# MODULE DESCRIPTION FORM

# نموذج وصف المادة الدراسية

| Module Information<br>معلومات المادة الدراسية | | | | | |
|---|---|---|---|---|---|
| **Module Title** | **Programming Fundamentals** | | | **Module Delivery** | |
| **Module Type** | **Core** | | | ☒**Theory**<br>☒**Lecture**<br>☒**Lab**<br>☐**Tutorial**<br>☐**Practical**<br>☒**Seminar** | |
| **Module Code** | COM-121 | | | | |
| **ECTS Credits** | 8 | | | | |
| **SWL (hr/sem)** | 200 | | | | |
| **Module Level** | 1 | | **Semester of Delivery** | 2 | |
| **Administering Department** | com | | **College** | cos | |
| **Module Leader** | Taha Mohammed Hasan | | **e-mail** | dr.tahamh@uodiyala.edu.iq | |
| **Module Leader's Acad. Title** | Professor | | **Module Leader's Qualification** | Ph.D. | |
| **Module Tutor** | Name (if available) | | **e-mail** | E-mail | |
| **Peer Reviewer Name** | Name | | **e-mail** | E-mail | |
| **Scientific Committee Approval Date** | 01/06/2023 | | **Version Number** | 1.0 | |

| Relation with other Modules<br>العلاقة مع المواد الدراسية الأخرى | | | |
|---|---|---|---|
| **Prerequisite module** | **Introduction to Programming** | **Semester** | 1 |
| **Co-requisites module** | None | **Semester** | |

| Module Aims, Learning Outcomes and Indicative Contents<br>أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية | |
|---|---|
| **Module Objectives**<br>أهداف المادة الدراسية | The educational objectives of this course are<br>1- Demonstrate a thorough understanding of modular programming by designing programs that require the use of programmer-defined functions.<br>2- Demonstrate a thorough understanding of arrays by designing and implementing programs that search and sort arrays.<br>3- Demonstrate a thorough understanding of the object-oriented programming concepts of encapsulation, data abstraction and composition by designing and implementing classes including the use of overloaded functions and constructors.<br>4- Demonstrate a thorough understanding of the concept of pointers and dynamic memory allocation by designing and implementing programs using pointers and dynamic memory allocation.<br>5- Demonstrate a thorough understanding of the implementation of programmer-defined functions and classes by writing code, performing unit testing and debugging of multiple complex programs.<br>6- Demonstrate good documentation style in all of the programs written in this course.<br>7- Demonstrate proficiency in implementing data validation code, performing unit testing, and developing test plans while implementing robust solutions to the assignments in this course.<br>8- Demonstrate a thorough understanding of stream input/output for both console and files.<br>9- Demonstrate an understanding of the differences between C and C++ in the areas of strings, pass by reference/passing pointers, and structs by designing and implementing programs that use C strings, C++ strings, C language structs and classes. |
| **Module Learning Outcomes**<br>مخرجات التعلم للمادة الدراسية | Students will be exposed to the following concepts and/or skills at an introductory concepts level:<br>1- The analysis and design of programs based on requirements and performance considerations.<br>2- evaluation of various possible technical solutions.<br>3- object-oriented design consideration.<br>4- system integration.<br>5- program documentation.<br>6- program debugging procedures.<br>7- developing program testing plans.<br>8- consideration of program operating environment.<br>9- use of reusable software. |
| **Indicative Contents**<br>المحتويات الإرشادية | Introduction to the C++ programming language and its subset, Program structure, blocks, storage types, console and file I/O, functions, arrays, strings, pointers, call-by-reference, call-by-value, and dynamic memory |

| | |
|---|---|
| | allocation. The concept and use of classes will be covered in some detail. Differences between C, C++. Some new features in C++ will be introduced. |

## Learning and Teaching Strategies
استراتيجيات التعلم والتعليم

| | |
|---|---|
| **Strategies** | • Lectures<br>• Tutorials<br>• Problem solving<br>• Lab<br>• Case study<br>• Small project |

## Student Workload (SWL)
الحمل الدراسي للطالب محسوب لـ ١٥ اسبوعا

| | | | |
|---|---|---|---|
| **Structured SWL (h/sem)**<br>الحمل الدراسي المنتظم للطالب خلال الفصل | 109 | **Structured SWL (h/w)**<br>الحمل الدراسي المنتظم للطالب أسبوعيا | 7 |
| **Unstructured SWL (h/sem)**<br>الحمل الدراسي غير المنتظم للطالب خلال الفصل | 91 | **Unstructured SWL (h/w)**<br>الحمل الدراسي غير المنتظم للطالب أسبوعيا | 6 |
| **Total SWL (h/sem)**<br>الحمل الدراسي الكلي للطالب خلال الفصل | **200** | | |

## Module Evaluation
تقييم المادة الدراسية

| | | Time/Number | Weight (Marks) | Week Due | Relevant Learning Outcome |
|---|---|---|---|---|---|
| **Formative assessment** | **Quizzes** | 2 | 10% (10) | 4 and 9 | LO #1, #2 and #10, #11 |
| | **Assignments** | 2 | 10% (10) | 5 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** | 2hr | 50% (50) | 16 | All |
| **Total assessment** | | | 100% (100 Marks) | | |

## Delivery Plan (Weekly Syllabus)
المنهاج الاسبوعي النظري

| | **Material Covered** |
|---|---|

| | |
|---|---|
| **Week 1** | Introduction to computers & programming. |
| **Weeks 2,3** | Array:<br>• Array definition (Two-dimensional array).<br>• operations on Two- dimensional array (add, subtraction, multiplication and invers of array). |
| **Weeks 4,5,6** | Functions |
| **Week 7** | Mid-term Exam |
| **Weeks 8,9** | Introduction to Classes & Objects |
| **Weeks 10 and 11** | Searching, Sorting, Algorithm Performance Analysis |
| **Weeks 12 and 13** | Pointers, dynamic memory allocation |
| **Week 14** | More about Classes and OOP |
| **Week 15** | Recursion |
| **Week 16** | **Preparatory week before the final Exam** |

| **Delivery Plan (Weekly Lab. Syllabus)**<br>المنهاج الاسبوعي للمختبر | |
|---|---|
| | **Material Covered** |
| **Weeks 1 and 2** | Array (Two-dimensional array).<br>Operations on tow-dimensional array |
| **Weeks 3 and 4** | Decisions, Loops, Functions |
| **Weeks 5,6 and 7** | Classes |
| **Weeks 8 and 9** | Searching/Sorting |
| **Weeks 10,11 and 12** | Searching/Sorting |
| **Weeks 13 and 14** | Dynamic Arrays, Pointers |
| **Week 15** | Recursion |

## Learning and Teaching Resources
مصادر التعلم والتدريس

| | Text | Available in the Library? |
|---|---|---|
| **Required Texts** | • Programming in C++<br>Frank Vahid and Roman Lysecky<br>Available through the zyBooks website directly<br>• A C++ compiler and/or IDE. There are many out there, but the only two that are officially supported:<br>- CLion (on Windows and macOS)<br>- Visual Studio (Windows only) | Yes |
| **Recommended Texts** | ▪ Think Like a Programmer, An Introduction to Creative Problem Solving<br>V. Anton Spraul<br>ISBN: 978-1593274245<br>▪ A good text editor, such as:<br>Notepad++ (This is my personal favorite)<br>Sublime Text<br>Atom, or Vim, or anything else you might prefer | No |
| **Websites** | 1-http://www.cplusplus.com/<br>2-https://www.w3schools.com/cpp/ | |

## 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.