🕮 Syllabus: CPIS 357 (Software Quality and Testing)

Course Description :

The aim and objective of this course is to teach students the concepts and skills needed for SQA and Testing. Software quality assurance (SQA or simply QA) is viewed as an activity that runs through the entire development process. It encompasses activities and related techniques to ensure the implementation of appropriate functionality that satisfy the requirements/needs of its targeted client/users for the intended software system, product, or service as the case may be, both correctly and efficiently.

General Course Objectives:

This course includes the study of the following broader areas:

1. Quality Assurance & Review Techniques,
2. Defect Prevention and Removal,
3. Testing Strategies,
4. Testing Conventional and Object Oriented Applications Techniques,
5. Acquaintance to Formal Modeling and Verification methods

Specific Course Objectives :

At the conclusion of this course (Theory & Lab.), students will be able to:

1. Understand general concepts about quality, quality assurance (QA), and software quality engineering (SQE)
2. Choose appropriate testing strategies and develop test cases
3. Understand how to detect, classify, prevent and remove defects
4. Be able to understand “Z” formally.
5. Write The Code for Unit and domain testing using JUnit framework and NetBeans as an editor.
6. Create control flow and data flow testing by using both control flow (graphs) and data flow diagrams respectively and hence getting a clear insight between control flow and data flow testing.

Detailed Course Schedule:

Week	Topics	Readings	Lab. Work
1	**Quality Management** – What is Software Quality – Quality Dimensions – The SQ Dilemma – Achieving Software Quality	Chapter 14 Lecture 14 Slides	Case Assignment #. 1 Describe Software Quality Assurance Methods Source: Internet
2	**Review Techniques** – Software Defects – Defect amplification and removal – Review Metrics and their use – Informal Reviews – Formal technical reviews – Review reporting and record keeping	Chapter 15 Lecture 15 Slides	Lab. Assignment #. 1 Perform Unit Testing using JUnit of Java in NetBeans Source: JAVA NetBeans
3	**Software Quality Assurance** – Elements of SQA – SQA Task, Goals and Metrics – Formal Approaches to SQA – Statistical SQA – Software Reliability – The ISO 9000 Quality Standards – The SQA Plan	Chapter 16 Lecture 16 Slides	
4			**Mid Term - 01 Examination**
5	**Software Testing Strategies** – The Strategic Approach for ST – Verification and Validation – Organizing for Software Testing – Criteria for Completing of Testing – Strategic Issues	Chapter 17 Lecture 17 Slides	Lab. Assignment #. 2 What are some major computer system failures caused by software bugs? Give 3 cases and briefly explain for each case. Source:

			Internet
6	**Testing Strategies for Conventional Soft-wares** – Unit Testing – Integration Testing – Strategies for OOS – Testing in OO context – Integration Testing in OO context – Strategies for Web Apps. – Validation Testing – Alpha and Beta Testing	Chapter 17 Lecture 17 Slides	Case Assignment #. 2 Write a simple java code (retrieve value from caller class and return the value to the caller class again Source: Internet
7	**System Testing** – Recovery Testing – Security Testing – Stress Testing – Performance Testing – Deployment Testing – The Art of Debugging	Chapter 17 Lecture 17 Slides	Lab. Assignment #. 3 Create Unit Test for above code (class) by using JUnit framework
8			**Mid Term - 02 Examination**
9	**Testing Conventional Applications** – White box testing – Basis Path Testing – Central Structure Testing – Black box Testing –	Chapter 18 Lecture 18 Slides	By using Net Bean, rewrite the source code below and execute JUnit to create unit test for Class Account. Source: JAVA NetBeans
10	– Model Based Testing – Testing Client Server Architectures – Testing for Real Time Systems – Patterns for Software Testing –	Chapter 18 Lecture 19 Slides	Case Study: Class Grade Book Class Grade Book which is used a two-dimensional array to store the grades of a number of students on multiple exams. In this case study, we use a then-

