|  |  |  |
| --- | --- | --- |
| **حكومة إقليم كوردستان ـ العراق**  **وزارة التعليم العالي والبحث العلمي رئاسة جامعة التقنية دهوك**  **مديرية ضمان الجودة**  **البرامج الدراسية** |  | **حوكمەتا هەرێما كوردستانێ ـ عيراق**  **وەزارەتا خواندنا بالا و توژينيت زانستى**  **زانكويا پوليتەكنيكى يا دهوك**  **رێڤەبەريا دلنيايێ جوري**  **پروگرامێت خواندنێ** |

Kurdistan Regional Government – Iraq

Ministry of Higher Education and Scientific Research, Duhok Polytechnic University

Directorate of Quality Assurance/ Curriculum Programs

**Course Specification**

**مفردات المقرر الدراسي للسنة الدراسية** 2025-2024

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **الكلية: الكلية التقنية في زاخو** | | | | | | | | | | **College: Zakho Technical College** | | | | | | | |
| **القسم : أنظمة المعلومات الحاسوبية** | | | | | | | | | | **Dept.: Computer Information System** | | | | | | | |
| **الدرجة العلمية: بكالوريوس** | | | | | | | | | | **Degree: Bachelor** | | | | | | | |
| **المرحلة :**  **الفصل الدراسي :** | | | | | | | | | | **Class:** **Second**  **Semester: Second** | | | | | | | |
| **المقرر الدراسي :** | | | | | | | | | | **Course Name: Data Structures and Algorithms** | | | | | | | |
| **رمز المقرر:** | | | | | | | | | | **Course Code: CS17141** | | | | | | | |
| **Hours/Week : 6** | | | | | | | | | | **ECTS Credits/ Week: 7** | | | | | | | |
| **النظري: 2** | **العملي: 2** | | | | **تمارين: 2** | | | | | **Tutorial: 2** | **Pr.:2** | | | | **Thr.: 2** | | |
| **Course benchmarking:**  **مرجعیة المقرر الدراسي:** | | | | | | | | | | | | | | | | | |
| **Date of Production: 3/2/2024 تاريخ الإصدار :** | | | | | | | | | | | | | | | | | |
| **Date of Review: 3/22024 تاريخ المراجعة:** | | | | | | | | | | | | | | | | | |
| **Course Description (وصف المقرر الدراسية):** This course provides a comprehensive introduction to data structures and algorithms, fundamental tools for efficient software development. Students will learn to implement and analyze various data structures, including arrays, linked lists, trees, graphs, and hash tables. The course covers essential algorithms for searching, sorting, and graph traversal, along with algorithm design techniques like dynamic programming and greedy approaches. Emphasis is placed on practical application through C# implementation, enabling students to build robust and performant software solutions. Students will gain the skills to choose appropriate data structures and algorithms for diverse programming challenges. | | | | | | | | | | | | | | | | | |
| **Learning Objective (أهداف التعلم):**  Upon successful completion of this course, students will be able to:   1. Understand and implement fundamental data structures like arrays, linked lists, stacks, queues, trees, graphs, and hash tables in C#. 2. Analyze the time and space complexity of algorithms using Big O notation. 3. Design and implement efficient algorithms for searching, sorting, and graph traversal. 4. Apply algorithm design techniques such as dynamic programming and greedy approaches. 5. Select appropriate data structures and algorithms for solving real-world programming problems. 6. Develop robust and performant software solutions using learned DSA principles. 7. Demonstrate proficiency in C# implementation of data structures and algorithms. | | | | | | | | | | | | | | | | | |
| **Learning Outcome (مخرجات التعلم):**  Upon completing this course, students will be able to:   1. Implement core data structures (arrays, linked lists, trees, graphs, hash tables) and algorithms (searching, sorting, graph traversal) in C#. 2. Analyze algorithm efficiency using Big O notation and choose optimal solutions. 3. Design and implement solutions for common programming problems using appropriate DSA principles. 4. Apply dynamic programming and greedy strategies to solve complex problems. 5. Evaluate and compare different data structures and algorithms for specific tasks. 6. Develop and debug efficient C# code demonstrating practical DSA application. 7. Confidently approach technical interviews involving DSA concepts. | | | | | | | | | | | | | | | | | |
