1. Course Title (Course Code)

Data Structure and Algorithms (2291)

2. Instructor

Muhammad Wannous, Ph.D.

3. Term

Spring 1

4. Outline and Objectives

Good organization of information (Data Structures) and logic flow (algorithms) ensures creating efficient programs. On the other hand, mastering the basic principles of clear design is very important for every programmer.

Computer science programs consider that gaining good programming skills starts with principles of fundamental software engineering. Then, once a programmer has learned the principles of clear program design and implementation, the next step is to study the effects of data organization and algorithms on program efficiency.

This course serves as an introduction to Data Structures and Algorithms. It describes many techniques for representing data and famous algorithms for solving common problems. The students will see:

- That each data structure and each algorithm has costs and benefits.
- That it is necessary to balance costs and benefits.
- Many common practices to avoid reinventing the wheel.
- That data structures follow needs.

5. Goals (Attainment Targets)

By the end of this course, students will be able to:

(1). Describe different data structures.
(2). Choose the data structures that model the information in a problem.
(3). Explain tradeoffs among alternative data structure implementations or combinations.
(4). Analyze common algorithms for searching and sorting.
(5). Apply and know when to use standard algorithms.
(6). Design, implement, and test programs using a variety of data structures.

6. Correspondence relationship between Educational goals and Course goals

Educational goals			Course goals
High level ICT skill	Basic academic skills		(1)
	Specialized knowledge and literacy		(1), (4)
Human skill (Tankyu skill)	Ability to continually improve own strengths		(3), (5)
	Ability to discover and resolve the problem in society	Problem setting	(2)
		Hypothesis planning	(2), (5)
		Hypothesis testing	(6)
		Practice	(2), (5)
	Fundamental Competencies for Working Persons	Ability to step forward	(3), (5)
		Ability to think through	(2), (5)
		Ability to work in a team	(6)
Professional ethics			(1)

7. Course Requirements

Programming skill (especially in an object-oriented programming language) is necessary for completing the practical examples.

A strong base in Math is also required for analyzing the performance of algorithms.

8. Textbooks

For this course, a set of lecture slides, handouts, and reports will be distributed in timely manner through Moodle.

9. Reference Books

The following book can enhance the experience of the students:

- Data Structures and Algorithm Analysis Edition 3.2 (Java Version)
  Free download (http://www.cs.vt.edu/~shaffer/book.html.)

10. Evaluation

Goals	Evaluation method & point					
	term-end exam	quiz	report	presentation	deliverable	other
(1)		○				
(2)		○				
(3)		○				
(4)		○				
(5)		○				
(6)			○			
Allocation		75	25			

Course Schedule

(Notice) This schedule is a tentative plan; there might be changes, additions, and revisions etc. at the time of delivering the course.

Lesson 1: Data structures and Algorithms (Lecture, 90 minutes)

This session includes these topics:

1. Orientation of the course and the syllabus.
2. Description of Data structures.
3. Abstract Data Types and Structures.
4. Problems, algorithms and Programs.

Lesson 2: Fundamentals of Data Structures [1] (Lecture, lab work, 90 minutes)

In this session we study about:

1. Lists
   - A) Array-Based List Implementation.
   - B) Linked Lists.
   - C) Comparison of List Implementations.
   - D) Element Implementations.
   - E) Doubly Linked Lists.

And analyze a sample code in Java on Lists.

Lesson 3: Fundamentals of Data Structures [2] (Lecture, lab work, 90 minutes)

In this session we study about:

1. Stacks
   - A) Array-Based Stacks.
   - B) Linked Stacks.
   - C) Comparison of Array-Based and Linked Stacks.
   - D) Implementing Recursion.

And analyze a sample code in Java on Stacks.

Lesson 4: Fundamentals of Data Structures [3] (Lecture, lab work, 90 minutes)

In this session we study about:

1. Queues
   - A) Array-Based Queues.
   - B) Linked Queues.
   - C) Comparison of Array-Based and Linked Queues.
2. Dictionaries.

And analyze a sample code in Java on Queues.

Lesson 5: Binary Trees [1] (Lecture, 90 minutes)

In this session we study about:

1. Binary Trees.
   - A) Definitions and Properties.
   - B) Binary Tree Traversals.

Lesson 6: Binary Trees [2] (Lecture, lab work, 90 minutes)

In this session we study about:

1. Binary Trees.
   - A) Binary Tree Node Implementations.
   - B) Binary Search Trees.
   - C) Heaps and Priority Queues

And analyze a sample code in Java on Binary Trees.

Lesson 7: Non-Binary Trees (Lecture, lab work, 90 minutes)

In this session we study about:

1. Non-Binary Trees.
   D) General Tree Definitions and Terminology.
   E) General Tree Traversals.
   F) The Parent Pointer Implementation.
   G) General Tree Implementations.
   H) K-ary Trees.
   I) Sequential Tree Implementations.

And analyze a sample code in Java on Non-Binary Trees.

Lesson 8: Review (lecture, discussion, 90 minutes)

In this session, we go through the main points we have studied so far in order to confirm what we have learnt and answer any questions about the concepts of Cloud Computing.

Lesson 9: Sorting and Searching [1] (Lecture, 90 minutes)

In this session we study about:

1. Sorting Terminology and Notation
2. Three $\theta(n^2)$ Sorting Algorithms
   A) Insertion Sort
   B) Bubble Sort
   C) Selection Sort
3. The Cost of Exchange Sorting

Lesson 10: Sorting and Searching [2] (Lecture, lab work 90 minutes)

In this session we study about:

1. Shellsort
2. Mergesort
3. Quicksort
4. Heapsort

And analyze a sample code in Java on Sorting and Searching.

Lesson 11: File Processing and External Sorting [1] (Lecture, 90 minutes)

In this session we study about:

1. Primary versus Secondary Storage.
2. Disk Drives
   A) Disk Drive Architecture
   B) Disk Access Costs
3. Buffers and Buffer Pools.
4. The Programmer's View of Files.

Lesson 12: File Processing and External Sorting [2] (Lecture, lab work, 90 minutes)

In this session we study about:

1. External Sorting.
   A) Simple Approaches to External Sorting.
   B) Replacement Selection.
   C) Multiway Merging.

And analyze a sample code in Java on Sorting and Searching.

Lesson 13: Searching [1] (Lecture, 90 minutes)

In this session we study about:

1. Searching Unsorted and Sorted Arrays.
   A) Self-Organizing Lists 307
   B) Bit Vectors for Representing Sets]

Lesson 14: Searching [2] (Lecture, lab work, 90 minutes)

In this session we study about:

1. Hashing
   A) Hash Functions.
   B) Open Hashing.
   C) Closed Hashing.
   D) Analysis of Closed Hashing.
   E) Deletion.

And analyze a sample code in Java on Hashing.

Lesson 15 Conclusion (discussion, 90 minutes)

This is the final session in which we try to recall the most important issues and concepts we learned in this course.