

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2019/2020

Wydział Mechaniczny

Kierunek studiów: Systemy i urządzenia przemysłowe

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: U

Stopień studiów: II

Specjalności: Systemy i urządzenia ciepłne

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Programowanie wybranych języków wysokiego poziomu
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	Programming selected high-level languages
KOD PRZEDMIOTU	WM SIUP oIIN C8 19/20
KATEGORIA PRZEDMIOTU	Przedmioty specjalnościowe
LICZBA PUNKTÓW ECTS	4.00
SEMESTRY	1

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	PROJEKT	SEMINARIUM
1	9	0	0	9	9	0

3 CELE PRZEDMIOTU

Cel 1 Zapoznanie z językami wysokiego poziomu oraz technikami i narzędziami ich programowania oraz nabycie umiejętności programowania w tych językach.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 Brak

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Student który zaliczy przedmiot uzyskuje wiedzę w zakresie budowy aplikacji służących do rozwiązywania i modelowania zagadnień inżynierskich z dziedziny mechaniki i budowy maszyn

EK2 Umiejętności Student który zaliczy przedmiot potrafi napisać program do zagadnień w zakresie inżynierii mechanicznej, szczególnie w zakresie swojej specjalności oraz zinterpretować dane uzyskane na drodze symulacji komputerowej

EK3 Umiejętności Student który zaliczy przedmiot potrafi zastosować do budowy programów wiedzę posiadaną lub zaczerpniętą z różnych źródeł, przy wykonywaniu analizy problemu technicznego, nie tylko w zakresie mechaniki i budowy maszyn ale także kierunków pokrewnych, tj. inżynierii produkcji, energetyki, zarządzania, mechatroniki.

EK4 Kompetencje społeczne Student który zaliczy przedmiot uzyskuje umiejętność ciągłego doksztalcania się podnoszenia kompetencji zawodowych i społecznych oraz inspirowania swojego zespołu do poszukiwania aktualnych rozwiązań technicznych

6 TREŚCI PROGRAMOWE

LABORATORIUM KOMPUTEROWE		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
K1	Zapoznanie się ze środowiskiem programistycznym dla języka Python, podstawy składni języka Python: Operatory, typy liczbowe i tekstowe. Instrukcje warunkowe, pętle.	2
K2	Zasady definiowania funkcji i obsługa pakietów i modułów. Wykorzystanie funkcji i zewnętrznych modułów w obliczeniach.	2
K3	Podstawy obiektowości w Pythonie, tworzenie i wykorzystanie klas, metod i konstruktorów. Dziedziczenie, polimorfizm i interfejsy	3
K4	Graficzne interfejsy użytkownika na przykładzie prostych projektów obliczeniowych	2

PROJEKT		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P1	Wybór tematyki projektu własnego. Specyfikacja wymagań funkcjonalnych i niefunkcjonalnych, określenie przypadków użycia.	1
P2	Rozpoczęcie pracy nad implementacją projektu, przygotowanie wymaganych zmiennych, implementacja pierwszych funkcji	2

PROJEKT		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P3	Praca nad implementacją projektu, implementacja głównych funkcjonalności projektu	3
P4	Praca nad implementacją projektu, przygotowanie interfejsu dla użytkownika, testy działania aplikacji	2
P5	Prezentacja projektu przed całą grupą i zaliczenie	1

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Wprowadzenie do programowania, typy programowanie, pojęcie algorytmu	1
W2	Wprowadzenie do języków programowania, omówienie wybranych języków wysokiego poziomu (Python i R)	2
W3	Typy zmiennych i danych (typ zmiennej i wartościowanie zmiennych, wyrażenia arytmetyczne i logiczne instrukcje warunkowe, pętle, tablice, rekordy, zbiory, pliki)	1
W4	Funkcje - składnia i semantyka, sposoby przekazywania parametrów: przez wartość i przez zmienną	2
W5	Programowanie obiektowe klasy, konstruktory, dziedziczenie, polimorfizm i interfejsy.	2
W6	Graficzny interfejs użytkownika	1

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Ćwiczenia projektowe

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	45
Konsultacje przedmiotowe	0
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	0
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	0
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	45
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	4.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Projekt zespołowy

F2 Test

OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

WARUNKI ZALICZENIA PRZEDMIOTU

W1 Student musi uzyskać pozytywną ocenę z każdego efektu kształcenia

W2 Dopuszcza się max. 1 nieobecność nieusprawiedliwioną na laboratorium komputerowym. W takim przypadku student(ka) otrzymuje brak oceny z tego laboratorium

W3 Ocena końcowa jest średnią z: laboratoriów komputerowych, projektu indywidualnego i testu z wykładów.

