Discipline	Algorithms and Data Structures code: 08 summer semester	
Specialty	SOFTWARE AND INTERNET TECHNOLOGIES	
ECTS credits: **9**	Form of assessment: Examination	
Lecturer	Assoc.prof. PhD /scientific title/ N.Kalcheva /name/ Room 404TB Phone: +359 899908366 E-mail: n_kalcheva@tu-varna.bg	
Department	SOFTWARE AND INTERNET TECHNOLOGIES	
Faculty	Faculty of Computer Sciences and Automation	

Annotation: The course is created for a first year students on Software and Internet Technologies. The course emphasis is on designing and analyzing of algorithms. Also fundamental data structures and its implementations are considered. The course forms knowledge and skills in the fields of correct and efficient programs and algorithms synthesis. It covers a range of important programming techniques and abstract data types (ADT).

At the end of the course, students understand date structures. They know how to use them, how to implement them several ways. The students can reason about efficiency with a big-***O*** analysis and argue for the correctness of their implementations by referring to the invariant of the ADT. Another important effect of the course is the specification, design, and implementation experience. Learning objectives:

- Algorithms and its basic features, definitions, efficiency, correctness and so on.
- Methods and approaches for algorithm’s and program’s design (such as structure programming, object-oriented programming, recursion, backtracking, “divide and conquer” approach and so on)
- Structures of data, static and dynamic, linear and non-linear (such as stack, queues, double ended queues, lists, trees and graphs)
- Well-known classical algorithms. This part of the course includes such topics as algorithms for sorting, searching and hashing, heuristic and greedy algorithms, probabilistic and randomized algorithms, genetic algorithms, etc.

CONTENTS:

Training Area	Hours lectures	Hours seminar exercises
Topic 1. Algorithms. 1.1. Algorithms as a Technology. 1.2. Turing Machine. 1.3. **Church-Turing Thesis**.	2	
Topic 2. Algorithm and Complexity 2.1. Big-*O* Notation. 2.2. Algorithm and Program Specification and Analysis. 2.3. Algorithm's Efficiency.	2	
Topic 3. Algorithm Design. 3.1. Structural Programming, 3.2. "Divide and Conquer" Approach 3.3. Backtracking. 3.4. Dynamic Programming. 3.5. Heuristic algorithms.	2	
Topic 4. Recursion. 4.1. Principles of the Recursion. 4.2. Recursive and Iterative Implementation. 4.3. General Forms of a Successful Recursive Functions. 4.5. Examples of the Classical Recursive Algorithms.	2	
Topic 5. Algorithm Correctness Evaluation. 5.1. Proofing 5.2. Program Testing 5.3. Documentation and Formatting.	2	
Topic 6. Data Structures. 6.1. Data Structures Classification.. 6.2. Abstract Data Types	2	
Topic 7. Static Data Structures. 7.1. Arrays, Dynamic Arrays 7.2. Sets 7.3. Main Operations and Implementations.	2	
Topic 8. Dynamic Data Structures. 8.1. Stack. Main Operations and Implementations 8.2. Queue. Main Operations and Implementations 8.3. Deque. Main Operations and Implementations.	2	
Topic 9. Dynamic Data Structures. Lists. 9.1. Linked lists. Definition, Main Operations and Implementation. 9.2. Double Linked Lists. 9.3. Circular Linked Lists.	2	
Topic 10. Dynamic Data Structures. Trees. 10.1. Trees. Definition, Main Operations and Implementation. 10.2. Binary Trees.	2	

10.3. Binary Search Trees. 10.4 Tree Traversal Algorithms. 10.5. AVL-Trees. Rotations. 10.6 B-trees. 10.7. Red-Black Trees.		
Topic 11. Dynamic Data Structures. Graphs. 11.1. Graphs. Definition, Main Operations and Implementations. 11.2. Program Testing 11.3. Documentation and Formatting. 11.4. Graph Traversal - BSF and DSF. 11.5. Shortest Paths in a Graph. 11.6. Transitive Closure of a Directed Graph. 11.7. Topological Sorting of a Directed Acyclic Graph. 11.8. Cycles in a Graph (Detection and Analyses). 11.9. Minimum Spanning Trees.	2	
Topic 12. Sorting Algorithm. 12.1 Sorting. Introduction and Notation. 12.2. Internal and External Sorting. 12.3. Insertion Sort. 12.4. Exchange Sort. Merge Sort. 12.5. Selection Sort. 12.6. Merge Sort. 12.7. Parallel Sorting.	2	
Topic 13. Searching Algorithms. 13.1. Searching. Introduction and Notation. 13.2. Sequential Search. 13.3. Search by Rearranging. 13.4. Jump Search. 13.5. Quadratic Search. 13.6. Binary Search. 13.7. Fibonacci Search. 13.8. Interpolation Search.	2	
Topic 14. Algorithms for Hashing. 14.1. Tables and information Retrieval. 14.2. Hashing. Hash Tables. Hash Functions. 14.3. Collision Resolution.	2	
Topic 15. Other Classical Algorithms. 15.1. Probabilistic Analysis and Randomized Algorithms. 15.2. Greedy Algorithms. 15.3. Genetic Algorithms.	2	
Topic 1. Introduction 1.1. Languages. Program environment. 1.2. Program Design Strategies. 1.3. Decomposing problems. Preconditions and Postconditions.		2

Topic 2. Program examples. 2.1. Running Time Analysis. 2.2. Worst-case, Average-case, and Best-case Analysis.2.3.		2
Topic 3. Program Examples. 3.1. Testing and Debugging. 3.2. Boundary Values.		2
Topic 4. Recursion. 4.1. Fibonacci Numbers Computation 4.2. The Eight Queens Problem		2
Topic 5. Backtracking. 5.1. Dynamic Programming. 5.2. Knapsack problem Solving.		2
Topic 6. Pointers and Dynamic Memory Control. 6.1. Dynamic Memory. 6.2. Pointers in C/C++. 6.3. Dynamic Variables. 6.4. Pointers and Linked structures.		2
Topic 7. Linear Dynamic Data Structure. 7.1. Stack. Examples. 7.2. Queue. Examples. 7.3. Deque. Examples.		2
Topic 8. Dynamic Lists. 8.1. Linked lists. Implementation and Applications. 8.2. Double Linked lists. Implementation and Applications.		2
Topic 9. Trees. 9.1. Binary Trees. Inserting an Item into a Binary Tree 9.2 Inserting an Item into a Binary Tree. 9.3. Searching for an Item in a Binary Tree. 9.4. Removing an Item from a Binary Tree.		2
Topic 10. Balanced Binary Trees. 10.1. Rotations of the Binary Search Tree. 10.2. Examples.		2
Topic 11. Graph. 11.1. Graph Implementations. 11.2. Graph Traversals. 11.3. Path Algorithms.		2
Topic 12. Sorting. 12.1. Internal sorting. 12.2. Implementation of Shellsort, Quicksort, Heapsort.		2
Topic 13. Sorting. 13.1. External sorting. 13.2. Balanced multi-way merging.		2

Topic 14. Searching. 14.1. Serial Search and Binary Search. 14.2. Implementations and Analysis.		2
Topic 15. Hashing. 15.1. Hash functions. 15.2. Collision resolution by Chaining. Examples.		2
TOTAL: 60 h	**30**	**30**