| **A - Knowledge Outcome :(مخرجات المعرفة)**  Upon completing this course, students will gain a solid understanding of:   1. Fundamental data structures and their properties (arrays, linked lists, trees, graphs, hash tables). 2. Core algorithm design principles and analysis techniques (Big O notation). 3. Classic algorithms for searching, sorting, and graph traversal. 4. Advanced algorithm design paradigms like dynamic programming and greedy approaches. 5. The relationship between data structures and algorithm performance. 6. C# implementation details for various data structures and algorithms. 7. Best practices for software development involving DSA. 8. The ability to critically evaluate and select appropriate DSA solutions. | | | | | | | | | | | | | | | | | |
| **Teaching methods:** **Lectures, quizz, assignments, workgroups**  طرق التدريس | | | | | | | | | | | | | | | | | |
| **Assessment or evaluation: Mid and final exam, discussions reports and quizzes at workload طرق القياس** | | | | | | | | | | | | | | | | | |
| **B- Cognitive Skills:**  This course cultivates several key cognitive skills, including:   * **Problem-solving:** Students learn to analyze problems, break them down into smaller parts, and devise algorithmic solutions. * **Logical reasoning:** Understanding how algorithms work and why certain data structures are appropriate for specific tasks strengthens logical thinking. * **Critical thinking:** Students evaluate different algorithms and data structures, comparing their efficiency and suitability for various scenarios. * **Abstract thinking:** Working with abstract data structures and algorithms enhances the ability to think at a higher level of abstraction. * **Analytical skills:** Analyzing algorithm complexity and performance develops analytical thinking. * **Algorithmic thinking:** Students develop the ability to think algorithmically, which is crucial for problem-solving in computer science. * **Attention to detail:** Implementing data structures and algorithms requires precision and attention to detail, fostering this important skill. * **Synthesis:** Students learn to combine different data structures and algorithms to create more complex solutions | | | | | | | | | | | | | | | | | |
| **C - Interpersonal Skills and Responsibility:**  While this course primarily focuses on technical skills, it also indirectly contributes to the development of some interpersonal skills and a sense of responsibility:   * **Collaboration:** Working on group projects can foster teamwork, communication, and conflict resolution skills. Students learn to divide tasks, share knowledge, and work together towards a common goal. * **Communication (through code reviews and discussions):** Explaining code and design choices to others (during code reviews, discussions, or presentations) improves communication skills, both written and verbal. Students learn to articulate technical concepts clearly and concisely. * **Time management:** Completing assignments and projects within deadlines requires effective time management skills. Students learn to prioritize tasks and allocate their time efficiently. * **Self-directed learning:** Much of the learning in a DSA course involves independent study and practice. Students develop the ability to learn independently, research topics, and solve problems on their own. * **Responsibility:** Completing assignments, attending classes, and actively participating in discussions fosters a sense of responsibility. Students learn to take ownership of their learning and are accountable for their progress. * **Professionalism (related to code quality):** Writing clean, well-documented, and efficient code promotes professionalism. Students learn the importance of producing high-quality work that is maintainable and understandable by others. This is a crucial aspect of being a responsible software developer. | | | | | | | | | | | | | | | | | |