W4 Brak oceny z obowiązkowego zaliczenia skutkuje wartością 0 (zero) wliczaną do średniej.

W5 Obecność na wykładach ma wpływ na podwyższenie lub obniżenie oceny końcowej w przypadkach dyskusyjnych.

W6 Zajęcia laboratoryjne mogą być odrabiane na innych zespołach pod warunkiem dostępnego miejsca (stanowiska) i zgody prowadzącego.

W7 Każde przekroczenie wyznaczonego terminu zaliczenia lub oddania sprawozdania/laboratorium/projektu skutkuje obniżeniem oceny proporcjonalnym do czasu przekroczenia terminu.

OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA
B1 Projekt zespołowy
KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 3.0	Student zna podstawowe zasady budowy aplikacji służących do rozwiązywania i modelowania zagadnień inżynierskich i podstawy składni co najmniej jednego języka programowania.
NA OCENĘ 4.0	Student zna zasady budowy interfejsu aplikacji oraz podstawy obiektowości w co najmniej jednym języku programowania
NA OCENĘ 5.0	Student zna zagadnienia z niższych ocen w kilku językach programowania, zna zasady debugowania i wzorce projektowe
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 3.0	Student potrafi prostą aplikację do nieskomplikowanych zagadnień w zakresie inżynierii mechanicznej
NA OCENĘ 4.0	Student potrafi napisać aplikację o średnim stopniu złożoności do zagadnień w zakresie inżynierii mechanicznej
NA OCENĘ 5.0	Student potrafi samodzielnie dobrać niezbędne narzędzia oraz napisać aplikację o średnim stopniu złożoności do rozbudowanych zagadnień w zakresie inżynierii mechanicznej
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 3.0	Student potrafi zastosować wiedzę z różnych źródeł przy wykonaniu prostej analizy problem technicznego
NA OCENĘ 4.0	Student potrafi zastosować wiedzę z różnych źródeł przy wykonaniu analizy problem technicznego o średnim stopniu złożoności
NA OCENĘ 5.0	Student potrafi zastosować wiedzę z różnych źródeł przy wykonaniu analizy problem technicznego o wysokim stopniu złożoności
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 3.0	Student potrafi samodzielnie znaleźć rozwiązania na proste problemy z zakresu tworzenia aplikacji w językach wysokiego poziomu
NA OCENĘ 4.0	Student potrafi współdziałać w zespole w celu stworzenia funkcjonalnej aplikacji
NA OCENĘ 5.0	Student potrafi samodzielnie znaleźć nowe sposoby rozwiązania problemów związanych z programowaniem i zaimplementować samodzielnie lub w zespole

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1	M2_W01	Cel 1	K2 K4 P1 P2 W4	N1 N2 N3	F1 F2 P1
EK2	M2_U09	Cel 1	K1 K2 K3 K4 P2 P3 P4 W1 W2 W3 W4 W5 W6	N1 N2 N3	F1 F2 P1
EK3	M2_U13	Cel 1	K2 K3 P1 P2 P3 P4 W6	N1 N2 N3	F1 F2 P1
EK4	M2_K01	Cel 1	K1 P5 W6	N1 N2 N3	F1 F2 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] Mark Lutz — *Python. Wprowadzenie*, Gliwice, 2011, Helion
- [2] Dusty Phillips — *Python 3 Object-Oriented Programming*, Birmingham, 2018, Packt Publishing
- [3] Tilman M. Davies — *The Book of R*, San Francisco, 2016, No Starch Press

LITERATURA UZUPEŁNIAJĄCA

- [1] Michał Jaworski — *Expert Python Programming*, Birmingham, 2019, Packt Publishing
- [2] Norman Matloff — *Art of R Programming*, San Francisco, 2011, No Starch Press

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Paweł Lempa (kontakt: plempe@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejsowość, data)

(odpowiedzialny za przedmiot)

(dziekan)