Course Specifications

Program(s) on which this course is given:	Systems and Biomedical Engineering
Department offering the program:	Systems and Biomedical Engineering
Department offering the course:	Systems and Biomedical Engineering
Academic Level:	Second year
Date	2013-2014
Semester (based on final exam timing)	☐ Fall ■ Spring

A- Basic Information

1. Title:	Computer Systems II			**Code:**	SBE 201			
2. Units/Credit hours per week:	Lectures	3	Tutorial	0	Practical	1	Total	4

B- Professional Information

1. Course description:

After completing the course the students are expected to have acquired basic knowledge in:

I. Acquire a firm knowledge of data structures and practical experience using their implementation.

II. Study the concept of abstract data types (ADTs) and the most commonly used ADTs in software development

III. Learn how data may be structured and instructions sequenced in algorithms and programs as well as the relationship between appropriate data and control structures and tasks from the "real world".

IV. Understand and contrast the operation of common data structures in terms of time complexity, space utilization, and the abstract data types they implement.

V. Learn how to assess how the choice of data structures and algorithm design methods influences the performance of programs.

VI. Learn how to choose the appropriate data structure and algorithm design method for a specified application.

VII. Practice tools, including analytical skills that will enable students to create programming solutions for real world problems.

2. Intended Learning Outcomes of Course (ILOs):

a) Knowledge and Understanding
Having successfully completed this module a student should have Knowledge & Understanding of:

1- Learn the Concepts and theories of data structures and practical experience using their implementation.

2- Understand the concept of abstract data types (ADTs) and the most commonly used ADTs in software development.

b) Intellectual Skills
Having successfully completed this module a student will have the ability to:

3- Analyze, and interpret data from the "real world" to decide on how data may be structured and instructions sequenced in algorithms and programs.

4- Determine the relationship between appropriate data and control structures and tasks.

5- Contrast the operation of common data structures in terms of time complexity, space utilization, and the abstract data types they implement.

c) Professional and Practical Skills

	Having successfully completed this module a student will have the ability to:
	6- Apply the principles of data structure to choose the appropriate data structure and algorithm design method for a specified application.
	7- Assess how the choice of data structures and algorithm design methods influences the performance of programs.
	8- Use laboratory computer and related software packages.
	d) General and Transferable Skills **Having successfully completed this module a student will have the ability to:**
	9- Work in stressful environment and within constraints while communicating effectively.
	10- Demonstrate efficient IT capabilities and effectively manage tasks, time, and resources.

3. Contents

Topic	Total hours	Lectures hours	Tutorial/ Practical hours
Part I: Introduction to Programming - (2 lectures) To learn basic skills and concepts of computer programming in structured approach using C-Language	8	6	2
Part II: Abstract Data Types (ADTs) - (4 lectures) To study the concept of Abstract Data Types (ADTs) and the abstract data types most commonly used in software development (stacks, queues, lists, sets, etc).	20	12	4
Part III: Basic Data Types - (4 lectures) To study the basic data types most commonly used to represent these Abstract Data Types (ADT such as arrays, linked lists, binary trees, etc), together with algorithms operating on these data structures.	20	12	4
Part IV: Algorithm Efficiency - (4 lectures) 1. C implementation of common algorithms, ADTs, data structures, and to study the concept of algorithm efficiency. 2. Learn how to choose the appropriate data structure and algorithm design method for a specified application. 3. Practice tools, including	20	12	4

analytical skills to create programming solutions for real world problems.			
Total	**56**	**42**	**14**
4. Teaching and Learning Methods	Lectures (*)	Practical Training/ Laboratory (*)	Seminar/Workshop ( )
	Class Activity (*)	Case Study ( )	Projects (*)
	E-learning ( )	Assignments/Homework (*)	Other:

5. Student Assessment Methods

• **Assessment Schedule**	**Week**
-Assessment 1; Class work	Every week
-Assessment 2; Project Assignment	13
-Assessment 3; Presentations	
-Assessment 3; Midterm Exam	8
-Assessment 4; Final Exam	16
• **Weighting of Assessments**	
-Mid-Term Examination	10%
-Final-term Examination	75%
-Project	10%
-Class work, Quizzes, and Assignments	5%
-Presentation	
-Total	100%

6. List of References

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "*Introduction to Algorithms*", The MIT Press, Third Edition, 2009

Recommended Textbooks:

Reema Thareja, "*Data Structures Using C*", Oxford University Press, USA, March, 2011

Charles F. Bowman, "Algorithms and Data Structures – An Approach in C", Oxford University Press, New York, 1994

Any C programming book such as Brian W. Kernighan and Dennis M. Ritchie. "The C programming Language*",* Published by Prentice-Hall in 1988

7. Facilities Required for Teaching and Learning

- Classroom White board (*)
- Classroom Laptop and data-show (*)
- Electronics Laboratory ( )
- Computer Laboratory (*)
- Others ( )

Course Coordinator:	**Prof. Dr. Ayman M. Eldeib**
Head of Department:	**Prof. Dr. Ahmed Badawi**