



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie skryptowe [S1Cybez1>PS]

Przedmiot

Kierunek studiów

Cyberbezpieczeństwo

Rok/Semestr

1/2

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

24

Laboratorium

24

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Paweł Sroka

pawel.sroka@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający naukę powinien posiadać poszerzoną wiedzę z matematyki i fizyki, w tym podstaw algebry liniowej i rachunku prawdopodobieństwa. Dodatkowo, powinien posiadać wiedzę i podstawowe umiejętności programowania oraz obsługi komputera typu PC. Wskazana jest też podstawowa znajomość wybranych protokołów telekomunikacyjnych, w tym protokołu HTTP. Student powinien rozumieć konieczność zdobywania nowej wiedzy i umiejętności związanych z realizowanym kierunkiem studiów.

Cel przedmiotu

Celem przedmiotu jest nabycie przez studentów podstawowej wiedzy i umiejętności z zakresu programowania z wykorzystaniem wybranych języków skryptowych. Dodatkowo, w ramach przedmiotu przekazana zostanie wiedza i umiejętności z zakresu przetwarzania i wizualizacji zgromadzonych danych przy użyciu wybranych języków skryptowych, a także programowej implementacji wybranych zagadnień matematycznych.

Przedmiotowe efekty uczenia się

Wiedza:

1. Ma zaawansowaną wiedzę w zakresie zasady tworzenia programów komputerowych, wykorzystania algorytmów i struktur danych korzystając z poznanych języków skryptowych.

Umiejętności:

1. Potrafi opracować proste aplikacje korzystając z poznanych języków skryptowych w celu przeprowadzenia analizy i rozwiązywania problemów właściwych dla kierunku studiów.
2. Potrafi zapisać, przedstawić i przetworzyć zebrane dane w formie liczbowej i graficznej korzystając z poznanych języków skryptowych.
3. Potrafi zaimplementować i wykorzystać znane modele matematyczne i algorytmy do rozwiązywania problemów korzystając z poznanych języków skryptowych.

Kompetencje społeczne:

1. Rozumie konieczność poszerzania wiedzy na temat korzystania z języków skryptowych; ma świadomość, że wiedza i umiejętności w tym obszarze szybko ewoluują.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w trakcie wykładów jest weryfikowana podczas zaliczenia, który ma formę pisemną. Zaliczenie pisemne składa się 6-10 pytań (testowych i/lub otwartych), które mogą być różnie punktowane. Umiejętności nabyte podczas laboratorium są weryfikowane podczas realizacji 4-8 ćwiczeń praktycznych polegających na implementacji oprogramowania o określonej funkcjonalności w wybranym języku skryptowym, zgodnie z przygotowaną instrukcją do zadania. Każde z ćwiczeń jest oceniane w postaci punktów na podstawie przygotowanego sprawozdania, przy czym liczba punktów odpowiadająca zadaniu jest zależna od jego stopnia skomplikowania. Wpływ na ocenę końcową ma też ocena pracy i zaangażowania studenta na zajęciach, a także realizacja ewentualnych dodatkowych zadań domowych.

W każdej formie zaliczenia przedmiotu ocena zależy od liczby zdobytych przez studenta punktów w stosunku do maksymalnej liczby punktów obowiązkowych. Warunkiem pozytywnego zaliczenia jest otrzymanie co najmniej 50% punktów możliwych do zdobycia. Zależność oceny od liczby punktów definiuje Regulamin Studiów. Dodatkowo zasady zaliczania przedmiotu i dokładne progi zaliczeniowe zostaną przekazane studentom na początku semestru z wykorzystaniem uczelnianych systemów elektronicznych oraz na pierwszych zajęciach (w każdej formie zajęć).

Treści programowe

W ramach zajęć omawiane są aspekty teoretyczne i praktyczne wykorzystania wybranych języków skryptowych do rozwiązywania wybranych problemów z zakresu matematyki, informatyki i telekomunikacji. Tematyka zajęć obejmuje podstawowe zagadnienia implementacji programów z wykorzystaniem języków skryptowych, w tym omówienie składni języków oraz sposobów generowania, zapisu i przetwarzania danych.

Tematyka zajęć

W ramach wykładów omawiane są następujące zagadnienia:

1. Wprowadzenie do języków skryptowych - podstawowe elementy i składnia wybranych języków skryptowych (np. Python, JavaScript): typy danych, operatory arytmetyczne i kolejność działań, operacje wejścia-wyjścia, pętle, podejmowanie decyzji (instrukcje warunkowe), funkcje, operacje z wykorzystaniem tablic.
2. Odczyt, zapis i przetwarzanie danych z wykorzystaniem wybranych języków skryptowych. Graficzna prezentacja wyników.
3. Implementacja wybranych zagadnień matematycznych z wykorzystaniem wybranych języków skryptowych - elementy rachunku prawdopodobieństwa, statystyki matematycznej i algebry liniowej.
4. Realizacja wybranych usług sieciowych (np. HTTP) z wykorzystaniem wybranych języków skryptowych.
5. Podstawy korzystania z baz danych przy użyciu wybranych języków skryptowych.

W ramach laboratoriów realizowane są następujące tematy:

1. Podstawowe elementy i składnia wybranych języków skryptowych (np. Python, JavaScript): typy danych, operatory arytmetyczne i kolejność działań, operacje wejścia-wyjścia, pętle, podejmowanie decyzji (instrukcje warunkowe), funkcje, operacje z wykorzystaniem tablic.

2. Przetwarzanie danych z wykorzystaniem wybranych języków skryptowych - odczyt i zapis plików w różnych formatach, graficzna prezentacja wyników, wybrane biblioteki .
3. Implementacja wybranych zagadnień rachunku prawdopodobieństwa - generowanie liczb pseudolosowych, wyznaczanie momentów, elementy analizy statystycznej (zgodność z rozkładem, losowość).
4. Implementacja wybranych zagadnień algebry liniowej i rachunku macierzowego.
5. Realizacja wybranych usług sieciowych (np. HTTP) i użycie baz danych z wykorzystaniem wybranych języków skryptowych.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, ilustrowana przykładami realizowanymi na komputerze lub przedstawianymi na tablicy, ew. forma warsztatowa.
2. Ćwiczenia laboratoryjne: wykonanie zadań podanych przez prowadzącego z wykorzystaniem komputerów - ćwiczenia praktyczne, ew. wspomagane prezentacją multimedialną.

Literatura

Podstawowa:

Mark Lutz, Python. Wprowadzenie. Wydanie V, Helion 2020

Boschetti A., Massaron L., Python: podstawy nauki o danych, Helion, 2017

Suehring S., JavaScript: krok po kroku, RM, 2009

Uzupełniająca

McKinney W., Python for data analysis, O'Reilly, 2013

Robert Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion 2021

Krajka A., Python: podstawy języka i aplikacje internetowe, Wyd. Uniwersytetu Marii Curie-Skłodowskiej, 2011

www.w3schools.com

Uzupełniająca:

-

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	88	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	48	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	40	1,50