MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module Information**  **معلومات المادة الدراسية** | | | | | | | |
| **Module Title** | Problem Solving and Programming | | | | **Module Delivery** | | |
| **Module Type** | Core | | | | * **☒ Theory** * **☐ Lecture** * **☒ Lab** * **☒ Tutorial** * **☒ Practical** * **☐ Seminar** | | |
| **Module Code** | PRSP121 | | | |
| **ECTS Credits** | 7 | | | |
| **SWL (hr/sem)** | 175 | | | |
| **Module Level** | | UGx11 UGI | **Semester of Delivery** | | | | 2 |
| **Administering Department** | | Type Dept. Code | **College** | Type College Code | | | |
| **Module Leader** | **Dr. Saif Ghassan** | | **e-mail** | saif.g.mohammed@uotechnology.edu.iq | | | |
| **Module Leader’s Acad. Title** | | Lecturer | **Module Leader’s Qualification** | | | | Ph.D. |
| **Module Tutor** | **Assist.Lect. Zahraa Abbas Hassan** | | **e-mail** | Zahraa.A.Alzubydi@uotechnology.edu.iq | | | |
| **Peer Reviewer Name** | | **Assist.Lect.Enas A.Raheem** | **e-mail** | enas.a.raheem@uotechnology.edu.iq | | | |
| **Scientific Committee Approval Date** | | 13/06/2023 | **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** | 1. Introduce the core ideas and guidelines of C++ programming to the class. 2. Improve students' comprehension of fundamental programming concepts such variables, data types, control structures, and functions. 3. Through programming assignments, strengthen students' abilities in algorithmic thinking and problem-solving. 4. Introduce students to the debugging and testing of C++ applications. 5. Encourage teamwork and collaboration through programming projects. 6. Prepare students for more higher-level of computer science and software engineering courses. |
| **Module Learning Outcomes**  **مخرجات التعلم للمادة الدراسية** | 1. Study and describe the basic ideas behind C++ programming, such as variables, data types, control structures, and functions. 2. Create C++ programs that address certain programming issues by applying your problem-solving and algorithmic thinking skills. 3. Offer the ability to debug and test C++ applications to find and fix bugs and errors. 4. In C++ programs, use file handling techniques to read from and write to files. 5. Manage faults and prevalent situations in C++ applications by using exception handling techniques. 6. To effectively address programming challenges, evaluate and choose the best data structures and techniques . 7. Use C++ to build programming projects in a team environment. 8. Consider the ethical implications of your coding habits and operate in a responsible and ethical manner. |
| **Indicative Contents**  **المحتويات الإرشادية** | Indicative contents provide an overview of the specific topics, concepts, and skills that will be covered within a module.indicative contents for a C++ programming module include:   1. Introduction to C++ Programming    * Overview of programming languages and the role of C++    * C++ programming environment setup (compiler, IDE)    * Basic syntax and structure of a C++ program   [15 hrs]   1. Variables, Data Types, and Input/Output Operations    * Declaring and initializing variables    * Understanding different data types in C++    * Input and output operations in C++   [15 hrs]   1. Control Structures and Decision Making    * Conditional statements: if, else, switch    * Looping structures: for, while, do-while    * Control statements: break, continue. [10 hrs] 2. Functions and Procedural Programming    * Defining and calling functions    * Passing arguments to functions and returning values    * Recursive functions and function overloading.   [15 hrs]  Revision problem classes [6 hrs]   1. Arrays and Strings    * Introduction to arrays and their manipulation    * Character arrays and string handling    * Multi-dimensional arrays   [15 hrs]  Project revision and monitoring. [7 hrs]   1. Pointers and Dynamic Memory Allocation    * Understanding pointers and their usage    * Dynamic memory allocation and deallocation    * Pointer arithmetic and pointer-related concepts.   [15 hrs] |

