

Nazwa Wydziału	Wydział Matematyki i Informatyki
Nazwa jednostki prowadzącej moduł	Instytut Informatyki i Matematyki Komputerowej
Nazwa modułu kształcenia	<b>Programowanie w języku Python</b>
Kod modułu	
Język kształcenia	polski
Efekty kształcenia dla modułu kształcenia	<p>Wiedza</p> <ul style="list-style-type: none"> <li>• Student zna składnię języka Python oraz wybranych bibliotek</li> </ul> <p>Umiejętności</p> <ul style="list-style-type: none"> <li>• Student potrafi tworzyć kod proceduralny i obiektowy języka Python</li> <li>• rozumie pojęcie „pythonic code” i potrafi ją stosować</li> <li>• potrafi przeprowadzać testy jednostkowe, profilować aplikację i znajdować wąskie gardła wydajnościowe</li> <li>• zna biblioteki naukowe scipy i numpy i potrafi z nich efektywnie korzystać</li> </ul>
Typ modułu kształcenia (obowiązkowy/fakultatywny)	fakultatywny – ograniczonego wyboru (dla studentów wydziału)
Rok studiów	
Semestr	
Imię i nazwisko osoby/osób prowadzących moduł	mgr Wojciech Czarnecki
Imię i nazwisko osoby/osób egzaminującej/egzaminujących bądź udzielającej zaliczenia, w przypadku gdy nie jest to osoba prowadząca dany moduł	
Sposób realizacji	
Wymagania wstępne i dodatkowe	Ukończony kurs programowania lub równoważny
Rodzaj i liczba godzin zajęć dydaktycznych wymagających bezpośredniego udziału nauczyciela akademickiego i studentów, gdy w danym module przewidziane są takie zajęcia	<p>Wykład, laboratorium</p> <p>Wykład: 30 Laboratorium: 30 <b>Łącznie: 60</b></p>

Liczba punktów ECTS przypisana modułowi	
Bilans punktów ECTS	
Stosowane metody dydaktyczne	<ul style="list-style-type: none"> <li>• Metody podające <ul style="list-style-type: none"> <li>○ wykład informacyjny</li> <li>○ objaśnienie lub wyjaśnienie</li> </ul> </li> <li>• metody praktyczne: <ul style="list-style-type: none"> <li>○ metoda projektów</li> <li>○ ćwiczenia przedmiotowe</li> </ul> </li> </ul>
Metody sprawdzania i kryteria oceny efektów kształcenia uzyskanych przez studentów	<p>Studenci są oceniani w sposób ciągły na podstawie uczestnictwa i aktywności na zajęciach oraz realizacji projektów.</p> <p>Metody kształtujące dla oceny ciągłej to:</p> <ul style="list-style-type: none"> <li>- bieżąca ocena i ewentualna korekta realizacji zadań wykonywanych w trakcie laboratoriów</li> <li>- ocena projektów na podstawie ich wydajności w odniesieniu do rozwiązywanego, praktycznego problemu</li> </ul> <p>Metody podsumowujące:</p> <ul style="list-style-type: none"> <li>- ostateczna, końcowa ocena wykonania zadań w tym zgodności z zaleceniami.</li> </ul>
Forma i warunki zaliczenia modułu, w tym zasady dopuszczenia do egzaminu, zaliczenia, a także forma i warunki zaliczenia poszczególnych zajęć wchodzących w zakres danego modułu	<p>Projekt końcowy.</p> <p>Zaliczenie z laboratorium na ocenę. Zaliczenie na podstawie oddanych zadań oraz aktywnego uczestnictwa.</p> <p>Obowiązuje następująca skala ocen (1-100 pkt.):</p> <ul style="list-style-type: none"> <li>• 0-50 pkt. – ocena ndst</li> <li>• 51-60 pkt. – ocena dst</li> <li>• 61-70 pkt. – ocena + dst</li> <li>• 71-80 pkt. – ocena db</li> <li>• 81-90 pkt. – ocena + db</li> <li>• 91-100 pkt. – ocena bdb</li> </ul>
Treści modułu kształcenia	<p>Celem zajęć jest zapoznanie studentów z językiem Python oraz jego zaawansowanymi funkcjonalnościami (testowaniem kodu, profilowaniem, bibliotekami naukowymi)</p> <p>Ramowy plan zajęć:</p> <ol style="list-style-type: none"> <li>1. Podstawy języka</li> <li>2. Pakiety, moduły i biblioteka standardowa</li> <li>3. Model obiektowy i wyjątki</li> <li>4. Kolekcje, listy/słowniki/zbiory składane</li> </ol>

	<ol style="list-style-type: none"> <li>5. Iteratory i generatory</li> <li>6. Pliki i strumienie</li> <li>7. Testowanie i analiza kodu</li> <li>8. Obliczenia naukowe - scipy i numpy</li> <li>9. Dekoratory, adnotacje</li> <li>10. Wątki i procesy</li> <li>11. Cython</li> </ol>
<p>Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego modułu</p>	<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> <li>1. Python in practise. Mark Summerfield.</li> <li>2. Programming Python. Mark Lutz et al.</li> <li>3. Fundaments of Python: First Programs. Ken Lambert</li> <li>4. Fundaments of Python: Data Structures. Ken Lambert</li> </ol>
<p>Wymiar, zasady i forma odbywania praktyk, w przypadku, gdy program kształcenia przewiduje praktyki</p>	