	Course Syllabus
Course Title: **Object Oriented Data Structures**	**Course code: 721221**
Course Level: 2	**Course prerequisite (s): 721220, 0250104** **Course corequisite (s): 721341, 750321, 750322, 750472, 760261, 711471**
Lecture Time:8:15 – 9:30 :ن ر	**Credit hours: 3**

		Academic Staff Specifics		
Name	**Rank**	**Office Number and Location**	**Office Hours**	**E-mail Address**
A. Obidat	Teacher	**IT306**	**ح ث خ 13-14** **ن ر 9:45-11:15**	aobidat@philadelphia.edu.jo

Course module description:

In this course students are exposed to fundamental programming constructs and data types such as lists, stacks, queues, tree, graphs…

An Outside-In approach will be adopted for this course. It consists in first teaching standard data types interfaces and their use to solve real life problems, and other specific domains like Games, Multimedia problems (Outside view). The implementation aspect of these standard data types will then be covered (Inside view). Moreover the theoretical lectures will be reinforced by a set of tutorials and laboratory sessions where simple and more complex problems are first solved then programmed.

Course module objectives:

The objectives of this module are to:

1- Provide a clear introduction to the concepts and the importance of standard data types.

2- Teach students how to use existing data types (through suitable APIs), adapt/modify or design new data types to solve simple and more complex problems.

3- Enhance the programming skills of students.

Course/ module components:

Text Books :

1.Data Structures Outside-In With Java, Sesh Venugopal, Prentice Hall, 2006

2 Problem Solving with Data Structures: A Multimedia Approach, Mark Guzdial and Barbara Ericson, 2009.

3. Data Structures &Problem Solving Using Java, mark allen weiss, fourth edition

Support material: Slides**,** Programming Tools

Teaching methods:

Duration: 16 weeks in semester, 64 hours in total)
Lectures: 32 hours (2 hours per week),
Tutorials: 16 hours, 1 per week,
Laboratories: 16 hours, 1 per week

Learning outcomes:

Knowledge and understanding

1- Recognize and understand the data type and data structure concept, and there importance for solving complex problems. (A2)

2- Recognize and deep understand for the interfaces of the standards data structures. (A2)

Cognitive skills (thinking and analysis)

3- Select the most appropriate data structure to solve particular problem.(B5)

4- Design, if not available, appropriate data structure to solve specific problem.(B3)

5- Modify and adapt existing data structure to deal with specific needs. (B3)

Practical skills

6. Write software code using appropriate software coding tools (C2)
7. Practice specific programming languages (C3).
8**.** Prepare and deliver coherent and structured verbal and written technical report. (C7)
9. Use the scientific literature effectively. (C8)
10. Use data structures libraries (like Java API). (C4)

Transferable skills

11. Solve problems (D3)
12. Use creativity (D2)
13. Communicate effectively with non-specialist as well as computer scientist, (D4)

Assessment instruments

- Final examination: (50 marks)
- Assignments (20 marks)

Allocation of Marks		
Assessment Instruments	**Date**	**Mark**
First examination		**20 %**
Second examination		**20 %**
Final examination: 40 marks		**40 %**

Assignments		**20 %**
Total		**100%**

** Make-up exams will be offered for valid reasons only with consent of the Dean. Make-up exams may be different from regular exams in content and format.*

Documentation and Academic Honesty

Submit your home work covered with a sheet containing your name, number, course title and number, and type and number of the home work (e.g. tutorial, assignment, and project).

Any completed homework must be handed in to my office (room IT ---) by 15:00 on the due date. After the deadline "zero" will be awarded. You must keep a duplicate copy of your work because it may be needed while the original is being marked.

You should hand in with your assignments:

1- A printed listing of your test programs (if any).
2- A brief report to explain your findings.
3- Your solution of questions.

