MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

| **Module Information**  **معلومات المادة الدراسية** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module Title** | Data Structures | | | | **Module Delivery** | | |
| **Module Type** | Core | | | | * **☒ Theory** * **☐ Lecture** * **☒ Lab** * **☐ Tutorial** * **☐ Practical** * **☐ Seminar** | | |
| **Module Code** | ITC320090 | | | |
| **ECTS Credits** | 6 | | | |
| **SWL (hr/sem)** | 150 | | | |
| **Module Level** | | 2 | **Semester of Delivery** | | | | 3 |
| **Administering Department** | | BID | **College** | BMIC | | | |
| **Module Leader** | Dunia Hamid Hameed | | **e-mail** | [Dr.dunia.hamid@uoitc.edu.iq](mailto:Dr.dunia.hamid@uoitc.edu.iq) | | | |
| **Module Leader’s Acad. Title** | | Lecturer | **Module Leader’s Qualification** | | | | Ph.D. |
| **Module Tutor** |  | | **e-mail** |  | | | |
| **Peer Reviewer Name** | | jwan k alwan | **e-mail** | jwanism@uoitc.edu.iq | | | |
| **Scientific Committee Approval Date** | | 18/06/2023 | **Version Number** | | | 1.0 | |

| **Relation with other Modules**  **العلاقة مع المواد الدراسية الأخرى** | | | |
| --- | --- | --- | --- |
| **Prerequisite module** | Computer Programming II / BMI121 | **Semester** | 2 |
| **Co-requisites module** |  | **Semester** |  |

| **Module Aims, Learning Outcomes and Indicative Contents**  **أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية** | |
| --- | --- |
| **Module Aims**  **أهداف المادة الدراسية** | 1. To learn how to write algorithms and analyze problems. 2. To learn the fundamental principles of Data Structures. 3. To understand types of Data Structures. 4. To learn how to design and implement code of different Data Structures. 5. To apply the principles and concepts in the practical part. |
| **Module Learning Outcomes**  **مخرجات التعلم للمادة الدراسية** | 1. Understand Data types (primitive and non- primitive). 2. Learning Design of algorithms and analysis. 3. Understand the array representation and allocation. 4. Implement stack algorithms and understand its applications. 5. Implement queue algorithms and understand its applications. 6. Implement Single linked linear list algorithms. 7. Implement Circular Single linked linear list algorithms. 8. Implement Doubly linked linear list algorithms. 9. Implement Circular doubly linked linear list algorithms. 10. Implement Tree algorithms. 11. Demonstrate the Sorting and Searching algorithms. 12. Understanding File organization. |
| **Indicative Contents**  **المحتويات الإرشادية** | Indicative content includes the following.  Understanding data types and algorithms   * Introduction to data structures * Data structure types * algorithms   Principles of linear data structures   * Stack * Queue * Single linked linear list * Circular Single linked linear list * Double linked linear list * Circular Doubly linked linear list   Principles of Non- linear data structures   * Tree * Tree applications   Demonstration of Sorting and Searching algorithms   * Sorting Algorithms * Searching Algorithms   Understanding File Organization  Presentations (4 hours) |

| **Learning and Teaching Strategies**  **استراتيجيات التعلم والتعليم** | |
| --- | --- |
| **Strategies** | The module strategy for data structures concentrate on teaching the student how to use data types in programming and make the student familiar with the applications of data structures. |

| **Student Workload (SWL)**  **الحمل الدراسي للطالب** | | | |
| --- | --- | --- | --- |
| **Structured SWL (h/sem)**  **الحمل الدراسي المنتظم للطالب خلال الفصل** | 63 | **Structured SWL (h/w)**  **الحمل الدراسي المنتظم للطالب أسبوعيا** | 4 |
| **Unstructured SWL (h/sem)**  **الحمل الدراسي غير المنتظم للطالب خلال الفصل** | 87 | **Unstructured SWL (h/w)**  **الحمل الدراسي غير المنتظم للطالب أسبوعيا** | 6 |
| **Total SWL (h/sem)**  **الحمل الدراسي الكلي للطالب خلال الفصل** | 150 | | |

