|  |  |  |
| --- | --- | --- |
| المملكة العربية السعودية  وزارة التعليم العالي  **جامعة أم القرى**  الكلية الجامعية بالجموم – قسم الحاسب الآلي |  | Kingdom of Saudi Arabia  Ministry of Higher Education  **Umm Al-Qura University**  University College in Al-Jamoum  Computer Dept. |

Course Specification

1. **Course number and name:** (2316318-4) Data Structures and Algorithms
2. **Credits and contact hours:** 4Credits

(Lecture: 4/week – Practical Session: Non)

1. **Instructor’s or course coordinator’s name:** Dr. Kheir Eddine Bouazza
2. **Text books**
3. **Main Text book:** Mark weiss, Data Structures and Algorithm Analysis in C++, 3rd Edition, Adisson-Wesley, 2006.
4. **Reference:** M.T. Goodrich and R. Tamassia, Data Structures and Algorithms in Java, 4th Edition, John Wiley and Sons, 2005.
5. **Specific course information**
6. **brief description of the content of the course (Catalog Description):**

This course covers fundamental principles of computer programming, with an

emphasis on basic concepts of data and their representations inside a computer (scalar, structured and dynamic). Abstract data structures including manipulation of arrays, linked lists, strings, stacks, queues, linear lists, circular lists, orthogonal lists, trees and graphs. Implementations of Sorting, searching algorithms, the use of recursive algorithms, binary search trees, and hash tables.

All of the mentioned concepts and techniques are studied using C++ language. It is important to note that this course emphasis is covering the abstract concepts and techniques of these algorithms rather than the practical and training programming course.

1. **prerequisites or co-requisites:** Advanced Programming (2316205-3)
2. **indicate whether a required, elective, or selected elective course in the program:** required
3. **Specific goals for the course**

The student will be able to:

1. Understand the abstract data types of linked lists, trees, stacks and queues, write stack, and queue applications.
2. Design and implement efficient algorithms for manipulating data structures.
3. Understand the characteristics of hash tables for access and retrieval.
4. Apply and analyze the learned data structures and algorithmic techniques to write efficient searching and sorting algorithms and other data manipulating to determine their efficiencies.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Course*  *Goals* | *Program Outcomes* | | | | | | | | | | |
| SOa | SOb | SOc | SOd | SOe | SOf | SOg | SOh | SOi | SOj | SOk |
| 1 | ✓ | ✓ |  |  |  |  |  |  | ✓ | ✓ |  |
| 2 | ✓ | ✓ | ✓ |  |  |  |  |  | ✓ | ✓ |  |
| 3 | ✓ | ✓ | ✓ |  |  |  |  |  | ✓ | ✓ |  |
| 4 | ✓ | ✓ | ✓ |  |  |  |  |  | ✓ | ✓ |  |

|  |  |
| --- | --- |
| **Relationship of Course Goals to the Program Student Outcomes** | |
| **SOa** | An ability to apply knowledge of computing and mathematics appropriate to the discipline   * *Students appreciate the use of mathematical proofs to reason about and compare the asymptotic complexity of various algorithms through the use of Big-Oh and other notations.* |
| **SOb** | An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.   * *Students can analyze the time and space requirements of a particular problem by performing asymptotic analysis.* |
| **SOc** | An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.   * *Students are asked to modify algorithms to produce different outputs or combine algorithms and data structures to offer new solutions e.g. search trees and sorting.* |
| **SOi** | An ability to use current techniques, skills, and tools necessary for computing practices.   * *The students learn how to use the most up-to-date libraries in the course-selected programming language to implement different data structures.* |
| **SOj** | An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.   * *Students are able to compare various algorithms through different criteria.* |

1. **Brief list of topics to be covered**

* Abstract Data Types
* Analysis of Algorithms
* Fundamental data structures
* Arrays, Pointers and Classes
* Recursion
* Linked Listed, Stack and Queue
* Trees, heaps and graphs
* Searching and sorting algorithms