Course Title:	Software Quality Engineering
Course Code:	SEN-309
Credit Hours Theory:	Three (3)
Credit Hours Lab (If Applicable):	0
Instructor Name with Qualification:	Tamim Ahmed Khan, PhD
Course Objectives:	This course aims to equip the students with a good grasp of software quality metrics and models. The students would learn software testing techniques and software reliability analysis techniques. The following is an itemized list of objectives: • to introduce quality assurance and quality control techniques and develop a QA plan and Test Plan • to be able to document and report the findings • to carry out inspections and carry out testing in a production environment
Course Learning Outcomes:	The students are able to start their carriers as quality engineering and as test professionals in leading software houses both domestic and international. More specifically: **Professional and reflective practitioner skills** 1. An understanding of the importance to introduce quality assurance and quality control techniques in software development process 2. To have a good grasp of software quality metrics and models 3. Be able to decide which of the software testing techniques are relevant for a particular case and know software reliability analysis tools and techniques **Practical skills** 4. To be able to develop a Quality Assurance plan and Software Test Plan, and 5. To be able to document and report the findings of a quality assurance cycle 6. To be able to conduct inspections and carry out testing in a production environment
Contents (Catalog Description):	This course introduces the student fundamental notions of software quality and the techniques used to build and check quality in software systems. A particular emphasis is placed on quantitative assessment of software quality and quality control using software testing techniques. The

<table>
<tr><td></td><td>students would not only be introduced with the theoretical background of these concepts but they would also be given hands-on experience of applying these concepts. The assignments would be planned carefully to enhance students' learning of applying the learnt concepts from practical standpoint.</td></tr>
<tr><td>Recommended Text Books:</td><td>P Ammann and J Offutt, Introduction to Software Engineering, Cambridge University Press, 2008</td></tr>
<tr><td>Reference Books:</td><td>"Ian Sommerville, Software Engineering, ninth edition, Addison Wesley, 2009
B Beizrer, Software Testing Techniques, 2nd Edition, The Coriolis Group,</td></tr>
<tr><td>Helping Web Sites:</td><td>http://www.softwarecertifications.org/qai_cste.htm
http://istqbexamcertification.com/</td></tr>
<tr><td>General Instructions for students:</td><td>
<u>Home Works and Assignments</u>
Attendance is mandatory. Every class is important. All deadlines are hard. Under normal circumstances late work will not be accepted. Students are required to take all the tests. No make-up tests will be given under normal circumstances. Any form of cheating on exams/assignments/quizzes is subject to serious penalty

<u>Attendance</u>
75% attendance is mandatory. Latecomers will be marked as absent.

<u>Evaluation Criteria</u>
Assignments/projects 20%
Quizzes 10%
Mid-Term 20%
Final 50%

<u>Quizzes Schedule</u>

<table>
<tr><td>Quiz # 1</td><td>Week # 4</td></tr>
<tr><td>Quiz # 2</td><td>Week # 7</td></tr>
<tr><td>Quiz # 3</td><td>Week # 11</td></tr>
<tr><td>Quiz # 4</td><td>Week # 14</td></tr>
</table>

<u>Assignments Schedule</u>

<table>
<tr><th>Assignment</th><th>Delivery date</th><th>Submission Date</th></tr>
<tr><td>Assignment # 1</td><td>Week # 2</td><td>Week # 4</td></tr>
<tr><td>Assignment # 2</td><td>Week # 5</td><td>Week # 6</td></tr>
<tr><td>Assignment # 3</td><td>Week # 9</td><td>Week # 10</td></tr>
<tr><td>Assignment # 4</td><td>Week #11</td><td>Week #12</td></tr>
</table>
</td></tr>
</table>

Sixteen Week Lesson Plan	The following is a broad outline of this course: 1.0 Software Quality: Overview and Introduction 1.1 Basic Introduction. 1.2 Software Quality Attributes. 1.3 Introduction to Quality Engineering. 2.0 Testing: Concepts, Issues, and Techniques 2.1 Introduction to software testing. 2.2 Software testing lifecycle. 2.3 Testing Scopes. 2.4 Testing Approaches. 2.5 Testing Concepts. 3.0 Test Planning Process 3.1 Introduction to testing process 3.2 Requirement of software test planning 3.3 Testing documentation 3.4 Reporting and historical data recording. 4.0 Software testing techniques 4.1 Testing philosophies 4.2 Testing strategies 4.3 Model based testing: 5.0 Software testing techniques (Continued…) 5.1 Testing using models: Using finite state machine 5.2 Control-flow and dataflow based testing 5.3 Domain and combinatorial testing 6.0 Unit and integration testing 6.1 Unit testing 6.2 Integration testing 6.3 Acceptance testing 7.0 Integration testing 7.1 Introduction to integration testing 7.2 Introduction to MM paths 7.3 Test automation 8.0 Slicing 8.1 Introduction 8.2 Different forms of slicing 8.3 Slicing and testing 9.0 Software reliability models and engineering 9.1 Introduction 9.2 Exponential model. 9.3 Reliability growth models 9.4 Modeling process 10.0 Software inspections 10.1 Fagan inspections 10.2 Software reviews 10.3 Inspection checks and metrics

	11.0 Quality Models 11.1 Introduction 11.2 Models for quality assessment 11.3 Product quality metrics 12.0 Quality Measurements 12.1 In-Process metrics for software testing 12.2 In-Process quality management 12.3 Effort/outcome models 13.0 System testing 13.1 Introduction to sub-system testing 13.2 From functional to system aspects of testing 13.3 14.0 System testing 14.1 Introduction to system testing 14.2 Scenarios development 15.0 System testing 15.1 Use-cases for testing 15.2 Specification-based testing 16.0 Open issues on software testing

Course Learning Outcomes mapping to Program Learning Objectives

CONTRIBUTION OF COURSE LEARNING OUTCOMES TO PROGRAMME LEARNING OUTCOMES													
SOFTWARE ENGINEERING		SOFTWARE QUALITY ENGINEERING											
No.	Program Outcomes	Course Learning Outcomes											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Engineering Knowledge	√											
2	Problem analysis		√										
3	Design/Development of solutions	√											
4	Investigation			√			√						
5	Modern tool usage												
6	Engineer and society												
7	Environment and sustainability												
8	Ethics												
9	Individual and Team work												
10	Communication												
11	Project Management				√	√							
12	Lifelong learning												