CSC 16: Fundamentals of Computer Science II:
Data Structures, Algorithms and Object-Oriented Programming
Spring 2017

Classes: MW 12:50 – 2:15, Adams 200
Lab: Tuesday 2:05– 3:55 pm, Adams 204

Instructor: Dr. Simona Doboli
Office: 101 Adams Hall
Email: Simona.Doboli@hofstra.edu - the best way to reach me.
Phone: 516-463-4786
Office Hours: M: 9:30 - 11:00, Tu: 11:10 - 12:10, Tu: 4:00 - 4:30
Course materials: Blackboard

Bulletin course description: Continuation of CSC 15. Introduction to classes and objects. Investigates the essential properties of data structures, abstract data types, algorithms for operating them, use of these structures as tools to assist algorithm design. Introduces searching and sorting techniques. (3 hours lecture, 2 hours laboratory.)

Prerequisites: CSC 15 and CSC 14.

Textbook:

1. Larry Nyhoff: *ADTs*, *Data Structures and Problem Solving with C++,* Second Edition 2005, Person Prentice Hall.
2. Boswell, D. and Foucher, T. (2012), *The art of readable code*, O'Reilly.

Reference books (a C++ programming book):

1. Kyle Loudan: *C++ Pocket Reference,* O'Reilly, 2003
2. Deitel & Deitel: *C++ How to Program,* Prentice Hall.
3. S.B. Lippman, J. Lajoie: *C++ Primer.* Addison Wesley.

Reference web-site: www.cplusplus.com/**reference**/

Course Outline

1. Introduction to C++: variables, functions, arrays, pointers (Appendix C, Chapter 2 and 3).
2. Introduction to object-oriented programming (Chapter 4).
3. C++ string class (Chapter 5).
4. Lists (Chapter 6).
5. Stacks (Chapter 7)
6. Queues (Chapter 8)
7. Recursion (Chapter 10)

Course Learning Outcomes:

1. Design or select an appropriate algorithm for a particular problem.
2. Design or select an appropriate data structures for a particular problem.
3. Write programs that make good use of arrays, stacks, queues, and linked lists.

4. Design and write object-oriented programs.
5. Analyze the runtime performance of algorithms in terms of Big O notation.
6. Compare and contrast algorithms and data structures for a particular application.
7. Express an algorithm in pseudo-code.
8. Implement solution in C++ using 'readable' code and good programming techniques.
9. Debug and test a program.
10. Describe significant innovative software applications.

Disclaimer: Students may acquire different levels of competencies in the above outcomes. A good level requires regular attendance of classes and labs, completing assignments on-time, and respecting other course policies.

Something about me: I received my PhD in 2001 from University of Cincinnati with a thesis in computational neuroscience. Since Fall 2001, I have been a full-time faculty in the Computer Science Department. I usually teach Data Structures and Algorithms (CSC 16), Computer Architecture (CSC 110, CSC 110A), Artificial Intelligence (158), Robotics (159), Neural Cognitive Modeling. My current research is in neuro-cognitive models of idea generation and creativity, and data mining from large scientific and social networks.

Course components:

GradedAssignments (4-5)	**20 %**
Project (10th week)	**10 %**
News Flash	**5 %**
Lab attendance	**5 %**
Midterm 1 (March 6-8th Week 6)	**20 %**
Midterm 2 (April 10-12 Week 10)	**20 %**
Final Exam:	**20 %**

The course cannot be passed without any of the following: (1) 70% of assignments passed, (3) passing the final exam, (4) over 70% of laboratories attended, and (5) passing the project.

Grade Assignment:

100-96	A	90-95	A-		
85-89	B+	80-84	B	75-79	B-
70-74	C+	65-69	C	60-64	C-
55-59	D+	50-54	D	0 – 50	F

News flash: Each week for 15 minutes we will discuss news in computer and application technology with an emphasis on software applications. You have to look regularly for news in the computer industry and to submit a short description of it on the discussion board on Blackboard. You need to submit at least 2 submissions over the course of the semester.

> ***A submission needs to include:*** (a) a link to the web-site, (b) a short description of the application and the data/algorithms used, (c) what are the innovative aspects, (d) what is the potential for the application to become widely used, and (e) what do you see as shortcomings.

At least one submission must be from an article from IEEE Computer magazine, or Communications of the ACM magazine (see below).

Here is a list of relevant web-sites to start:

In the library on-line databases: IEEE CS digital library (Computer magazine), ACM digital library (Communications of the ACM).

