

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Nazwa modułu zajęć: Informatyka - kurs podstawowy

Rok akademicki: 2019/2020 Kod: CCHB-1-210-s Punkty ECTS: 2

Wydział: Inżynierii Materiałowej i Ceramiki

Kierunek: Chemia Budowlana Specjalność: —

Poziom studiów: Studia I stopnia Forma studiów: Stacjonarne

Język wykładowy: Polski Profil: Ogólnoakademicki (A) Semestr: 2

Strona www: —

Prowadzący moduł: dr hab. Wierzba Bartłomiej (bwierzba@agh.edu.pl)

Opis efektów uczenia się dla modułu zajęć

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Powiązania z KEU	Sposób weryfikacji i oceny efektów uczenia się osiągniętych przez studenta w ramach poszczególnych form zajęć i dla całego modułu zajęć
Wiedza: zna i rozumie			
M_W001	Student ma podstawową wiedzę z zakresu konstruowania i działania algorytmów komputerowych dzięki którym może rozwiązać proste zagadnienia fizyczne.	CHB1A_W08	Kolokwium
M_W002	Student ma podstawową wiedzę z zakresu analizy numerycznej. Wykorzystywania algorytmów do rozwiązywania zagadnień początkowo brzegowych, całkowania oraz różniczkowania numerycznego.	CHB1A_W08	Kolokwium
Umiejętności: potrafi			
M_U001	Student potrafi przeliczać liczby zapisane w różnych systemach liczbowych oraz zna arytmetykę komputerową.	CHB1A_U07	Kolokwium
M_U002	Student potrafi napisać prosty program w wybranym języku programowania.	CHB1A_U07	Kolokwium
Kompetencje społeczne: jest gotów do			

M_K001	Student rozumie potrzebę ciągłego dokształcania się z zakresu informatyki oraz wykorzystywania nowoczesnych metod programistycznych.	CHB1A_K01	Aktywność na zajęciach
--------	--	-----------	------------------------

Liczba godzin zajęć w ramach poszczególnych form zajęć

Suma	Forma zajęć dydaktycznych										
	Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
30	15	0	15	0	0	0	0	0	0	0	0

Matryca kierunkowych efektów uczenia się w odniesieniu do form zajęć i sposobu zaliczenia, które pozwalają na ich uzyskanie

Kod MEU	Student, który zaliczył moduł zajęć zna i rozumie/potrafi/jest gotów do	Forma zajęć dydaktycznych										
		Wykład	Ćwiczenia audytoryjne	Ćwiczenia laboratoryjne	Ćwiczenia projektowe	Konwersatorium	Zajęcia seminaryjne	Zajęcia praktyczne	Zajęcia terenowe	Zajęcia warsztatowe	Prace kontrolne i przejściowe	Lektorat
Wiedza: zna i rozumie												
M_W001	Student ma podstawową wiedzę z zakresu konstruowania i działania algorytmów komputerowych dzięki którym może rozwiązać proste zagadnienia fizyczne.	+	-	+	-	-	-	-	-	-	-	-
M_W002	Student ma podstawową wiedzę z zakresu analizy numerycznej. Wykorzystywania algorytmów do rozwiązywania zagadnień początkowo brzegowych, całkowania oraz różniczkowania numerycznego.	+	-	-	-	-	-	-	-	-	-	-
Umiejętności: potrafi												
M_U001	Student potrafi przeliczać liczby zapisane w różnych systemach liczbowych oraz zna arytmetykę komputerową.	-	-	+	-	-	-	-	-	-	-	-
M_U002	Student potrafi napisać prosty program w wybranym języku programowania.	-	-	+	-	-	-	-	-	-	-	-
Kompetencje społeczne: jest gotów do												

M_K001	Student rozumie potrzebę ciągłego doksztalcania się z zakresu informatyki oraz wykorzystywania nowoczesnych metod programistycznych.	+	-	-	-	-	-	-	-	-	-	-
--------	--	---	---	---	---	---	---	---	---	---	---	---

Nakład pracy studenta (bilans punktów ECTS)

Forma aktywności studenta	Obciążenie studenta
Udział w zajęciach dydaktycznych/praktyka	30 godz
Przygotowanie do zajęć	10 godz
Samodzielne studiowanie tematyki zajęć	10 godz
Egzamin lub kolokwium zaliczeniowe	2 godz
Sumaryczne obciążenie pracą studenta	52 godz
Punkty ECTS za moduł	2 ECTS

