Course Information

Course Code: 5670441  
Course Section: 1  
Course Title: DATA STRUCTURES  
Course Credit: 3  
Course ECTS: 5.0  
Course Catalog Description: Arrays, stacks, queues, linked lists, trees, hash tables, graphs: Algorithms and efficiency of access. Searching and sorting algorithms.

Prerequisites: Students must complete one of the following sets to take this course.

<table>
<thead>
<tr>
<th>Set</th>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5710230</td>
</tr>
<tr>
<td>2</td>
<td>5710229</td>
</tr>
</tbody>
</table>

Schedule: 
Monday, 09:40 - 10:30, EA208  
Wednesday, 13:40 - 15:30, EA208

Instructor Information

Name/Title: Prof. Dr. ŞENAN ECE SCHMİDT  
Office Address: A-402  
Email: eguran@metu.edu.tr  
Personal Website: http://users.metu.edu.tr/eguran/  
Office Phone: 210 4405  
Office Hours: None

Course Objectives

This course primarily aims to acquaint the student with basic data structures frequently used in software engineering and programming practices. Concepts of object-oriented programming, abstract data types, dynamic memory management and algorithm complexity are given. Searching and sorting algorithms are also discussed.

Course Learning Outcomes

i. Create awareness of and experience in object-oriented programming discipline.
   - Understand the meaning of and the usage of object oriented programming.
   - Learn basics of object oriented programming and class definition.
   - Learn recursive algorithms and argument passing.

ii. Create awareness of and experience in dynamic memory management.
   - Understand the meaning of and the usage of memory and dynamic memory.
   - Learn to create and delete dynamic memory.
   - Use dynamic memory in various data structures.

iii. Learn fundamental data structures with their searching, insertion and deletion capabilities.
   - Learn array data structure.
   - Learn stack and queue data structure.
   - Learn linked list and tree data structure.
   - Learn graph data structure.
   - Learn hash coding.
- Learn searching, insertion and deletion in the covered data

iv. Learn fundamental sorting algorithms with their complexity analysis.

- Understand the meaning of sorting.

- Learn sorting algorithms (selection sort, bubble sort, quick sort, radix sort).

- Compare the algorithms in terms of their complexity.

v. Create awareness of and experience in algorithm complexity.

- Understand the meaning of algorithm complexity.

- Learn Big-O, Ω, Θ complexity analysis details.

- Perform complexity analysis for various algorithms.

vi. Create awareness of and experience in computational complexity.

- Understand the meaning of computational complexity

- Learn decision problems, P, NP, NP complete, NP hard problem classes, relation between these classes and problem reduction

Instructional Methods
Lectures, programming assignments, exams

Tentative Weekly Outline

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Relevant Reading</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Object-Oriented Programming, Classes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Argument passing, references</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Pointers, arrays</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Algorithmic Complexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Stacks and Queues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Dynamic memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Linked lists</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Trees</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Sorting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Hashing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Computational Complexity</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Textbook(s)
Reference books:
(1) Preiss, B.R., Data Structures and Algorithms with Object-Oriented Design Patterns in C++, Wiley, 1999;
(2) Ford & Topp, Data Structures with C++, Prentice-Hall, 1999;

Course Material(s) and Reading(s)
Material(s)
Lecture notes on odtuclass.
Reading(s)
Lecture notes on odtuclass.

Supplementary Readings / Resources / E-Resources
Resources
by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

Assessment of Student Learning

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Dates or deadlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam</td>
<td></td>
</tr>
<tr>
<td>Quizes</td>
<td></td>
</tr>
<tr>
<td>Final Exam</td>
<td></td>
</tr>
<tr>
<td>Programming Assignment I</td>
<td></td>
</tr>
<tr>
<td>Programming assignment II</td>
<td></td>
</tr>
<tr>
<td>Programming assignment III</td>
<td></td>
</tr>
</tbody>
</table>

Course Grading

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Grade Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam</td>
<td>30</td>
</tr>
<tr>
<td>Final Exam</td>
<td>40</td>
</tr>
<tr>
<td>Programming Assignments, Quizes</td>
<td>30</td>
</tr>
<tr>
<td>Attendance (Bonus)</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>105</strong></td>
</tr>
</tbody>
</table>
Course Policies

Class Attendance
Class attendance is graded

Late Submission of Assignments
Late submissions of assignments will be penalized according to the following policy:

- 1 day late submission: HW will be evaluated out of 70.
- 2 days late submission: HW will be evaluated out of 50.
- 3 days late submission: HW will be evaluated out of 30.
- 4 or more days late submission: HW will not be evaluated.

It is NOT allowed to prepare homeworks as groups. METU honor code is essential.

To COPY or BEING COPIED will result in grade ZERO.

Make up for Exams and Assignments
Make-ups are to be given to those having medical report approved by METU medical center.

Final Exam Entrance Conditions
Students who miss all the exams, or who do not submit any HW will be graded as NA.

Other
It is not allowed:
- to use calculators, cell phones or other electronic devices
- going outside
during exams.

Academic Honesty
The METU Honour Code is as follows: "Every member of METU community adopts the following honour code as one of the core principles of academic life and strives to develop an academic environment where continuous adherence to this code is promoted. The members of the METU community are reliable, responsible and honourable people who embrace only the success and recognition they deserve, and act with integrity in their use, evaluation and presentation of facts, data and documents."