**Nazwa przedmiotu:**

Algorithms & Data Structures

**Koordynator przedmiotu:**

Roman Podraza

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Computer Science

**Grupa przedmiotów:**

Technical Courses

**Kod przedmiotu:**

EADS

**Semestr nominalny:**

3 / rok ak. 2015/2016

**Liczba punktów ECTS:**

6

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

- lectures attendance: 15 x 2 h = 30 h;
- preparation to lectures (reviewing slides, notes and a textbook): 30 h;
- tutorial attendance: 7,5 x 2 h = 15 h;
- preparation to tutorials (homework): 12 h;
- preparation to test and exam: 20 h
- laboratory attendance: 6x 2.33 h = 14 h ;
- preparation to laboratories (learning exemplary lab tasks, reviewing slides, notes and a textbook): 30 h;

Total: 30 h + 30 h+ 15 h + 12 h + 20 h + 14 h + 30 h = 151 h.

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

2

**Język prowadzenia zajęć:**

angielski

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

2

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

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 15h |
| Laboratorium:  | 15h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Object-oriented programming (C++)

**Limit liczby studentów:**

40

**Cel przedmiotu:**

The students should learn to properly operate on different data structures in programming languages. They should become familiar with designing and implementing abstract data types in object oriented environment (the C++ language). They should realize consequences of implementing the same complex data types by different data structures.

**Treści kształcenia:**

Lectures:

1. Data Structures Overview (4h): problem analysis and specification, top-down design, abstract data types, data type system, object-oriented paradigm, data structure, linear data structure, tree, graph, evaluating efficiency and the O-notation. Data types: built-in types, collections, indexed collections.
2. Linear Data Structures (4h): singly-linked list, doubly-linked list, singly-linked ring, doubly-linked ring, stack, queue, priority queue, applications of the linear data structures, multiply-ordered lists, generalized lists.
3. Sorting Algorithms(4h): simple sorting (Insertion Sort, Selection Sort, Exchange Sort, Bubble Sort), indirect sorting, advanced sorting (Quicksort, Merge Sort, Heap Sort, Shell Sort), Straight Radix Sort, Radix Exchange Sort, Shell Sort.
4. Searching Algorithms (2h): linear searching, binary searching, interpolation searching, Fibonacci searching, hashing strategy, hashing functions and hash search methods, collision strategies.
5. Trees (6h): Binary Search Tree, traversing, searching a BST, node insertion and removal, Threaded Binary Search Tree, Binary Tree Sort, trees with arbitrary degree, AVL Trees, balancing, rotations, digital trees, tries, 2-3-4 Trees, Red-Black Trees, B-trees, BB-trees.
6. Recursion (2h): recursive functions, Divide-and-Conquer strategy, examples of recursion: Hanoi Towers and parsing.
7. Graphs (6h): directed and undirected graphs, paths, loops, reachability, implementation methods, adjacency matrix, linked adjacency lists, graph traversals, Depth-First traversals and Breadth-First traversals, weighted and unweighted graphs, positive-weighted shortest path problem (Dijkstra's algorithm).
8. Overivew of STL Library (2h): function genericity (overloading and templates), class genericity, containers, algorithms and iterators, standard containers (vector, deque, stack, queue).

Tutorials:

During tutorials some typical problems of data structures are formulated, discussed and solved using the C++ programming language. Arbitrary chosen compound data types are implemented with support of different data structures.

Laboratories:

The laboratory consists of 3 simple tasks involving data structures. Classes employing
• single-linked lists,
• doubly-linked rings
• and binary search trees
have to be designed, implemented and tested.

**Metody oceny:**

During the lab exercises it is possible to score up to 30 points (10 points for each lab task) and mid-term test is for 20 points. The examination is evaluated for 0-50 points. The final result is based on the following pattern:
• 5.0: 91-100 points
• 4.5: 81-90 points
• 4.0: 71-80 points
• 3.5: 61-70 points
• 3.0: 51-60 points
• 2.0: 0-50 points

**Egzamin:**

tak

**Literatura:**

1. L. Nyhoff, ADTs, Data Structures and Problem Solving with C++, Second Edition, Pearson, Prentice Hall, 2005.
2. M. A. Weiss, Data Structures and Problem Solving Using C++, Second Edition, Addison Wesley Longman Inc., 2000.
3. T. Budd, Data Structures in C++ Using the Standard Template Library, Addison Wesley Longman Inc., 1997. T. Budd, Data Structures in C++ Using the Standard Template Library, Addison Wesley Longman Inc., 1997.
4. J.H. Kingston, Algorithms and Data Structures. Design, Correctness, Analysis, Addison Wesley Longman Inc., 1998.
5. S. Sengupta, C.Ph. Korobkin, C++, Object-Oriented Data Structures, Springer-Verlag, 1994.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt EADS\_W01:**

Student knows basic operations on elements (searching, inserting, deleting) for linear data structures and trees.

Weryfikacja:

Evaluation of laboratory exercises, test and examination

**Powiązane efekty kierunkowe:** K\_W07

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07

**Efekt EADS\_W02:**

Student knows basic operations on graphs (including a solution of the shortest path problem).

Weryfikacja:

Evaluation of examination.

**Powiązane efekty kierunkowe:** K\_W07

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt EADS\_U01:**

Student can successfully implement basic linear data structures, Binary Search Trees and apply them for defining complex classes in C++.

Weryfikacja:

Laboratory exercises.

**Powiązane efekty kierunkowe:** K\_U03, K\_U07

**Powiązane efekty obszarowe:** T1A\_U03, T1A\_U07