This course introduces the concept of abstract data types and the advantages of data abstraction. Various commonly used abstract data types including vector, list, stack, queue, tree, and set and their implementations using different data structures (array, pointer based structures, linked list, 2-3 tree, B-tree, etc.) will be discussed. Sample applications such as searching, sorting, etc., will also be used to illustrate the use of data abstraction in computer programming. Analysis of the performance of searching and sorting algorithms. Application of data structure principles.

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.

D

有關等級說明的資料，請參閱英文版本。
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. To be able to implement the following data structures as abstract data types in a high level programming language: stack, queue, hash table, list, binary search tree (including AVL tree, red black tree and splay tree), B-tree, trie, disjoint set, graph (including minimum spanning tree and shortest path);
2. To be able to use appropriate data structures in different applications;
3. To be able to implement abstract data types;
4. To be able to analyse the complexity of simple algorithms (such as searching and sorting);

Course Syllabus:

This course introduces the concept of abstract data types and the advantages of data abstraction. Various commonly used abstract data types including vector, list, stack, queue, tree, and set and their implementations using different data structures (array, pointer based structures, linked list, 2-3 tree, B-tree, etc.) will be discussed. Sample applications such as searching, sorting, etc., will also be used to illustrate the use of data abstraction in computer programming. Analysis of the performance of searching and sorting algorithms. Application of data structure principles.
Assessment Type:

- Essay test or exam: 50%
- Others: 25%
- Short answer test or exam: 25%

Feedback for Evaluation:

1. Mid-term course and teaching evaluation
2. Final course and teaching evaluation

Required Readings:

- 

Recommended Readings:

1. Horowitz, Sahni and Freed, Fundamentals of Data Structures in C
2. Weiss, Data Structures and Algorithm Analysis in C++
3. Goodrich and Tamassia, Data Structures and Algorithms in Java
4. Roberts, Programming Abstractions in C: A Second Course in Computer Science

OFFERINGS

1. CSCI2100
   Acad Organization=CSD; Acad Career=UG

COMPONENTS

<table>
<thead>
<tr>
<th>Component</th>
<th>Size</th>
<th>Final Exam</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEC</td>
<td>30</td>
<td>Y</td>
<td>3</td>
</tr>
<tr>
<td>TUT</td>
<td>30</td>
<td>N</td>
<td>1</td>
</tr>
</tbody>
</table>

ENROLMENT REQUIREMENTS

1. CSCI2100

   **Enrollment Requirement Group:**
   - Not for students who have taken ESTR2102 or CSCI2520;
   - Pre-requisite: AIST1110 or CSCI1110 or 1120 or 1130 or 1510 or 1520 or 1530 or 1540 or ESTR1100 or ESTR1102 or ESTR2306 or IERG2080.
   - For senior-year entrants, the prerequisite will be waived.

   **New Enrollment Requirement(s):**
   - Pre-requisite = Change from "CSCI1110 or 1120 or 1130 or 1510 or 1520 or 1530 or 1540 or ENGG1110 or ESTR1100 or ESTR1102 or ESTR2306 or IERG2080 or its equivalent" to "AIST1110 or CSCI1110 or 1120 or 1130 or 1510 or 1520 or 1530 or 1540 or ESTR1100 or ESTR1102 or ESTR2306 or IERG2080"
   - Co-requisite = no change
   - Exclusion = no change
<table>
<thead>
<tr>
<th></th>
<th>CAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>eLearning hrs for blended cls</td>
<td>0</td>
</tr>
<tr>
<td>No. of micro-modules</td>
<td>0</td>
</tr>
<tr>
<td>Research components (UG)</td>
<td>0%</td>
</tr>
</tbody>
</table>

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