MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module Information**  **معلومات المادة الدراسية** | | | | | | | |
| **Module Title** | Computer Programming I | | | | **Module Delivery** | | |
| **Module Type** | B (Basic learning activities) | | | | * **☐ Theory**   **☐ Lecture**   * **☒ Lab** * **☐ Tutorial** * **☐ Practical** * **☐ Seminar** | | |
| **Module Code** | AME122 | | | |
| **ECTS Credits** | 5 | | | |
| **SWL (hr/sem)** | 125 | | | |
| **Module Level** | | 1 | **Semester of Delivery** | | | | 2 |
| **Administering Department** | | AME | **College** | Type College Code | | | |
| **Module Leader** | Huda Hatam Dalef | | **e-mail** | huda@kecbu.uobaghdad.edu.iq | | | |
| **Module Leader’s Acad. Title** | | Lecturer | **Module Leader’s Qualification** | | | | Ph.D. |
| **Module Tutor** | Tuqa Hani Abd-Alamir | | **e-mail** | Tuqa.hani2105m@coeng. uobaghdad.edu.iq | | | |
| **Peer Reviewer Name** | |  | **e-mail** |  | | | |
| **Scientific Committee Approval Date** | | 01/06/2023 | **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. To provide a comprehensive understanding of C++ programming language. Students learn how the language syntax, data and file structures, input/output devices, and files. 2. To learn programming concepts such as variables, data types, control structures (conditionals and loops), functions, and basic algorithms. 3. To develop students' problem-solving skills and their ability to think computationally and start building their programming skills. 4. To learn how to write code to solve simple problems and automate tasks by logical thinking, algorithm design, debugging, and code documentation. 5. To gain exposure about software development practices, such as version control and testing.Top of Form |
| **Module Learning Outcomes**  **مخرجات التعلم للمادة الدراسية** | 1. Understand the definition of steps for computer programming. 2. Gain the skills of creation the flowchart of system. 3. familiar with the basics of C++ Program, Data Types, Variables, Assignment and Input Statements. 4. Learn I/O Streams, Predefined Functions and Output Formatting. 5. Acquire the skills of creating control structures for selection by using functions of Relational Operators, Logical Expressions, If/If…else and Block Statements. 6. Learn to design the control structures for repetition depends on various types of loops. 7. Manipulate various C++ datatypes, such as arrays includes One-Dimensional Array, Indexing, Array Searching and c-Strings. 8. Learn how to find the maximum and minimum in the parallel arrays, calculate the average of all the elements present in an array, merge two arrays and print a 2D array. 9. Create C++ Program for an Array of Pointers, Reference to a Pointer and Function Pointer. 10. Use memory appropriately, including proper allocation/deallocation procedures. 11. familiar with the C++ Searching and Sorting Programs by Search an Element in an Array (Linear Search / Binary Search) and Sort it. 12. Isolate and fix common errors in C++ programs. 13. Write small-scale C++ programs using the above skills.   As above, By the end of the Computer Programming I module, students will have a thorough knowledge of C++ language and would be able to implement it with ease. Students will come away with an understanding of when and why you might want to use C++ language over another language, how both “low-level” and more abstracted programming can help you, and how to best develop your own software projects in these languages. |
| **Indicative Contents**  **المحتويات الإرشادية** | Indicative content includes the following;   * Introduction to Computer Programming (definition of steps for computer programming with flowchart of view the steps of system). [5 hrs] * Introduction to C++ Programming Language (Basics of a C++ Program, Data Types, Variables, Arithmetic, Operators, Casting, Assignment and Input Statements). [5 hrs] * Basic Input and Output in C++ Language (I/O Streams, Predefined Functions and Output Formatting). [5 hrs] * Control Structures for Selection (Relational Operators, Logical Expressions, If/If…else and Block Statements). [10 hrs] * Control Structures for Repetition (while Looping, for Looping and do…while Looping). [10 hrs] * User-Defined Functions (create the functions, Value-Returning Functions, return Statements, Parameters and Overloading). [5 hrs] * Arrays/One-Dimensional Array (One-Dimensional Array, Indexing, Array Searching and c-Strings). [10 hrs] * Arrays/Parallel Arrays (check if two arrays are equal or not, find the maximum and minimum in an array, calculate the average of all the elements present in an array, merge two arrays and print a 2d array). [10 hrs] * Pointers of C++ language (C++ Program for an Array of Pointers, void Pointer, Reference to a Pointer and Function Pointer. [5 hrs] * C++ Searching and Sorting Programs (Search an Element in an Array (Linear Search / Binary Search), Sort an Array (Selection Sort / Bubble Sort / Insertion Sort), Merge Sort and Sort a String). [5 hrs] * Error and Records (Accessing struct Members, I/O structs and Arrays vs Structs). [5 hrs]Top of Form |