Pozostałe informacje

Szczegółowe treści kształcenia w ramach poszczególnych form zajęć (szczegółowy program wykładów i pozostałych zajęć)

Wykład

Wiadomości wstępne

Wiadomości wstępne: co to jest informatyka; podstawowe rozkazy, które potrafi wykonać komputer; kodowanie liczb w komputerze; arytmetyka komputerowa; systemy pozycyjne;

Algorytmika

Algorytmika. Zasady budowania algorytmów: metody poprawnego zapisywania algorytmów; reguły stylu programowania; czytelność kodu; konwencje notacyjne, notacje opisowe, formalne, graficzne; instrukcje proste i strukturalne;

Algorytmy iteracyjne

Zasady działania algorytmów iteracyjnych; przykłady algorytmów.

Pseudokody

Podstawy programowania z wykorzystaniem pseudokodu: ogólne zasady programowania; zasady programowania iteracyjnego; najistotniejsze elementy języków oprogramowania; typy danych, nazewnictwo stałych i zmiennych, zmienne tablicowe, tablice jedno- i wielowymiarowe, instrukcje przypisania;

Wyrażenia arytmetyczne

Zapis wyrażen arytmetycznych; zasady poprawnego budowania algorytmów iteracyjnych;

Instrukcje proste

Instrukcje warunkowe proste i złożone; metody konstruowania wyrażen logicznych; instrukcja

wyboru;

Instrukcje iteracyjne

Instrukcje iteracyjne – pętle, rodzaje i zasady konstruowania; instrukcje wejścia/wyjścia;
zapisywanie programów w pseudokodzie;

Programowanie

Przykłady prostych programów.

Algorytmy szyfrujące

Omówienie algorytmów kryptograficznych. Algorytmy szyfrujące dane typu RSA.

Ćwiczenia laboratoryjne

kodowanie liczb

kodowanie liczb w komputerze; arytmetyka komputerowa; systemy pozycyjne;

pseudokody prostych algorytmów

Opracowanie prostych algorytmów oraz ich opis za pomocą pseudo kodu.

Algorytmika

Opracowanie algorytmów rozwiązywania prostych zadań z zakresu podstawowych wiadomości z zakresu matematyki i fizyki.

języki programowania

Zapoznanie się z wybranym językiem programowania spośród języków: Fortran, C, C++, VBA.

Metody i techniki kształcenia:

Wykład: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pokazy odnoszące się do prezentowanych zagadnień.

Ćwiczenia laboratoryjne: W trakcie zajęć laboratoryjnych studenci samodzielnie rozwiązują zadany problem praktyczny, dobierając odpowiednie narzędzia. Prowadzący stymuluje grupę do refleksji nad problemem, tak by otrzymane wyniki miały wysoką wartość merytoryczną.

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:

Nie określono

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:

Wykład:

– Obecność obowiązkowa: Nie

– Zasady udziału w zajęciach: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego.

Ćwiczenia laboratoryjne:

– Obecność obowiązkowa: Tak

– Zasady udziału w zajęciach: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

Sposób obliczania oceny końcowej

Ocena końcowa jest średnią ocen uzyskanych z kolokwiiów.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:

Nie określono

Wymagania wstępne i dodatkowe, z uwzględnieniem sekwencyjności modułów

brak

Zalecana literatura i pomoce naukowe

1. Wirth N., Wstęp do programowania systematycznego, WNT, 1978
2. Wirth N., Algorytmy + struktury danych = programy, WNT, 1989
3. Cormen T., Wprowadzenie do algorytmów, WNT, 1997
4. D. Harel, Rzecz o istocie informatyki. WNT, Warszawa, 2000.
5. P. Silvester, System operacyjny UNIX, WNT, W-wa, 1990.
6. B.W. Kernighan, D.M. Ritchie, Język ANSI C, PWN, 1994.
7. Dostępne podręczniki z zakresu algorytmiki i systemów operacyjnych.
8. Dostępne podręczniki z zakresu podstaw programowania.

Publikacje naukowe osób prowadzących zajęcia związane z tematyką modułu

Nie podano dodatkowych publikacji

Informacje dodatkowe

Brak