|  |  |  |
| --- | --- | --- |
| 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:** (2316205-3) Advanced Programming
2. **Credits and contact hours:** 3 Credits

(Lecture: 2/week – Practical Session: 2/week)

1. **Instructor’s or course coordinator’s name:** Dr. Wael Deabes
2. **Text books**
3. **Main Text book:** Harvey M. Deitel, Paul, J. Deitel, Java How to Program, 9th Edition, Prentice Hall, 2012.
4. **Reference:** Y. Daniel Liang, Intro to Java Programming, 10th Edition, Pearson, 2014.
5. **Specific course information**
6. **brief description of the content of the course (Catalog Description):**

This course is to study the fundamental concepts and techniques necessary to write high-quality programs, including basic concepts of Object-Oriented programming, exception handling, and class libraries. Students will learn how to use inheritance, interfaces, exception handling, and how to incorporate graphical user interfaces (GUIs) into their programming applications. Students will also learn how to apply object-oriented design and programming principles to their programs. Typical assignments and projects include using built-in and programmer-defined classes to develop full-featured, easy-to-use programs. All of the mentioned concepts and techniques are studied using the Java language. It is important to note that this course is not a Java training course. The emphasis is on the concepts and techniques rather than the language itself.

1. **prerequisites or co-requisites:** Structured Programming (2316104-3)
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. Implement and design object-oriented programs that use advanced features and concepts such as encapsulation and inheritance.
2. Demonstrate an introductory understanding of graphical user interfaces, multi-threaded programming, and event-driven programming.
3. Write self-documenting code with an appropriate user interface that meets the style requirements for readability and usability.
4. Demonstrate basic knowledge of software engineering concepts.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *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 apply knowledge of object-oriented programming to solve programming problems.* |
| **SOc** | An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.   * *Students are required to design and develop programming projects to meet the desired requirements.* |
| **SOd** | An ability to function effectively on teams to accomplish a common goal.   * *Students work on teams to accomplish the building blocks of their projects.* |
| **SOi** | An ability to use current techniques, skills, and tools necessary for computing practices.   * *Students will use an IDE to facilitate the development process of the programming projects.* |
| **SOk** | An ability to apply design and development principles in the construction of software systems of varying complexity.   * *Students acquire the principles of OOD through the use of UML design principles and tools.* |

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

* Introduction to Object-Oriented concepts and design
* Overview of Java programming
* Data Types in Java
* Classes and Objects
* Exceptions and Input/Output
* Extending Classes
* Inheritance
* Encapsulation
* Graphical User Interface (GUI)
* Events