Basic Course Information **CS5860**

Course Title: **Software Metrics and Models**

Units: **3**

C/S Classification #: **C-2**

Component (select one): **Lecture**

Instructional Mode (select all appropriate choices): **Face-to-Face and Web-Assisted**

Grading Basis (select one): **Graded only**

Repeat Basis (select one): **May be taken only once**

Cross listed Course (if offered with another department):

Dual-listed Course (if offered as lower/upper division or undergraduate/graduate):

Major course/Service course/GE course (select all appropriate choices): **Major Course**

Prepared by: **Salam N. Salloum**

Date Prepared**: April 2, 2015**

I Catalog Description

**Measuring internal product attributes (size and Structure). Program decomposition (control-flow graph, prime components, sequencing and nesting). Hierarchical measures. Test coverage measures. Basis coverage du-path coverage. Complexity measures. Modularity and information flow measures. Coupling and Cohesion measures. Object-oriented metrics. Software cost estimation models. Software reliability models. Measuring external product attributes (modeling and measuring software quality).**

II Required Coursework and Background

Pre-requisite(s): **CS 4800 or consent of instructor**.

III Expected Outcomes

On successful completion of this course, students will be able to **acquire the following knowledge and skills:**

**1 Different measures for software size and expected software cost development;**

**2. Aspects of structural measures**

**3 Program decomposition, prime components, control flow graph, and hierarchical measures;**

**4 Different test coverage measures and related algorithms: basis, du-path, branch, statement, and simple loop coverages;**

**5. Different software quality measures;**

**6. Software reliability models.**

Outcomes of this course will build student capacity in each of the following areas as defined by programmatic objectives for the computer science major.

P-SLO 3. An ability to build applications, either individually or in a team, that are robust, reliable, and maintainable.

P-SLO 4. A breadth of advanced knowledge and skills in applied areas of computer science.

IV Instructional Methods

Texts may vary with instructor and over time. Examples of possible texts include:

**1. Norman E. Fenton and James Bieman, Software Metrics: A Rigorous & Practical Approach Third Edition, CRC Press, 2015.**

**2. Stephen H. Kan, Metrics and Models in Software Quality Engineering Second Edition, Addison-Wesley, 2003.**

V Minimum Student Material

**Textbook and Notebook**

VI Minimum College Facilities

**A classroom with a projection system, whiteboard, and a computer lab.**

VII Course Outline

**Software metrics data collection**

**Measuring internal product attributes: size and cost**

**Measuring internal product attributes: structure**

**Control flow graph, prime components and CFG decomposition**

**Hierarchical measures**

**Test coverage measures and related algorithms**

**Object-oriented structural attributes and measures**

**Measuring external product attributes (quality, usability, maintainability, and security)**

**Metrics for decision support**

**Bayes theorem and Bayesian networks**

**Reliability models**

VIII Instructional Methods

**Lecture**

**Problem-Solving**

**Discussion**

**Programming projects**

IX Evaluation of Outcomes

A. Student Assessment

* **Mid-term**
* **Homework**
* **Programming Assignments**
* **Presentation**

B. Meaningful Writing Assignment

* Students shall produce written solutions or proofs or programs to problems that are assigned as homework and/or programming projects and explain their reasoning.
* Short answer essay questions on exams will require students to explain and justify their response in writing.

C. A Matrix of Course Student Learning Outcomes vs Methods of Assessment

If the course is being evaluated for accreditation purposes, approved department accreditation assessment tools will additionally be utilized.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Course Learning Outcomes | Methods of Assessment | | | | |
| Class Participation | Homework | Programming Projects | Presentation | Exams |
| **1. Different measures for software size and expected software cost development** | x | x |  |  | x |
| **2. Aspects of structural measures** | x | x |  |  | x |
| **3. Program decomposition, prime components, control flow graph, and hierarchical measures** | x | x | x | x | x |
| **4. Different test coverage measures and related algorithms: basis, du-path, branch, statement, and simple loop coverages** | x | x | x | x | x |
| **5. Different software quality measures** | x | x |  |  | x |
| **6. Software reliability models** | x | x |  |  | x |