MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module Information**  **معلومات المادة الدراسية** | | | | | | | |
| **Module Title** | Data structure and algorithms هياكل البيانات و الخوارزميات | | | | **Module Delivery** | | |
| **Module Type** | Core | | | | * **☒ Theory** * **☐ Lecture** * **☒ Lab** * **☒ Tutorial** * **☐ Practical** * **☒ Seminar** | | |
| **Module Code** | COMP205 | | | |
| **ECTS Credits** | 8 | | | |
| **SWL (hr/sem)** | 200 | | | |
| **Module Level** | | UGII | **Semester of Delivery** | | | | 4 |
| **Administering Department** | | Computer Science | **College** | Education college for Pure Science | | | |
| **Module Leader** | Nada Ali Noori | | **e-mail** | Nada.ali@uobasrah.edu.iq | | | |
| **Module Leader’s Acad. Title** | | Assistant Lecturer | **Module Leader’s Qualification** | | | | MSc |
| **Module Tutor** | Nada Ali Noori | | **e-mail** | Nada.ali@uobasrah.edu.iq | | | |
| **Peer Reviewer Name** | |  | **e-mail** | E-mail | | | |
| **Scientific Committee Approval Date** | |  | **Version Number** | | | 1.0 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Relation with other Modules**  **العلاقة مع المواد الدراسية الأخرى** | | | |
| **Prerequisite module** | None | **Semester** |  |
| **Co-requisites module** | None | **Semester** |  |

|  |  |
| --- | --- |
| **Module Aims, Learning Outcomes and Indicative Contents**  **أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية** | |
| **Module Objectives**  **أهداف المادة الدراسية** | تهدف هذه المادة إلى تمكين الطالب من:   1. **فهم المفاهيم الأساسية لهياكل البيانات**    * التعرف على أهمية تنظيم البيانات في البرمجة.    * التمييز بين أنواع هياكل البيانات المختلفة مثل المصفوفات، القوائم المرتبطة، الأكوام، الطوابير، والأشجار. 2. **اختيار الهيكل المناسب للمشكلة البرمجية**    * مقارنة بين أداء وسلوك الهياكل المختلفة من حيث الكفاءة والذاكرة.    * تحديد الهيكل الأكثر ملاءمة لتطبيق معين. 3. **تطبيق العمليات الأساسية على هياكل البيانات**    * تنفيذ عمليات الإدخال، الحذف، البحث، والترتيب على الهياكل المختلفة.    * التعامل مع الهياكل الديناميكية باستخدام المؤشرات (Pointers). 4. **تصميم خوارزميات فعالة لمعالجة البيانات**    * استخدام هياكل البيانات لتطوير خوارزميات مرتبة ومنظمة.    * تحليل تعقيد الخوارزميات (Time and Space Complexity). 5. **البرمجة باستخدام لغة C++ أو لغة مناسبة أخرى**    * تنفيذ المفاهيم النظرية من خلال مشاريع وتطبيقات عملية باستخدام لغة برمجة.    * تعزيز مهارات التفكير المنطقي وحل المشكلات البرمجية. 6. **التمهيد لفهم مواد متقدمة**    * تهيئة الطالب لفهم مواضيع متقدمة مثل تحليل الخوارزميات، وهياكل البيانات المتقدمة (مثل الرسوم البيانية والهيب) |
| **Module Learning Outcomes**  **مخرجات التعلم للمادة الدراسية** | بنهاية هذه المادة الدراسية، يُتوقع من الطالب أن يكون قادراً على:   1. **شرح المفاهيم الأساسية لهياكل البيانات**    * تعريف أنواع هياكل البيانات واستخداماتها المختلفة في البرمجة. 2. **تحليل خصائص وكفاءة هياكل البيانات**    * تقييم كل هيكل بيانات من حيث الأداء (الزمن والمساحة) واختيار الهيكل الأنسب للمشكلة المطروحة. 3. **تطبيق العمليات الأساسية على الهياكل**    * تنفيذ خوارزميات للإدخال والحذف والبحث والترتيب باستخدام هياكل مثل: المصفوفات، القوائم المرتبطة (Singly & Doubly Linked Lists)، (Stacks)، الطوابير (Queues)، والأشجار (Trees). 