| **Module Evaluation**  **تقييم المادة الدراسية** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **As** | | **Time/Number** | **Weight (Marks)** | **Week Due** | **Relevant Learning Outcome** |
| **Formative assessment** | **Quizzes** | 2 | 10% (10) | 5, 10 | LO #1-5, 6-10 |
| **Assignments** | 2 | 10% (10) | 4,11 | LO # 4, 11 |
| **Projects / Lab.** | 1 | 10% (10) | Continuous |  |
| **Report** | 1 | 10% (10) | 15 | LO 1-9 |
| **Summative assessment** | **Midterm Exam** | 2 hr | 10% (10) | 7 | LO # 1-10 |
| **Final Exam** | 3hr | 50% (50) | 16 | All |
| **Total assessment** | | | 100% (100 Marks) |  |  |

| **Delivery Plan (Weekly Syllabus)**  **المنهاج الاسبوعي النظري** | |
| --- | --- |
| **Week** | **Material Covered** |
| **Week 1** | Introduction to data structures, definitions, types |
| **Week 2** | Algorithms Definition, Features, analysis reasons |
| **Week 3** | Linear Data Structures: Arrays |
| **Week 4** | Linear Data Structures: Stack, Stack Applications |
| **Week 5** | Linear Data Structures: Queue |
| **Week 6** | Linear Data Structures: Single Linked Linear List |
| **Week 7** | **Midterm Exam** |
| **Week 8** | Linear Data Structures: Circular Single Linked Linear List |
| **Week 9** | Linear Data Structures: Double Linked Linear List |
| **Week 10** | Linear Data Structures: Circular Double Linked Linear List |
| **Week 11** | Non Linear Data Structures: Tree |
| **Week 12** | Sorting Algorithms |
| **Week 13** | Searching Algorithms |
| **Week 14** | File organization |
| **Week 15** | Presentation of reports |

| **Delivery Plan (Weekly Lab. Syllabus)**  **المنهاج الاسبوعي للمختبر** | |
| --- | --- |
| **Week** | **Material Covered** |
| **Week 1** | Lab 1: Primitive data structures review |
| **Week 2** | Lab 2: Algorithm Example Greatest Common Divisor (GCD) |
| **Week 3** | Lab 3: Computing Location of specific element |
| **Week 4** | Lab 4: implementation of Stack Algorithms |
| **Week 5** | Lab 5: implementation of Queue Algorithms |
| **Week 6** | Lab 6: implementation of Single linked linear list Algorithms |
| **Week 7** | Lab 7: implementation of Circular Single linked linear list Algorithms |
| **Week 8** | Lab 8: implementation of double linked linear list algorithms |
| **Week 9** | Lab 9: implementation of Circular double linked linear list Algorithms |
| **Week 10** | Lab 10: implementation of tree Algorithms |
| **Week 11** | Lab 11: Sorting Algorithms |
| **Week 12** | Lab 12: Searching Algorithms |
| **Week 13-14** | Lab13: File organization |
| **Week 15** | Lab Exam |

| **Learning and Teaching Resources**  **مصادر التعلم والتدريس** | | |
| --- | --- | --- |
|  | **Text** | **Available in the Library?** |
| **Required Texts** | 1. "Problem-Solving with Algorithms and Data Structures using Python" by Bradley N. Miller, David L. Ranum, second Edition.  1. "Data Structures and Algorithms Made Easy: Data Structure and Algorithm Puzzles" by Narasimha Karumanchi. | No |
| **Recommended Texts** | "Introduction to Algorithms: A Creative Approach" by Udi Manber. | No |
| **Websites** | 1. <https://www.edureka.co/blog/data-structures-in-python/#datastructure> 2. <https://www.geeksforgeeks.org/data-structures/> 3. <https://www.javatpoint.com/data-structure-tutorial> | |

| **Grading Scheme**  **مخطط الدرجات** | | | | |
| --- | --- | --- | --- | --- |
| **Group** | **Grade** | التقدير | **Marks (%)** | **Definition** |
| **Success Group**  **(50 - 100)** | **A –** Excellent | **امتياز** | 90 - 100 | Outstanding Performance |
| **B -** Very Good | **جيد جدا** | 80 - 89 | Above average with some errors |
| **C –** Good | **جيد** | 70 - 79 | Sound work with notable errors |
| **D –** Satisfactory | **متوسط** | 60 - 69 | Fair but with major shortcomings |
| **E –** Sufficient | **مقبول** | 50 - 59 | Work meets minimum criteria |
| **Fail Group**  **(0 – 49)** | **FX –** Fail | **راسب (قيد المعالجة)** | (45-49) | More work required but credit awarded |
| **F –** Fail | **راسب** | (0-44) | Considerable amount of work required |
|  |  |  |  |  |
| **Note:** Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above. | | | | |