane narzędzia programistyczne przydatne dla jego pracy badawczej.	-	-	-	-	+	-	-	-	-	-	-
M_U002	Potrafi planować projekty programistyczne na potrzeby badań naukowych.	-	-	-	-	+	-	-	-	-	-	-
Kompetencje społeczne												
M_K001	Rozumie potrzebę tworzenia otwartego oprogramowania, w szczególności oprogramowania wykorzystywanego w badaniach naukowych.	-	-	-	-	+	-	-	-	-	-	-

Treść modułu zajęć (program wykładów i pozostałych zajęć)

Konwersatorium

W ramach wykładu przedstawione zostaną wygodne narzędzia i metody umożliwiające łatwe i szybkie napisanie własnych programów (skryptów) potrzebnych do badań naukowych.

Wykorzystany zostanie język Python wraz z licznymi dodatkowymi pakietami ułatwiającymi wykonywanie obliczeń naukowych, tworzenie grafik, symulację czy optymalizację.

1. Programowanie proceduralne
2. Programowanie obiektowe
3. Zalety i wady języków skryptowych
4. Zalety i wady programowania wsadowego i interaktywnego
5. Przechowywanie danych: interfejsy do bazy danych, szybkie zapisywanie złożonych struktur danych za pomocą serializacji, obsługa plików CSV.
6. Szybkie obliczenia: sterowanie programami napisanymi w języku C, Cython, kompilatory just-in-time
7. Tworzenie wykresów i diagramów za pomocą Matplotlib
8. Obliczenia naukowe i inżynierskie za pomocą NumPy i SciPy
9. Sterowanie solverami programowania liniowego-całkowitoliczbowego (MIP): CPLEX,

Gurobi, GLPK.

Sposób obliczania oceny końcowej

Zaliczenie (1 ECTS):

Podstawą zaliczenia jest indywidualna realizacja prostego projektu programistycznego i ustane odpowiedzi na pytania odnoszące się do wykorzystanych w nim metod i narzędzi.

Egzamin (3 ECTS):

Podstawą zaliczenia jest indywidualna realizacja złożonego projektu programistycznego, ustane odpowiedzi na pytania odnoszące się do wykorzystanych w nim metod i narzędzi, a także innych zagadnień omawianych na zajęciach.

Wymagania wstępne i dodatkowe

Podstawowy kurs programowania w dowolnym języku.

Zalecana literatura i pomoce naukowe

1. Guido van Rossum, *Przewodnik po języku Python*, Angielska wersja jest załączona do każdej dystrybucji języka Python. Polska wersja dostępna jest na stronie: <https://pl.python.org/docs/tut/tut.html>
2. Jeffrey Elkner, Allen B. Downey, and Chris Meyers, *How to Think Like a Computer Scientist, Learning with Python*, <http://www.openbookproject.net/thinkcs/python/english2e/>
3. Mark Lutz, *Python. Wprowadzenie*, Helion, 2010.
4. Mark Pilgrim, *Dive Into Python*, <http://www.diveintopython.net/>
5. *Python Scientific Lecture Notes*, <http://scipy-lectures.github.io/index.html>
6. Wilson, Greg, et al., Best practices for scientific computing, *PLoS biology* 12.1 (2014): e1001745. (<http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745>)
7. Travis E. Oliphant, Python for Scientific Computing, *Computing in Science & Engineering*, May/June 2007, pp.10-20.
8. Fernando Pérez, Brian E. Granger, John D. Hunter, Python: An Ecosystem for Scientific Computing, *Computing in Science & Engineering*, March/April 2011, pp. 13-21.
9. Oprogramowanie open source Pierra Raybauta: WinPython (Python 3.5), <http://sourceforge.net/projects/winpython/>
10. Oprogramowanie Continuum Analytics: Anaconda (Python 3.5), <https://store.continuum.io/cshop/anaconda/>

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Wykładowcy nie prowadzą badań nad rozwojem omawianych metod i narzędzi, ale na co dzień korzystają z nich w swojej pracy naukowej.

Informacje dodatkowe

Brak

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w konwersatoriach	14 godz
Wykonanie projektu	14 godz
Samodzielne studiowanie tematyki zajęć	14 godz
Sumaryczne obciążenie pracą studenta	42 godz
Punkty ECTS za moduł	3 ECTS