| **D - Communication, Information Technology Skills**:  This course enhances communication and information technology skills in several ways:  Communication Skills:   * Technical Writing: Documenting code, explaining design choices, and writing reports on algorithm analysis improves technical writing skills. Students learn to communicate complex technical information clearly and concisely in written form. * Verbal Communication: Participating in class discussions, presenting solutions, and explaining code during code reviews enhances verbal communication skills. Students learn to articulate technical concepts effectively and answer questions clearly. * Code Reviews: Giving and receiving constructive feedback on code during code reviews develops communication skills related to technical critique and collaboration. Students learn to express their opinions respectfully and provide specific suggestions for improvement. * Collaboration (if group projects are involved): Working in teams requires effective communication for task delegation, information sharing, and conflict resolution. Students learn to communicate clearly and respectfully within a team environment.   Information Technology Skills:   * Programming Proficiency: The course significantly improves programming skills in C#, including the use of data structures, algorithms, and best practices for code development. * Software Development Tools: Students become familiar with the tools and environments used for C# software development (e.g., IDEs, debuggers, version control). * Problem-Solving with Technology: Students learn to use computational thinking and programming skills to solve problems, a core IT skill. * Data Analysis (indirectly): Analyzing the performance of algorithms introduces basic data analysis concepts, as students compare running times and resource usage. * Research Skills: Students may need to research specific algorithms or data structures, improving their ability to find and evaluate information from technical resources. | | | | | | | | | | | | | | | | | |
| **(جهد العمل المطلوب) Student workload required** | | | | | | | | | | | | | | | | | |
| Note | | | | Degree (Grade %) | | | | NO. | Type of Workload | | | | | | | | |
|  | | | | 0 | | | | 0 | **Seminars** | | | | | | | 1 | |
|  | | | | 0 | | | | 0 | **Assignment** | | | | | | | 2 | |
|  | | | | 5 | | | | 5 | **Homework** | | | | | | | 3 | |
|  | | | | 5 | | | | 1 | **Reports** | | | | | | | 4 | |
|  | | | | 15 | | | | 5 | **Quizzes** | | | | | | | 5 | |
| **1- Course Structure (Theoretical Part):** | | | | | | | | | | | | | | | | | |
| **Workload & Assessments**  **جهد العمل والتقييم** | | | Teach Method  طريقة التدريس | | | | Topic  المفردات | | | | | | ILOs | | Hours  الساعات | WEEK  الاسبوع | |
| Report 1 | | | Lecture note | | | | Introduction to DSA, Algorithm Analysis   * Introduction to Data Structures and Algorithms * Importance of DSA * Types of Data Structures * Algorithm Analysis (Big O Notation) | | | | | |  | | 2 | 1 | |
| Homework | | | Lecture note | | | | Arrays and Strings:   * Arrays: Basic Operations, Dynamic Arrays * Strings: String Manipulation, Pattern Matching | | | | | |  | | 2 | 2 | |
| Examples | | | Lecture note | | | | **Linked Lists:**   * Singly Linked Lists * Doubly Linked Lists * Circular Linked Lists | | | | | |  | | 2 | 3 | |
| quiz 1 | | | Lecture note | | | | Stacks and Queues:   * Stacks: Implementation and Applications * Queues: Implementation and Applications | | | | | |  | | 2 | 4 | |
| Examples | | | Lecture note | | | | Recursion:Recursive FunctionsExamples of Recursive AlgorithmsRecursion vs. Iteration | | | | | |  | | 2 | 5 | |
| Homework | | | Lecture note | | | | **Sorting Algorithms (Part 1)**   * Bubble Sort * Insertion Sort * Selection Sort | | | | | |  | | 2 | 6 | |
| Midterm Exam | | | | | | | | | | | | | | | | Lecture note | |
| Report 2 | | | Lecture note | | | | **Sorting Algorithms (Part 2)**   * Merge Sort * Quick Sort * Heap Sort | | | | | |  | |  | 8 | |
| Examples | | | Lecture note | | | | **Searching Algorithms**   * Linear Search * Binary Search | | | | | |  | | 2 | 9 | |
| Homework | | | Lecture note | | | | **Trees (Part 1)**   * Binary Trees: Properties and Traversal | | | | | |  | | 2 | 10 | |