|  |  |
| --- | --- |
| **Learning and Teaching Strategies**  **استراتيجيات التعلم والتعليم** | |
| **Strategies** | Strategies for teaching mainly focus on facilitating student understanding and application of programming concepts. We summarize them below:   1. **Lectures:** In-class lectures are used to present theoretical concepts, syntax, and programming techniques. 2. **Interactive Discussions**: Engaging students in discussions fosters active learning and allows for clarification of concepts. Instructors can encourage student participation by asking questions, facilitating peer-to-peer discussions. 3. **Hands-on Programming**: Students are given opportunities to apply the concepts learned in lectures and practice programming techniques through coding exercises, programming projects, and problem-solving activities. 4. **Code Review and Feedback**: Providing constructive feedback on students' code helps them improve their programming skills. instructors can review students' code and offer guidance on code optimization and readability. 5. **Use of Visuals and Multimedia:** Incorporating visual aids, multimedia resources, and interactive tools can enhance understanding and engagement. 6. **Assessment and Feedback**: Regular assessments, including quizzes, tests, and examinations to show how well the students understand the subject. 7. **Practice and Revision Sessions**: Providing dedicated practice sessions and revision classes enables them to improve students’ comprehension and strengthen their information. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Workload (SWL)**  **الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا** | | | |
| **Structured SWL (h/sem)**  **الحمل الدراسي المنتظم للطالب خلال الفصل** | 108 | **Structured SWL (h/w)**  **الحمل الدراسي المنتظم للطالب أسبوعيا** | 7 |
| **Unstructured SWL (h/sem)**  **الحمل الدراسي غير المنتظم للطالب خلال الفصل** | 42 | **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 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** | The algorithm design and programming technique : Structure of a program, algorithms, and flowchart |
| **Week 2** | Variables and data types, constants, and basic input / output operators |
| **Week 3** | Types of conditional statements |
| **Week 4** | Iteration (Repetition) statements |
| **Week 5** | fundamentals of arrays in C++ |
| **Week 6** | Functions: The Advantage of the Functions, Function syntax |
| **Week 7** | Mid-term Exam |
| **Week 8** | Structures: Defining the structure variable, Accessing the members of the structure |
| **Week 9** | Structures with passing by value and passing by reference |
| **Week 10** | Pointers: The Address-of Operator &, Uses of pointers |
| **Week 11** | Classes I |
| **Week 12** | Classes II |
| **Week 13** | Special members |
| **Week 14** | Friendship and inheritance |
| **Week 15** | polymorphism |
| **Week 16** | **Preparatory week before the final Exam** |

|  |  |
| --- | --- |
| **Delivery Plan (Weekly Lab. Syllabus)**  **المنهاج الاسبوعي للمختبر** | |
| **Week** | **Material Covered** |
| **Week 1** | Lab 1: Introduction: Identifiers, Keywords, Constants , Variables and Data Types |
| **Week 2** | Lab 2: Selection (conditional) statement: if statement- if…else statements |
| **Week 3** | Lab 3: Selection (conditional) statement: - Nested if statements- Switch statement |
| **Week 4** | Iteration (Repetition) statements: - while statement do/while statement |
| **Week 5** | Lab 5: Iteration (Repetition) statements: - for statement Nested for statement |
| **Week 6** | Lab 6: Arrays initialization, one dimensional array, two dimensional array |
| **Week 7** | Lab 7: Mid-term Exam |
| **week 8** | Strings: one dimensional of characters, two dimensional of characters |
| **week 9** | Functions: declaration, calling and definitions |
| **week 10** | Passing arguments to functions : passing by value, passing by reference |
| **week 11** | Functions with default arguments , functions overloading |
| **week 12** | Structures: Accessing and manipulating structures members |
| **week 13** | Structures: arrays of structures ,passing by value, passing by reference and nested structures |
| **week 14** | Pointers: accessing the content of pointers |
| **week 15** | pointers & arrays, pointers & functions and pointer & structures |
| **week 16** | Preparatory week before the final Exam |

|  |  |  |
| --- | --- | --- |
| **Learning and Teaching Resources**  **مصادر التعلم والتدريس** | | |
|  | **Text** | **Available in the Library?** |
| **Required Texts** | Fundamentals of Electric Circuits, C.K. Alexander and M.N.O Sadiku, McGraw-Hill Education | Yes |
| **Recommended Texts** | DC Electrical Circuit Analysis: A Practical Approach  Copyright Year: 2020, dissidents. | No |
| **Websites** | https://www.coursera.org/browse/physical-science-and-engineering/electrical-engineering | |

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