

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim oraz angielskim Algorytmy i struktury danych/Algorithms and Data Structures
2.	Dyscyplina Nauki fizyczne
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii
5.	Kod przedmiotu/modułu 24-FZ-S1-ASD
6.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub do wyboru</i>) obowiązkowy
7.	Kierunek studiów (specjalność/specjalizacja) fizyka (fizyka komputerowa)
8.	Poziom studiów (<i>I lub II stopień, jednolite studia magisterskie, studia doktoranckie</i>) I stopień
9.	Rok studiów (<i>jeśli obowiązuje</i>) 3
10.	Semestr (<i>zimowy lub letni</i>) letni
11.	Forma zajęć i liczba godzin Wykład 30 godz., laboratorium 30 godz. Metody nauczania wykład, ćwiczenia laboratoryjne
12.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia Cezary Juszcak, dr
13.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu Podstawy C++: struktury, tablice, wskaźniki, przydział i zwalnianie pamięci
14.	Cele przedmiotu Efektywność algorytmu czy struktury danych jest w dużym stopniu niezależna od języka programowania. Znajomość klasycznych algorytmów i umiejętność projektowania własnych efektywnych algorytmów i struktur danych, jest niezbędnym atrybutem dobrego programisty, analityka czy fizyka posługującego się komputerem. Oprócz zrozumienia algorytmów na

	poziomie ogólnym w praktyce potrzebna jest umiejętność ich implementacji w wybranym języku programowania.	
15.	Treści programowe Najprostsze algorytmy: Euklidesa NWD, sito Erastotenesa. Podstawowe struktury danych: tablica, lista, drzewo BST. Kolejka i stos. Algorytmy rekurencyjne (operacje na drzewach). Złożoność algorytmów. Notacja O. Twierdzenie o rekursji uniwersalnej. Algorytmy sortujące: insertion sort, heapsort, mergesort, quick sort, counting sort, radix sort. Zasada Dziel i Zwyciężaj. Struktury słownikowe: Drzewa BST (z implementacją), drzewa czerwono-czarne, B-drzewa. Kopce łączalne. Haszowanie: łańcuchowe, otwarte. Programowanie dynamicznie: triangulacja optymalna, optymalne mnożenie macierzy. Algorytmy zachłanne na przykładzie generatora kodów Huffmana. Grafy: implementacje przez macierz sąsiedztwa, oraz listy sąsiadów. Algorytmy: Kruskala i Prima (MST) oraz Dijkstry. Problem Union-Find. Szybka transformata Fouriera. Algorytm Strassena. Sieci sortujące.	
16.	Zakładane efekty uczenia się Student zna algorytmy i struktury występujące w trakcie kursu, potrafi szacować złożoność algorytmów, i projektować algorytmy efektywne do rozwiązania nietypowych problemów. Zaprojektowane algorytmy potrafi zrealizować w języku C++.	Symbole odpowiednich kierunkowych efektów uczenia się: F1_W09 F1_U07 F1_U08 F1_K01
17.	Literatura obowiązkowa i zalecana (<i>źródła, opracowania, podręczniki itp.</i>) - Cormen, Leiserson, Rivest - Wstęp do Algorytmów - Banachowski, Diks, Rytter – Algorytmy i struktury danych - Wróblewski – Algorytmy, struktury danych i techniki programowania	
18.	Metody weryfikacji zakładanych efektów uczenia się: - egzamin pisemny, - sprawdziany - ocena pracy na zajęciach, - ocena programów napisanych w domu	
19.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu/modułu: - ciągła kontrola obecności i kontroli postępów w zakresie tematyki zajęć, - praca kontrolna, - egzamin pisemny	
20.	20. Nakład pracy studenta/doktoranta	
	forma działań studenta/doktoranta	liczba godzin na realizację działań

	Zajęcia (wg planu studiów) z prowadzącym: - wykład: - konwersatorium: - laboratorium: - inne:	30 - 30
	Praca własna studenta, doktoranta (w tym udział w pracach grupowych): - przygotowanie do zajęć: - opracowanie wyników: - czytanie wskazanej literatury: - przygotowanie prac/wystąpień/projektów: - napisanie raportu z zajęć: - przygotowanie do sprawdzianów i egzaminu:	30 - 10 - - 30
	Łączna liczba godzin	130
	Liczba punktów ECTS	5