**Nazwa przedmiotu:**

Algorytmy i struktury danych

**Koordynator przedmiotu:**

dr Jadwiga Chudzicka

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Zarządzanie i Inżynieria Produkcji

**Grupa przedmiotów:**

Technologie informatyczne

**Kod przedmiotu:**

ASDAN

**Semestr nominalny:**

2 / rok ak. 2012/2013

**Liczba punktów ECTS:**

3

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

godziny kontaktowe: 30h, zapoznanie się ze wskazaną literaturą 20h, samodzielne ćwiczenia praktyczne + konsultacje 20h, przygotowanie się do egzaminu 20h. Razem 90h = 3 ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

godziny kontaktowe: 30h. Razem 30h = 1 ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

zapoznanie się ze wskazaną literaturą 20h, samodzielne ćwiczenia praktyczne + konsultacje 20h, przygotowanie się do egzaminu 20h. Razem 60h = 2 ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

algorytmika, programowanie komputerów, algorytm, typy danych, struktury danych, złożoność algorytmów, rekurencja, sortowanie danych, algorytmy numeryczne, kodowanie danych, kompresja danych, grafy.

**Limit liczby studentów:**

min 15

**Cel przedmiotu:**

Dostarczenie narzędzi wspomagających rozwiązywanie różnorodnych zagadnień matematycznych, inżynierskich i innych przy pomocy komputerów. Zapoznanie z najpopularniejszymi technikami programowania, strukturami danych i algorytmami. Prezentacja szerokiego wachlarza przykładów do wykorzystania w przyszłości wg potrzeb. Rozwinięcie umiejętności studentów skutecznego i efektywnego rozwiązywania problemów z różnych dziedzin metodami stosowanymi w informatyce.

**Treści kształcenia:**

1. Wprowadzenie do metodologii programowania. Geneza algorytmiki. Pojęcia podstawowe. Definicja i cechy charakterystyczne poprawnych i efektywnych algorytmów. Sposoby zapisu algorytmów. Poprawność algorytmów. 2. Typy i struktury danych. Typy proste i złożone. Przetwarzanie liczb, wartości logicznych i tekstów. 3. Struktury danych: listy i ich tablicowe implementacje, kolejki, pojęcie stosu i jego zastosowania, drzewa, zbiory. 4. Analiza złożoności algorytmów Terminologia i definicje. Typy złożoności obliczeniowej. Techniki optymalizacji programów. Ilustracja metod na przykładach. 5. Techniki rekurencyjne Pojęcie rekurencji. Ilustracja rekurencji na przykładach: wyznaczanie wartości funkcji silnia, ciąg Fibonacciego. Typy programów rekurencyjnych. Pułapki programów rekurencyjnych. Rekurencje i ich odpowiedniki iteracyjne. 6. Techniki sortowania Prezentacja metod sortowania: sortowanie przez wstawianie, sortowanie bąbelkowe, szybkie sortowanie, sortowanie przez kopcowanie, sortowanie przez scalanie, sortowanie zewnętrzne. Klasy algorytmów sortowania. 7. Techniki przeszukiwania Przeszukiwanie liniowe. Przeszukiwanie binarne. Metody transformacji kluczowej. 8. Przeszukiwanie tekstów. Metody wykorzystywane w algorytmach przeszukiwania tekstów. 9. Przeszukiwanie tekstów - cd. 10. Algorytmy numeryczne cz. I Metody iteracyjne rozwiązywania układów równań liniowych. Metoda eliminacji Gaussa rozwiązywania układów równań liniowych. Poszukiwanie miejsc zerowych funkcji. Iteracyjne obliczanie wartości funkcji. 11. Algorytmy numeryczne cz. II Interpolacja funkcji. Ekstrapolacja funkcji. Różniczkowanie funkcji. Całkowanie funkcji. 12. Algorytmika grafów. Kodowanie i kompresja danych. Sposoby reprezentacji grafów. Operacje na grafach. Algorytmy poszukiwania najkrótszych dróg w grafach. 13. Podstawowe wiadomości na temat kodowania danych (szyfrowania wiadomości) i popularne metody łamania szyfrów. 14. Optymalizacja algorytmów. Optymalizacja poprzez transformację algorytmów rekurencyjnych na ich postać iteracyjną. Powody, dla których stosuje się derekursywację. Wykorzystanie stosu. Schematy derekursywacji. 15. Inne techniki programowania Metody heurystyczne. Algorytmy genetyczne. Programowanie typu „dziel i zwyciężaj”. Programowanie dynamiczne.

**Metody oceny:**

Egzamin pisemny sprawdzający praktyczne umiejętności rozwiązywania typowych problemów spotykanych w praktyce inżynierskiej i pytania sprawdzające wiedzę teoretyczną.

**Egzamin:**

tak

**Literatura:**

• Wróblewski P., Algorytmy: struktury danych i techniki programowania, Wyd. Helion, Gliwice 2010.
• Czech Z., Deorowicz S., Fabian P., Algorytmy i struktury danych: wybrane zagadnienia, Wydawnictwo Politechniki Śląskiej, Gliwice 2007.
• Banachowski L., Diks K., Rytter W., Algorytmy i struktury danych, Wydawnictwa Naukowo-Techniczne, Warszawa 2006.
• Dańko A., Algorytmy i struktury danych: zadania, Wydawnictwo Polsko-Japońskiej Wyższej Szkoły Technik Komputerowych, Warszawa 2006.

**Witryna www przedmiotu:**

www.electurer.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt Wpisz opis:**

ma uporządkowaną wiedzę z zakresu narzędzi wspomagających rozwiązywanie różnorodnych zagadnień matematycznych, inżynierskich i innych za pomocą komputera.

Weryfikacja:

egzamin pisemny

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Efekt Wpisz opis:**

potrafi ocenić przydatność rutynowych metod i narzędzi służących do rozwiązania prostego zadania inżynierskiego o charakterze praktycznym, typowego dla studiowanej dyscypliny inżynierskiej oraz wybrać i zastosować właściwą metodę (procedurę) i narzędzia.

Weryfikacja:

egzamin pisemny

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt Wpisz opis:**

potrafi wykazać się skutecznością w realizacji projektów o charakterze społecznym, naukowo-badawczym lub programistyczno-wdrożeniowym, wchodzących w skład studiów lub realizowanych poza studiami.

Weryfikacja:

egzamin pisemny

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**