**SWE 321 –** SOFTWARE DESIGN AND ARCHITECTURE

**Instructor:** Dr. Zakarya Alzamil

**Office:** 2117

**Email:** zakarya@ksu.edu.sa

**Required Course:** 3 hours lecture and 1 hour tutorial/lab per week

**Course Description:**

This course covers the fundamental design principles and strategies for softwarearchitecture and design. Architectural styles and patterns, different types of software architecture such as data-centered, object-oriented, hierarchical, distributed (e.g., CORBA, SOA), component oriented, heterogeneous architecture. Architecture implementation, analysis, and designing non-functional properties are presented as well. Students participate in a group project on software design andarchitecture and design tools.

**Prerequisite(s):** SWE 312: Software Requirement Engineering

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

* Kai Qian et al. (2009): Software Architecture and Design Illuminated, Jones and Bartlett.(**primary**)

**Supplementary:**

* Richard Taylor et al. (2010): Software Architecture: Foundations, Theory, and Practice, John Wiley & Sons. (**supplementary**)
* David Garlan and Mary Shaw, An Introduction to Software Architecture, January 1994, CMU-CS-94-166.

**SWE Student Outcomes (SO) Addressed by the Course:**

|  |  |  |
| --- | --- | --- |
| **#** | **Outcome Description** | **Contribution** |
| **General Engineering Student Outcomes** |
| **(1)** | an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics | M |
| **(2)** | an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors  | H |
| **(3)** | an ability to communicate effectively with a range of audiences  | L |
| (4) | an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts  |  |
| **(5)** | an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives  | M |
| (6) | an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions  |  |
| **(7)** | an ability to acquire and apply new knowledge as needed, using appropriate learning strategies  | M |

**H**=High, **M**= Medium, **L**=Low

**Specific Outcomes of Instruction (Course Learning Outcomes):**

1. Be able to acquire knowledge about the **science of design** in the context of software engineering: the nature of design, design processes, design notations, design principles, design heuristics, and design styles. **[SO 7]**
2. Be able to analyze an engineering problem and to **choose** the most appropriate **design solution. [SO 1]**
3. Be able to design software systems **using the appropriate architectural styles**. **[SO 2]**
4. Be able to **apply** **different architecture styles**. **[SO 2]**
5. Be able to understand the link between the **non-functional requirements** for a system and its design goals. **[SO 2]**
6. Be able to use software design tools to effectively design the software system architecture and its sub elements. **[SO 1]**
7. Be able to design suitable user interfaces. **[SO 2]**
8. Be able to communicate effectively using correct English through a written report and project presentation. **[SO 3]**
9. Be able to function effectively on a team in a common project **[SO 5]**

**Major Topics Covered:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Topic** | **No.** | **Topic** |
| 1 | Introduction to Software Architecture | 2 | Software Architecture Design Space |
| 3 | Models for Software Architecture | 4 | Architecture View Models |
| 5 | Data Flow Architecture | 6 | Data Centered Software Architecture |
| 7 | Hierarchical Architecture | 8 | Implicit Asynchronous Communication Software Architecture |
| 9 | Interaction Oriented Software Architecture | 10 | Distributed Architecture |
| 11 | Component-Based Software Architecture | 12 | Heterogeneous Architecture |
| 13 | Architecture of Graphical User Interfaces | 14 | Software Architecture Implementation |

**Assessment & Evaluation Plan for the Course:**

Students’ performance is evaluated based on homework, quizzes, exams, and group projects.

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **No. of points** |
| ***Quizzes*** | Quizzes  | 10 points |
| ***Tutorial*** | Set of lab tutorials | 10 points |
| ***Project*** | Teamwork semester project  | 20 points |
| ***Exams*** | Midterm | 20 points |
| Final | 40 points |

**Contribution of Course to Meeting the Requirements of Criterion 5 (Curriculum):**

|  |  |
| --- | --- |
| **Curriculum Discipline** | **Contribution**  |
| General Education |  |
| Mathematics and Basic Sciences |  |
| Supporting Computing Topics |  |
| Core Software Engineering Topics: Science | 35 % |
| Core Software Engineering Topics: Design | 50 % |
| Major Design Experience | 15 % |

**Policies:**

#### Type your project deliverables, use some tools e.g., IBM Rational Software Architect, ArchStudio, Visio, etc.

#### Students can discuss homework, but no copying!

#### Late Submission Penalty (*50 % of earned points if submitted after the due date within one week, 0 if submitted late more than seven days of the due date).*

#### Plagiarism will never be tolerated and will result in failing the course according to the University bylaws.

**Calendar & Outline of Topics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Topics** | **Due Dates** |
| 1 | 1 September | Introduction to Software Architecture |  |
| 2 | 8 September | Software Architecture Design Space | Team Formation |
| 3 | 15 September | Models for Software Architecture |  |
| 4 | 22 September | Architecture View Models | Project deliverable 1 |
| 5 | 29 September | Data Flow Architecture  | Quiz 1 |
| 6 | 6 October | Data Centered Software Architecture |  |
| 7 | 13 October | Hierarchical Architecture | Project deliverable 2 |
| 8 | 20 October | Implicit Asynchronous Communication Software Architecture  | MIDTERM |
| 9 | 27 October | Interaction Oriented Software Architecture |  |
| 10 | 3 November | Distributed Architecture | Project deliverable 3 |
| 11 | 10 November | Component-Based Software Architecture | Quiz 2 |
| 12 | 17 November | Heterogeneous Architecture |  |
| 13 | 24 November | Architecture of Graphical User Interfaces |  |
| 14 | 1 December | Software Architecture Implementation |  |
| 15 | 8 December  | PROJECT PRESENTATION | Project Presentation & Final Report Submission |
|  | 15 December | FINAL EXAMS | According to the official Final Exams’ schedule |