This course introduces the basics of algorithm analysis: correctness and time complexity. Techniques for designing efficient algorithms: greedy method, divide and conquer, and dynamic programming. Fundamental graph algorithms: graph traversals, minimum spanning trees and shortest paths. Introduction to complexity theory: polynomial-time reductions and NP-completeness.

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.

**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.

**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.
the ability to solve simple problems.

有关等級說明的資料，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.

有关等級說明的資料，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. Understanding of some fundamental algorithms;
2. Ability to design some simple algorithms;
3. Ability to analyze the correctness and time complexity of some simple algorithms;
4. Ability to construct simple reductions to demonstrate NP-completeness;

Course Syllabus:
This course introduces the basics of algorithm analysis: correctness and time complexity. Techniques for designing efficient algorithms: greedy method, divide and conquer, and dynamic programming. Fundamental graph algorithms: graph traversals, minimum spanning trees and shortest paths. Introduction to complexity theory: polynomial-time reductions and NP-completeness.

Assessment Type:
Essay test or exam: 50%
Others: 50%
Feedback for Evaluation:

The course will be evaluated by course evaluation done by the students.

Required Readings:

The recommended reading list/references will be determined by the instructor(s) of the course.

Recommended Readings:

The recommended reading list/references will be determined by the instructor(s) of the course.

---

OFFERINGS

1. CSCI3160
   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. CSCI3160
   Enrollment Requirement Group:
   Not for students who have taken ESTR3104 or CSCI3190;
   Pre-requisites: (CSCI2100 or CSCI2520 or ESTR2102) AND (CSCI2110 or ENGG2440 or ESTR2004 or ESTR2362 or MIEG2440)

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

CAF

<table>
<thead>
<tr>
<th>eLearning hrs for blended cls</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of micro-modules</td>
<td>0</td>
</tr>
<tr>
<td>Research components (UG)</td>
<td>50%-74%</td>
</tr>
</tbody>
</table>

<END OF REPORT>