|  |  |
| --- | --- |
| **Learning and Teaching Strategies**  **استراتيجيات التعلم والتعليم** | |
| **Strategies** | The learning and teaching strategies in the Computer Programming I (C++ language) module involve lectures an introduction to the languages of C++, and topics that need to know in order to be an effective programmer in them. These include the syntax, compilers, debugging, working on C++ projects, object-oriented programming in C++, the power of generic programming, writing a compiled library, memory management, modern best practices, and other powerful tools available (such as threading / parallelism, new features of C++, and optimization techniques). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Workload (SWL)**  **الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا** | | | |
| **Structured SWL (h/sem)**  **الحمل الدراسي المنتظم للطالب خلال الفصل** | 78 | **Structured SWL (h/w)**  **الحمل الدراسي المنتظم للطالب أسبوعيا** | 5 |
| **Unstructured SWL (h/sem)**  **الحمل الدراسي غير المنتظم للطالب خلال الفصل** | 47 | **Unstructured SWL (h/w)**  **الحمل الدراسي غير المنتظم للطالب أسبوعيا** | 3 |
| **Total SWL (h/sem)**  **الحمل الدراسي الكلي للطالب خلال الفصل** | **125** | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Module Evaluation**  **تقييم المادة الدراسية** | | | | | |
| **As** | | **Time/Number** | **Weight (Marks)** | **Week Due** | **Relevant Learning Outcome** |
| **Formative assessment** | **Quizzes** | 3 | 15% (15) | 4, 7, 12 | LO #1- #3, LO #5 and #6, LO #8 - #11 |
| **Assignments** | 4 | 15% (15) | 3, 5, 6, 13 | LO #1 and #2, LO #3 and # 4, LO #5,  LO #6 - #13 |
| **Projects / Lab.** | 1 | 10% (10) | Continuous | All |
| **Report** |  |  |  |  |
| **Summative assessment** | **Midterm Exam** | 2hr | 10% (10) | 8 | LO #1 - #7 |
| **Final Exam** | 3hr | 50% (50) | 16 | All |
| **Total assessment** | | | 100% (100 Marks) |  |  |

|  |  |
| --- | --- |
| **Delivery Plan (Weekly Syllabus)**  **المنهاج الأسبوعي النظري** | |
| **Week** | **Material Covered** |
| **Week 1** | Introduction to Computer Programming (Steps & flowchart) |
| **Week 2** | Introduction to C++ Programming Language |
| **Week 3** | Basic elements of C++ Language |
| **Week 4** | Basic Input and Output in C++ Language |
| **Week 5** | Control Structures (Selection) |
| **Week 6** | Control Structures (Repetition) |
| **Week 7** | User-Defined Functions |
| **Week 8** | Namespaces and User-Defined Data |
| **Week 9** | Arrays (One-Dimensional Array) |
| **Week 10** | Arrays (Parallel Arrays) |
| **Week 11** | Pointers of C++ language |
| **Week 12** | C++ Searching and Sorting Programs |
| **Week 13** | Errors in C++ language |
| **Week 14** | Storage Classes in C++ language |
| **Week 15** | Records |
| **Week 16** | **Preparatory week before the final Exam** |

|  |  |
| --- | --- |
| **Delivery Plan (Weekly Lab. Syllabus)**  **المنهاج الأسبوعي للمختبر** | |
| **Week** | **Material Covered** |
| **Week 1** | Lab 1: Basic elements of C++ Language |
| **Week 2** | Lab 2: Basic Input and Output in C++ Language |
| **Week 3** | Lab 3: Control Structures (Selection) |
| **Week 4** | Lab 4: Control Structures (Repetition-For loop) |
| **Week 5** | Lab 5: Control Structures (Repetition-while/do.. while loop) |
| **Week 6** | Lab 6: Arrays (One-Dimensional Array) |
| **Week 7** | Lab 7: Arrays (Parallel Arrays) |

|  |  |  |
| --- | --- | --- |
| **Learning and Teaching Resources**  **مصادر التعلم والتدريس** | | |
|  | **Text** | **Available in the Library?** |
| **Required Texts** | C++ How to Program, Paul Deitel and Harvey Deitel, Nine edition, Phi Learning Private Limited, Delhi,2014 | Yes |
| **Recommended Texts** |  |  |
| **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. | | | | |