For the research report, you are required to write a report similar to a research paper. It should include:

- **Abstract**: It describes the main synopsis of your paper.
- **Introduction**: It provides background information necessary to understand the research and getting readers interested in your subject. The introduction is where you put your problem in context and is likely where the bulk of your sources will appear.
- **Methods (Algorithms and Implementation)**: Describe your methods here. Summarize the algorithms generally, highlight features relevant to your project, and refer readers to your references for further details.
- **Results and Discussion (Benchmarking and Analysis)**: This section is the most important part of your paper. It is here that you demonstrate the work you have accomplished on this project and explain its significance. The quality of your analysis will impact your final grade more than any other component on the paper. You should therefore plan to spend the bulk of your project time not just gathering data, but determining what it ultimately means and deciding how best to showcase these findings.
- **Conclusion**: The conclusion should give your reader the points to "take home" from your paper. It should state clearly what your results demonstrate about the problem you were tackling in the paper. It should also generalize your findings, putting them into a useful context that can be built upon. All generalizations should be supported by your data, however; the discussion should prove these points, so that when the reader gets to the conclusion, the statements are logical and seem self-evident.
- **Bibliography:** Refer to any reference that you used in your assignment. Citations in the body of the paper should refer to a bibliography at the end of the paper.

- **Protection by Copyright**

1. Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.
2. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or

indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.

3. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.

- **Avoiding Plagiarism**.

1. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
2. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.
3. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.
4. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

Course/module academic calendar

week	Basic and support material to be covered	
(1, 2, 3)	**The Big Picture:** Data Structure definition, brief description for several Data Structures (*List, Queue, Stack, Trees, Heap, Hash Table and Graphs*), Abstract Data Types, using OOP and Java for Data Structures, choosing the Right Data Structure, Time complexity.	**Assignment 1**
(4, 5)	**Unordered List:** Unordered List case study, Unordered List Properties and operations, Unordered list implementation, a List Class, using List class, Linked List (Node, Insertion, Deletion and Access), Circular Linked List.	**Assignment 2**
First examination (6)	**Ordered List:** Ordered List case study, Binary Search, OrderedList Class, Ordered List implementation OrderedList Class Exceptions.	**Assignment 3**
(7)	**Queue:** Queue case study, Queue Properties and operations, Queue implementation, Queue Class, Using Queue class.	**Assignment 4**
(8, 9)	**Stack:** Stack case study, Stack Properties and operations, Stack implementation, Postfix Expression Evaluation using stack, Stack Class, using Stack Class. **Tutorial 1, Tutorial 2 and Tutorial 3**	**Assignment 5**
(10)	**Binary Tree and General Tree:** Tree Components, If-then-else tree, Expression tree, Recursive Definitions, Binary Tree Traversals, Binary Tree	**Assignment 6**

	implementation, Binary Tree Class, Huffman Coding.	
Second examination (11)	**Binary Search Tree :** Comparison Tree, Binary Search Tree Properties, Binary Search Tree Operations, Binary Search Tree implementation, Balancing, *BinarySearchTree* Class, Using *BinarySearchTree* Class *(Treesort, Counting Keys).*	**Assignment 7**
(12)	**Heap:** Heap as Priority Queue, Heap Properties, Max and Min Heaps, Heap Operations, Heap implementation, Heap Class, Priority Scheduling with Heap case study, Sorting with the Heap Class,	**Assignment 8**
(13, 14)	**Hash Table:** Motivation, Hashing, Collision Resolution, The *java.util.HashMap* Class, **Hash Table** implementation, Storage of Entries, Adding an Entry, Rehashing, Searching	**Assignment 9**
(15)	**Graphs :** Undirected Graphs, Directed Graphs, Weighted Graphs, Graph Representation, Graph Traversals (Depth-First Search, Breadth-First Search), Shortest Paths in a Weighted Directed Graph, Dijkstra's Single-Source Algorithm. **Tutorial 10: (implement Dijkstra's Single-Source Algorithm)**	**Assignment 10**
(16) Final Examination	**Revision and final exam**	

Expected workload:

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy:

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module references

Books

1- John Lewis, Joseph Chase, Java Software Structures: Designing and Using Data Structures, Addison-Wesley; 3 edition (February 23, 2009).

2- Maria Litvin, Gary Litvin, Java Methods: Object-Oriented Programming and Data Structures, Skylight Publishing; Second AP edition edition (December 15, 2010)

3- Frank M. Carrano, Data Structures and Abstractions with Java, Prentice Hall; 3 edition (September 23, 2011).

4- Goodrich and Tamassia. *Data Structures and Algorithms in Java, 2nd edition*. JohnWiley and Sons, Inc. 2000. ISBN 0471383678.

Websites

Htpp:// oodatastructures.jbub.com