4. **تصميم حلول برمجية باستخدام هياكل البيانات**    * بناء تطبيقات بسيطة تعتمد على تنظيم البيانات وتوظيف الهياكل المناسبة. 5. **كتابة كود برمجي فعال باستخدام لغة C++ أو لغة مشابهة**    * تحويل المفاهيم النظرية إلى تطبيقات عملية باستخدام البرمجة الكائنية. 6. **تحليل الخوارزميات المرتبطة بهياكل البيانات**    * حساب تعقيد الوقت والمساحة وتحسين أداء الخوارزميات. 7. **العمل الجماعي والتواصل التقني**    * التعاون مع الزملاء في مشاريع برمجية صغيرة، وعرض الحلول بطرق واضحة ومنهجية. |
| **Indicative Contents**  **المحتويات الإرشادية** |  **مقدمة في هياكل البيانات**   * تعريف وأهمية هياكل البيانات * تصنيف هياكل البيانات (خطية وغير خطية) * مفاهيم الذاكرة والمؤشرات    **المصفوفات (Arrays)**   * المصفوفات أحادية وثنائية الأبعاد * العمليات على المصفوفات * تطبيقات على المصفوفات    **القوائم المرتبطة (Linked Lists)**   * القوائم المرتبطة الأحادية * القوائم المرتبطة المزدوجة * القوائم الدائرية * الإدراج، الحذف، البحث    **الطوابير (Queues) والأكوام (Stacks)**   * مفهوم الكومة (Stack) وتطبيقاتها * مفهوم الطابور (Queue) وأنواعه (عادي، دائري، ذو أولوية) * تطبيقات الأكوام والطوابير في البرمجة    **الأشجار (Trees)**   * شجرة ثنائية (Binary Tree) * شجرة البحث الثنائية (Binary Search Tree) * التكرار (Traversal): Preorder, Inorder, Postorder * تطبيقات الأشجار    **الرسوم البيانية (Graphs)** (مقدمة فقط)   * تمثيل الرسم البياني (قائمة المجاورين، مصفوفة التجاور) * البحث في الرسوم البيانية (DFS, BFS)    **الجداول الهاشية (Hash Tables)**   * مفهوم التهشير (Hashing) * دوال التهشير وطرق معالجة التصادم    **تحليل الخوارزميات**   * دراسة خوارزميات الترتيب والبحث * استخدام Big-O notationفي تحليل وتقديم النتائج |

|  |  |
| --- | --- |
| **Learning and Teaching Strategies**  **استراتيجيات التعلم والتعليم** | |
| **Strategies** | تعتمد المادة على مجموعة متنوعة من الاستراتيجيات التعليمية التي تهدف إلى تعزيز الفهم العميق للمفاهيم النظرية وتطبيقها عمليًا، وتشمل:   1. **المحاضرات النظرية التفاعلية**    * تقديم المفاهيم الأساسية لهياكل البيانات بأسلوب تفاعلي.    * استخدام العروض التقديمية والأمثلة التوضيحية أثناء الشرح. 2. **الدروس العملية (المختبرات البرمجية)**    * تدريب الطلبة على كتابة وتنفيذ برامج بلغة C++ أو لغة برمجية أخرى.    * حل تمارين تطبيقية على كل نوع من هياكل البيانات. 3. **حل المشكلات (Problem Solving)**    * تكليف الطلبة بمسائل برمجية تتطلب التفكير المنطقي واختيار الهيكل المناسب لحلها.    * تشجيع النقاش الجماعي لتحليل الحلول. 4. **التعلم القائم على المشاريع (Project-Based Learning)**    * تنفيذ مشروع نهائي يستخدم عدة هياكل بيانات في تطبيق عملي.    * تقديم المشروع وعرض نتائجه أمام الزملاء. 5. **العصف الذهني والنقاش الجماعي**    * طرح سيناريوهات برمجية ومناقشتها داخل الصف.    * تعزيز التفكير النقدي والتعلم من الزملاء. 6. **المهام المنزلية والتقييمات الدورية**    * تقديم واجبات دورية لتعزيز الفهم والمتابعة.    * استخدام الاختبارات القصيرة والواجبات التكوينية لتقييم الأداء بشكل مستمر. 7. **استخدام أدوات التعليم الإلكتروني**    * الاعتماد على منصات تعليمية لمشاركة الموارد والواجبات.    * توفير محتوى تفاعلي ومقاطع فيديو تعليمية داعمة. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Workload (SWL)**  **الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا** | | | |
