**King Saud University**

**College of Computer and Information Sciences**

**Department of Information Systems**

**IS 442 – Information Systems Engineering**

**Course Coordinator: Prof. Alaaeldin Hafez**

**Textbook(s) and/or Other Required Materials:**

**Primary: Roger S. Pressman and Bruce R. Maxim, Software Engineering, A practitioner’s Approach, McGraw-Hill, 8th Edition, 2015.**

**Supplementary**: Ian Sommerville, Software Engineering, Addison Wesley, 9th Edition, 2006

**Course Description (catalog):**

This course covers the following topics: the advanced steps in software developing cycle, such as software development methodologies, applying UML and patterns, software installation strategies, information systems maintenance, types of maintenance, software testing, user acceptance testing, testing metrics, measuring and controlling of maintenance effectiveness, software quality assurance, quality concepts, quality factors, technical metrics for software and examples of function-based specification quality, software sizing and costing, configuration and version management, and web engineering.

**Prerequisites:** IS 240 (Information Systems Analysis & Design)

**Co-requisite:** None

**Course Type:** Elective

**Course Learning Outcomes:** After completing this course, the students will be able to:

* Use professionally Unified Modeling Language (UML).
* Understand techniques of analyzing, designing, implementing, testing, and installing information systems.
* Estimate software size and cost by using state of the art techniques.
* Evaluate the quality of developed information systems
* Read and understand scientific literature
* Work in groups to efficiently apply software engineering techniques using applied information systems.

**Student Outcomes Covered by Course**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Student Outcome Description** | **Coverage** |
| (a) | An ability to apply knowledge of computing and mathematics appropriate to the discipline |  |
| (b) | An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution |  |
| (c) | An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs |  |
| (d) | An ability to function effectively on teams to accomplish a common goal | **√** |
| (e) | An understanding of professional, ethical, legal, security and social issues and responsibilities |  |
| (f) | An ability to communicate effectively with a range of audiences |  |
| (g) | An ability to analyze the local and global impact of computing on individuals, organizations, and society |  |
| (h) | Recognition of the need for and an ability to engage in continuing professional development |  |
| (i) | An ability to use current techniques, skills, and tools necessary for computing practice. | **√** |
| (j) | An understanding of processes that support the delivery and management of information systems within a specific application environment. | **√** |

**Major Topics covered and schedule in weeks:**

|  |  |  |
| --- | --- | --- |
| * Introduction to Software Engineering | 1 |  |
| * Applying UML and Patterns | 2 |  |
| * Management of Systems Implementation | 1 |  |
| * Software Project Sizing and Cost Estimation | 3 |  |
| * Software Quality Assurance | * (Self study) |  |
| * Software Testing Techniques | 3 |  |
| * Software Configuration Management | 2 |  |
| * Web Engineering | 2 |  |

**Course Instructor:** Prof. Alaaeldin Hafez

**Course Coordinator:** Prof. Alaaeldin M. Hafez

**Textbook(s) and/or Other Required Materials:**

#### Primary: Roger S. Pressman, Software Engineering, A practitioner’s Approach, McGraw-Hill, 7th Edition, 2010.

**Supplementary**: Ian Sommerville, Software Engineering, Addison Wesley, 6th Edition, 2001

**Course Description (catalog):**

This course covers the following topics: the advanced steps in software developing cycle, such as software development methodologies, applying UML and patterns, software installation strategies, information systems maintenance, types of maintenance, software testing, user acceptance testing, testing metrics, measuring and controlling of maintenance effectiveness, software quality assurance, quality concepts, quality factors, technical metrics for software and examples of function-based specification quality, software sizing and costing, configuration and version management, and web engineering.

**Prerequisites:** IS 240 (System Analysis and Design)

**Co-requisite:** None

**Course Type:** Elective

**Course Learning Outcomes:** After completing this course, the students will be able to:

* Use professionally Unified Modeling Language (UML).
* Understand techniques of analyzing, designing, implementing, testing, and installing information systems.
* Estimate software size and cost by using state of the art techniques.
* Evaluate the quality of developed information systems
* Read and understand scientific literature
* Work in groups to efficiently apply software engineering techniques using applied information systems.

**Student Outcomes Covered by Course**

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Student Outcome Description** | **Coverage** |
| (a) | 1. An ability to apply knowledge of computing and mathematics appropriate to the discipline |  |
| (b) | 1. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution |  |
| (c) | 1. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs |  |
| (d) | 1. An ability to function effectively on teams to accomplish a common goal | **√** |
| (e) | 1. An understanding of professional, ethical, legal, security and social issues and responsibilities |  |
| (f) | 1. An ability to communicate effectively with a range of audiences |  |
| (g) | 1. An ability to analyze the local and global impact of computing on individuals, organizations, and society |  |
| (h) | 1. Recognition of the need for and an ability to engage in continuing professional development |  |
| (i) | 1. An ability to use current techniques, skills, and tools necessary for computing practice. | **√** |
| (j) | 1. An understanding of processes that support the delivery and management of information systems within a specific application environment. | **√** |

**Major Topics covered and schedule in weeks:**

|  |  |  |
| --- | --- | --- |
| * Introduction to Software Engineering | 1 |  |
| * Applying UML and Patterns | 2 |  |
| * Management of Systems Implementation | 1 |  |
| * Software Project Sizing and Cost Estimation | 3 |  |
| * Software Quality Assurance | * (Self-Reading) |  |
| * Software Testing Techniques | 3 |  |
| * Software Configuration Management | 2 |  |
| * Web Engineering | 2 |  |

**Assessment Plan for the Course**

Assignments and Quizzes 20%

Mid-Term Exams 20%

Self Reading (Report) 5%

Project 15%

Final Exam 40%

**Total 100%**

All homework assignments are due one week after the assignment date.

* **No late** homework will be accepted.
* The quizzes may be pop or announced, and may be given at anytime during class-time
* Students are encouraged to discuss homework problems but **not copy**.
* Copying project or home assignments results in zero grading.
* **All exams are closed book.**
* **The final exam will be comprehensive.**

**Important Dates:**

Midterm Date

Project Submission Date