

Academic Org: Dept of Computer Sci & Engg – Subject: Computer Science

Course: CSCI3100 **Course ID:** 002582 **Eff Date:** 2022-07-01 **Crse Status:** Active **Apprv. Status:** Approved **【Course Rev】**
Software Engineering 軟件工程

This course introduces software life-cycles: system modelling, requirements analysis and specifications, design techniques, implementation methodology, testings, maintenance and engineering laboratory. Analytical tools: software metrics, system performance measurement and evaluation. Management techniques: estimations, planning, project management, communication skills and documentations. Introductions to CASE tools and security.

本科介紹軟件生命週期：系統模型化、要求分析及規格、設計技術、實踐方案、測試、維護及工程實驗。分析工具：軟件度量、系統性能之測量及評價。管理技術：估計、規劃、計劃之管理、通信技巧及文件編制。計算機輔助系統工程（CASE）導論及保密性。

Grade Descriptor: A

EXCELLENT – exceptionally good performance and far exceeding expectation in all or most of the course learning outcomes; demonstration of superior understanding of the subject matter, the ability to analyze problems and apply extensive knowledge, and skillful use of concepts and materials to derive proper solutions.

有關等級說明的資料，請參閱英文版本。

B

GOOD – good performance in all course learning outcomes and exceeding expectation in some of them; demonstration of good understanding of the subject matter and the ability to use proper concepts and materials to solve most of the problems encountered.

有關等級說明的資料，請參閱英文版本。

C

FAIR – adequate performance and meeting expectation in all course learning outcomes; demonstration of adequate understanding of the subject matter and the ability to solve simple problems.

有關等級說明的資料，請參閱英文版本。

D

MARGINAL – performance barely meets the expectation in the essential course learning outcomes; demonstration of partial understanding of the subject matter and the ability to solve simple problems.

有關等級說明的資料，請參閱英文版本。

F

FAILURE – performance does not meet the expectation in the essential course learning outcomes; demonstration of serious deficiencies and the need to retake the course.

有關等級說明的資料，請參閱英文版本。

Equivalent Offering:

Units: 3 (Min) / 3 (Max) / 3 (Acad Progress)

Grading Basis: Graded

Repeat for Credit: N

Multiple Enroll: N

Course Attributes:

Topics:

COURSE OUTCOMES

Learning Outcomes:

1. Students will know how to apply state of the art methodology in software design, development, measurement and evaluation for large-scale software systems;
2. Students will know what are the following software engineering techniques:
 - software management;
 - software requirement engineering;
 - specification techniques;
 - structured design;
 - Unified Modeling Language (UML);
 - Design Patterns;
 - structured programming;
 - top-down design and development;
 - segmentation and modularization techniques;
 - information hiding;

- iterative enhancement;
 - design and code inspection techniques;
 - correctness;
 - software validation and verification techniques;
 - software metrics;
 - software reliability measurement;
 - data collection and analysis;
3. Students will learn how to apply software engineering techniques for the development of large software projects.

Course Syllabus:

This course introduces software life-cycles: system modelling, requirements analysis and specifications, design techniques, implementation methodology, testings, maintenance and engineering laboratory. Analytical tools: software metrics, system performance measurement and evaluation. Management techniques: estimations, planning, project management, communication skills and documentations. Introductions to CASE tools and security.

Assessment Type:

Essay test or exam	: 40%
Lab reports	: 30%
Others	: 10%
Short answer test or exam	: 20%

Feedback for Evaluation:

1. Results of assignments and examination;
2. Course evaluation and questionnaire;
3. Reflection of teachers;
4. Question-and-answer sessions during class;
5. Student consultation during office hours or online;

Required Readings:

-

Recommended Readings:

1. Fundamentals of Software Engineering, Ghezzi, Jazayeri, and Mandrioli, Prentice Hall, 2nd Edition, 2003.
2. Software Engineering: A Practitioner's Approach, Pressman, McGraw-Hill, 6th Edition, 2005.
3. Software Engineering, Sommerville, Pearson/Addison Wesley, 7th Edition, 2004.
4. Software Engineering: Theory and Practice, Pfleeger, Prentice Hall, 2nd Edition, 2001.

5. Object-Oriented Software Engineering – Using UML, Patterns, and Java, Bruegge and Dutoit, Pearson/Prentice Hall, 2nd Edition, 2004.
6. Handbook of Software Reliability Engineering, Lyu (ed.), McGraw-Hill, 1996.

OFFERINGS

1. CSCI3100 Acad Organization=CSD; Acad Career=UG

COMPONENTS

LEC : Size=30; Final Exam=Y; Contact=3
TUT : Size=30; Final Exam=N; Contact=1

ENROLMENT REQUIREMENTS

1. CSCI3100 **Enrollment Requirement Group:**
Not for students who have taken ENGG3820 or ESTR3308 or IERG3080;
Pre-requisite: CSCI1110 or 1120 or 1130 or 1510 or 1520 or 1530 or 1540 or ESTR1100 or 1102.
For 2nd-year entrants, the prerequisite will be waived.

New Enrollment Requirement(s):
Pre-requisite = no change
Exclusion = no change

CAF

eLearning hrs for blended cls 0
No. of micro-modules 0
Research components (UG) 0%
University theme/ priority Innovation and Design

< E N D O F R E P O R T >