			by-three array containing ten students' grades on three exams.
11	**<u>Testing Object Oriented Applications</u>** – Testing OOA and OOD Models – Object Oriented Testing Strategies – Testing methods applicable at Classes – Interclass Test case Design – Test Derived from Behavior Models	Chapter 19 Lecture 20 Slides	Consider the given grade book class, and create unit test by using control flow testing approach. Create Test Cases to Satisfy Path Coverage for the given code.
12	<u>Testing **Web Applications**</u> – Dimensions of Quality – Content Testing – User Interface Testing – Component Level Testing – Navigation Testing – Configuration Testing – Security Testing – Performance Testing	Chapter 20 Lecture 20 Slides	Example: Using Control-flow testing to Test Program COUNT
13	**<u>Formal Modeling and Verification</u>** - *Cleanroom software engineering* – *The Cleanroom Process Model* – *The Cleanroom Strategy* – *Design Refinement & Verification* – *Certification & Certification Models* – *Formal methods* – *Formal Concepts* – Sets and Constructive Specification	Chapter 21 Lecture 21 Slides	Lab. Activity The passing score for any course at XYZ department is 60/100. If a student scores less than 60 for a course, the student gets an 'F' grade in the course. What variables could be involved in analysis of this group of facts? What variable do we know enough about to perform equivalence class analysis and then a boundary value analysis? Develop a series of tests by performing equivalence class analysis and boundary value analysis

			on this variable. Assume integer values.
14	**Final Exam**	Text book Lecture Slides Assignments	**Final Submissions:** Assignments Coursework Lab Activities

Student Responsibilities:

Students have a responsibility to attend classes and the Lab. sessions regularly. A lack of attention to regular attendance deprives students of interaction and exchange of ideas and knowledge. As many classes involve teamwork, students have an increased responsibility to attend regularly in support of team learning. The students also are responsible for knowing the contents of the course syllabus and reading them carefully to avoid any trouble.

Grading Policy:

There will be **NO bonus point** for any exams or project**.** No additional work will be given to raise grades**.** The marks given are final.

Collaboration:

The students are permitted to study in groups to prepare for examinations so long as the resulting exam demonstrates their individual mastery of the concepts and skills tested.

Group work:

The students are permitted to work in groups only for designated 'group projects,' which they are to submit as a group. All other assignments are to be prepared individually.

Faculty Information:

Course Instructor	:	Dr. Syed Faizan Haider - FCIT (KAU)
E-mail Address	:	shaider@kau.edu.sa
Contact Number	:	+966-69520000 Ext. 67508
Office Days / Hrs	:	Sunday - Tuesday (11:00am - 1:00pm)
Office Location	:	Room #. 137,Building #. 31

Course Schedule:

Lecture Days & Timings : Sunday, Tuesday and Thursday (1:00 - 1:50 pm)
Class Location : LAB#. 13,Building #. 31

Grading & Evaluation :

1. Final Exam - 30%
2. Mid-1 Exam - 15%
3. Mid-2 Exam - 15%
4. Quizzes, Assignments - 10%
5. Lab. Project/Activities – (5+ 25)%

Text book :

Software Engineering - A Practitioner's Approach (7th edition) by Roger S. Pressman. **ISBN 13. 9780073375977.**

Reference book :

Software Testing and Quality Assurance : Theory and Practice by Kshirasagar Naik, Priyadarshi

🕮 Coursework / Lab. Activities Assessment Scheme

Lab. Activities - CPIS357

Laboratory activities are a substantial piece of work requiring methodology and rules to be followed over an extended period of time, which is organized, evaluated and presented as a hardcopy of assignments or report. Lab. Instructor choose, in conjunction with their Students, a well-defined user-driven problem which enables them to demonstrate their skills in Software Testing, design and software quality assurance, including use of tools both for traditional and OOAD, documentation and evaluation. Problems to be selected, that allows the students to demonstrate their skills.

The Lab. Require the students to produce well-defined solutions or codes to the problems, involving a third-party user, and to generate a solution. This is done using software tools chosen by the instructor and may include an appropriate applications package or other software. Work on the lab activities begin in parallel with work on theoretical activities in lectures.

Lab. Activities - Scheme of Assessment

Supported by evidence of the use of Software Quality Assurance tools, hardcopies of work, including codes, graphs and tables will be graded.

Final Project Report

A report presenting activities (Posters & Charts) / Coursework as follows.

Content

- Organize the report into sections as given in the Lab Manual
- Word process the report
- Documentation of each stage of Test Cases Development

Learning outcomes

Candidates should be able to:

(a) Organize and develop reports for test cases
(b) Use word-processing features where appropriate including the use of a spellchecker
(c) Include the evidence of diagrams
(d) Include codes, tables, and cases

3. Guidance on marking the Activities / Coursework:

The Lab. activities are assessed as follows:

S.#	Project / Coursework Items	Marks
1.	Quality of Final Test Cases Report	5
2.	Use of the software tools for testing	5
3.	Activities, codes, and test cases submissions	20
	Total	**30**