| **Structured SWL (h/sem)**  **الحمل الدراسي المنتظم للطالب خلال الفصل** | 93 | **Structured SWL (h/w)**  **الحمل الدراسي المنتظم للطالب أسبوعيا** | 6 |
| **Unstructured SWL (h/sem)**  **الحمل الدراسي غير المنتظم للطالب خلال الفصل** | 107 | **Unstructured SWL (h/w)**  **الحمل الدراسي غير المنتظم للطالب أسبوعيا** | 7 |
| **Total SWL (h/sem)**  **الحمل الدراسي الكلي للطالب خلال الفصل** | **200** | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Module Evaluation**  **تقييم المادة الدراسية** | | | | | |
| **As** | | **Time/Number** | **Weight (Marks)** | **Week Due** | **Relevant Learning Outcome** |
| **Formative assessment** | **Quizzes** | 2 | 10% (10) | 5 and 10 | LO #1, #2 and #10, #11 |
| **Assignments** | 2 | 10% (10) | 2 and 12 | LO #3, #4 and #6, #7 |
| **Projects / Lab.** | 1 | 10% (10) | Continuous | All |
| **Report** | 1 | 10% (10) | 13 | LO #5, #8 and #10 |
| **Summative assessment** | **Midterm Exam** | 2hr | 10% (10) | 7 | LO #1 - #7 |
| **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 structure -Non primitive data structure  -Linear data structure |
| **Week 2** | Array -One dimensional arrays and its operations |
| **Week 3** | Array -Two dimensional arrays |
| **Week 4** | Functions: -use of function, -Function details -function calling |
| **Week 5** | Recursively: Example on recursively functions |
| **Week 6** | Functions overloading, - Function templates |
| **Week 7** | String and Structure -Functions and purpose |
| **Week 8** | Pointers-Address of operator (&),-Deference operator(\*),Pointers and arrays |
| **Week 9** | Linked list:-Types of linked lists -Linked list implementation in c++ |
| **Week 10** | Operations on single linked list Examples -Comparison between array and linked list |
| **Week 11** | Stack -Implementation -Analysis of stacks -push operation  -pop operation, -top operation |
| **Week 12** | stack operations: -Infix, Postfix and prefix |
| **Week 13** | Queue: -Basic operations on queue -Program |
| **Week 14** | Sorting:-Bubble sort -Selection sort -Insertion sort-Quick sort -Merge sort |
| **Week 15** | Searching: -Linear search -Binary search |
| **Week 16** | **Preparatory week before the final Exam** |

|  |  |
| --- | --- |
| **Delivery Plan (Weekly Lab. Syllabus)**  **المنهاج الاسبوعي للمختبر** | |
| **Week** | **Material Covered** |
| **Week 1** | Array |
| **Week 2** | Recurssion Function |
| **Week 3** | String Functions |
| **Week 4** | LinkedList |
| **Week 5** | Stack |
| **Week 6** | Queue |
| **Week 7** | Example on Sorting and Searching |

|  |  |  |
| --- | --- | --- |
| **Learning and Teaching Resources**  **مصادر التعلم والتدريس** | | |
|  | **Text** | **Available in the Library?** |
| **Required Texts** | Data Structures and Algorithms  Alfred V. Aho, Bell Laboratories, Murray Hill, New Jersey  John E. Hopcroft, Cornell University, Ithaca, New York  Jeffrey D. Ullman, Stanford University, Stanford, California | No |
| **Recommended Texts** | Data structures and Algorithm Analysis in C++  Mark Allen Weiss , Florida International University , 2014 | No |
| **Websites** |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. | | | | |