CS4335: DESIGN AND ANALYSIS OF ALGORITHMS

Effective Term
Semester A 2022/23

Part I Course Overview

Course Title
Design and Analysis of Algorithms

Subject Code
CS - Computer Science

Course Number
4335

Academic Unit
Computer Science (CS)

College/School
College of Engineering (EG)

Course Duration
One Semester

Credit Units
3

Level
B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction
English

Medium of Assessment
English

Prerequisites
Nil

Precursors
CS2468 Data Structures and Data Management or CS3334 Data Structures or EE2331 Data Structures and Algorithms, or equivalent

Equivalent Courses
Nil

Exclusive Courses
Nil
Part II Course Details

Abstract
This course aims to introduce the algorithms in various domains, and techniques for designing efficient algorithms. It trains students the ability to analyse algorithms and the skills to design solutions to problems.

Course Intended Learning Outcomes (CILOs)

<table>
<thead>
<tr>
<th>CILOs</th>
<th>Weighting (if app.)</th>
<th>DEC-A1</th>
<th>DEC-A2</th>
<th>DEC-A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Prove the correctness and analyse the running time and performance</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2 Apply algorithmic paradigms and methods by using design techniques</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>3 Investigate the complexities of various problems in different</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>4 Propose new solutions for problems through independent study.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments
Demonstrate accomplishment of discovery/innovation/creativity through producing/constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Teaching and Learning Activities (TLAs)

<table>
<thead>
<tr>
<th>TLAs</th>
<th>Brief Description</th>
<th>CILO No.</th>
<th>Hours/week (if applicable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Lecture</td>
<td>General techniques will be taught in the lecture.</td>
<td>1, 2, 3, 4</td>
<td>3 hours per week</td>
</tr>
<tr>
<td>2 Tutorial</td>
<td>Exercises will be given in the tutorial and the lecturer (with the participation of students) will eventually give the answers.</td>
<td>1, 2, 3, 4</td>
<td>8 hours per semester</td>
</tr>
<tr>
<td>3 Assignment</td>
<td>Assignments contain problems that students should try to solve by adopting the best solutions.</td>
<td>1, 2, 3, 4</td>
<td></td>
</tr>
</tbody>
</table>
Assessment Tasks / Activities (ATs)

<table>
<thead>
<tr>
<th>ATs</th>
<th>CILO No.</th>
<th>Weighting (%)</th>
<th>Remarks (e.g. Parameter for GenAI use)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
<td>1, 2, 3, 4</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

Continuous Assessment (%)
30

Examination (%)
70

Examination Duration (Hours)
2

Additional Information for ATs
For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

Assessment Rubrics (AR)

Assessment Task
Assignment

Criterion
1.1 Each question is given a score

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Not even reaching marginal levels

Assessment Task
Examination

Criterion
2.1 Each question is given a score

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate
Part III Other Information

Keyword Syllabus

Syllabus
- 1. Algorithm analysis
  - Review on program correctness and complexities, and the mathematical tools for analysis.
- Graph algorithms
- String algorithms
  - String matching. Longest common subsequence. Dynamic programming.
- Theory of NP-completeness

Reading List
Compulsory Readings

<table>
<thead>
<tr>
<th>Title</th>
<th></th>
</tr>
</thead>
</table>

Additional Readings

<table>
<thead>
<tr>
<th>Title</th>
<th></th>
</tr>
</thead>
</table>