| Quiz 2 | | | Lecture note | | | | **Trees (Part 2)**   * Binary Search Trees (BSTs) * Balanced Trees (Introduction) | | | | | |  | | 2 | 11 | |
| Homework | | | Lecture note | | | | **Graphs**   * Graph Representations * Graph Traversal Algorithms (BFS, DFS) | | | | | |  | | 2 | 12 | |
| Examples | | | Lecture note | | | | **Hash Tables**   * Hash Functions * Collision Handling | | | | | |  | | 2 | 13 | |
| Examples | | | Lecture note | | | | **Dynamic Programming and Greedy Algorithms (Introduction)**   * Dynamic Programming Concepts * Greedy Algorithm Concepts | | | | | |  | | 2 | 14 | |
| Final exam | | | | | | | | | | | | | | | | 15 | |
| **2-Course Structure (Practical Part & Tutorial):** | | | | | | | | | | | | | | | | | |
| **Workload & Assessments**  **جهد العمل والتقييم** | | Teach Method  طريقة التدريس | | | | Topic  المفردات | | | | | | ILOs  مخرجات التعلم | | Hours  الساعات | | | WEEK  الاسبوع |
| Tutorial | | Workgroups, case studies | | | | **C# Review, Arrays, and Strings:**   * C# fundamentals review. * Working with arrays and List<T>. * String manipulation and basic pattern matching. | | | | | |  | | 4 | | | 1 |
| Tutorial | | Workgroups, case studies | | | | **Linked Lists**   * Implementing Singly and Doubly Linked Lists. * Practical exercises using linked lists. | | | | | |  | | 4 | | | 2 |
| Tutorial | | Workgroups, case studies | | | | **Stacks and Queues**   * Implementing Stacks and Queues. * Solving practical problems using Stacks and Queues. | | | | | |  | | 4 | | | 3 |
| Tutorial | | Workgroups, case studies | | | | **Recursion**   * Implementing recursive functions. * Debugging recursive code. | | | | | |  | | 4 | | | 4 |
| Tutorial | | Workgroups, case studies | | | | **Sorting Algorithms (Part 1)**   * Implementing Bubble Sort, Insertion Sort, Selection Sort. | | | | | |  | | 4 | | | 5 |
| Tutorial | | Workgroups, case studies | | | | **Sorting Algorithms (Part 2)**   * Implementing Merge Sort, Quick Sort. * Comparing sorting algorithm performance. | | | | | |  | | 4 | | | 6 |
| Med term exam | | | | | | | | | | | | | | | | | 7 |
| Tutorial | | Workgroups, case studies | | | | **Searching Algorithms**   * Implementing Linear and Binary Search. * Using C#'s built-in search methods. | | | | | |  | | 4 | | | 8 |
| Tutorial | | Workgroups, case studies | | | | **Trees (Part 1)**   * Implementing Binary Trees and BSTs. | | | | | |  | | 4 | | | 9 |
| Tutorial | | Workgroups, case studies | | | | **Trees (Part 2)**   * Implementing Tree Traversal algorithms. | | | | | |  | | 4 | | | 10 |
| Tutorial | | Workgroups, case studies | | | | **Graphs**   * Graph representations and traversal algorithms (BFS, DFS). | | | | | |  | | 4 | | | 11 |
| Tutorial | | Workgroups, case studies | | | | **Hash Tables**   * Implementing Hash Tables and using Dictionary<TKey, TValue>. | | | | | |  | | 4 | | | 12 |
| Tutorial | | Workgroups, case studies | | | | **Dynamic Programming**   * Implementing Dynamic Programming solutions. | | | | | |  | | 4 | | | 13 |
| Tutorial | | Workgroups, case studies | | | | **Greedy Algorithms and Review**   * Implementing Greedy Algorithms. * Course review and Q&A. | | | | | |  | | 4 | | | 14 |
| Final exam | | | | | | | | | | | | | | | | | 15 |

Reading List قائمة المصادر المطلوبة للقراءة

* McDowell, G. L. (2013). *Cracking the coding interview*. Iland Ebook
* Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
* Wengrow, J. (2020). A common-sense guide to data structures and algorithms: level up your core programming skills.

Course Requirements: متطلبات المقرر

Courses:

* CS17121
* CS17122

Grading التقييم

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Grading | Mid | Final | Workload | | | | | |
| Quizzes | Report | Seminars | Homework | … | Total |
| 25% | 50% | 15% | 5% | 0% | 5% |  | 25% |

Lecturer Name :( اسم المحاضر) Sipan Mohammed Hameed

Academic Title (اللقب العلمي): Assistant Lecturer

Email (الايميل): sipan.hameed@dpu.edu.krd

Lecture Time Table :( وقت المحاضرة)

Lecturer Office Hours :( اوقات الساعات المكتبية)