!!! The list of the most interesting 20 web-sites for computer news - this is the place where to start:
http://computer-news.nettop20.com

http://www.businessweek.com/technology/computers/
http://www.zdnet.com/news
http://www.computerworld.com/
http://news.cnet.com/
http://www.pcmag.com/
http://www.pcworld.com/news.html
http://www.sciencedaily.com/news/computers_math/computer_science/

Assignments Policy:

1. Each assignment has to be turned in by the DUE DATE at the beginning of the class. Late assignments will carry an up to 20% penalty depending on the delay.

- If you turn in your assignment by the due date, you will be allowed to revise it after you receive the first grade together with suggested corrections/feedback. You will have one week to revise your assignment after you get it back. If you turn in your assignment in time your maximum grade will not be affected. If you turn in late, than the first grade becomes the final grade for that assignment.

- If you turn in your assignment late, you will not have a chance to resubmit with revisions.

- Given unforeseeable circumstances, you can choose up to two times during the semester when you can turn in an assignment late without late penalty as long as you are within a week from the due date.

2. All assignments must be turned in stapled. Loose pages will not be accepted.

3. For programming assignments you will have to turn in the following:

(1) Cover page: name, turn in date, due date, assignment number, revision or not.
(2) Pseudo-code of the algorithm and explanations for your chosen algorithm and data structures,
(3) Code.
(4) Relevant output test samples.
(5) Any other required written material, and
(6) Acknowledgements of any external sources human/web/written you have used to complete your assignment. All code must be written by you.

4. Programming assignments will be demonstrated during laboratory hours. You will have to be able to answer any questions about the program.

5. You should plan on spending 5-10 hours a week on programming assignments. Please make sure you reserve this time in your weekly schedule.

Grading policy for programming assignments:

The grade for a programming assignment is determined by:

- **Correctness of the algorithm:** 50 pts. Points will be deducted for logical errors, incomplete requirements
- **Programming Style and Documentation:** 20 pts. Poorly documented programs and unclear programming will be penalized. These points will not be given if the algorithm/program is incorrect.
- **Efficiency:** 15 pts. Points will be deducted for inefficient or hard to understand and maintain solutions.
- **Testing:** 15 pts. Testing cases must be chosen carefully, to cover all types of inputs and well documented.
- **Status of the implementation:** up to **-**20 pts. Points will be deducted for syntax errors (the program compiles with errors) and/or run-time errors (errors when the program executes). These points will be assigned only when the program is shown in lab to the instructor.
- **Delivery:** up to -20 pts. Points will be deducted for late assignments.

Project

The project will be a larger programming assignment where you will be asked to analyze a problem, design an efficient data structure and algorithm, write efficient, simple and correct code, and test your implementation. You will have no less than two weeks to turn in your assignment.

Collaboration policies for assignments

1. Assignments must be written individually. Same code/parts of code on more than one assignment will carry 0 points on each copy, for the first time, and will be reported as cheating for subsequent times. Just changing variable names will have the same fate.

2. Copying code off the internet is not considered your work and it will have the same outcome as copying from a colleague.

3. You are encouraged to discuss assignments among you. But each student has to write his/her own implementation.

4. You have to acknowledge in writing, at the beginning of your assignment, what kind of help you used (e.g. help with this solution, etc.) and the exact source of the help (tutor in labs, other students, web pages, etc.). Be careful, you cannot receive help for code writing, only for coming up with a solution. **Copied code is not accepted even if acknowledged.**

Quizzes:

There will be 5-6 quizzes during the semester. They will be announced the class before. The purpose of the quizzes is to check your basic understanding of the material covered in class.

Exam Policy:

All exams have to be taken ONLY at the day and time they are offered. Only University approved excuses that are properly documented will be accepted and only if they are announced at least one week before the exam day.

Other rules:

1. Attendance will be taken in the lab. It is part of the grade. Presence starts no later than 5 minutes after the lab starts.

2. You can leave the room in the middle of the class/lab only for emergency situations, and only once per class/lab. Please do not abuse this. If you have to leave the second time, please do not come back.

3. No hand-held electronic devices can be used in the class or lab. You will be asked to leave the class and not come back.

4. Appropriate civil behavior is expected in class and lab. Use of bad language or behaviors towards your fellow students, instructor or teaching assistant is strictly prohibited. You will be asked to leave the class and will be reported to the Dean of students.

Disclaimer: *The above schedule, policies, procedures and assignments in this course are subject to change in the event of extenuating circumstances by mutual agreement, and/or to ensure better student learning.*

Disabilities Policy

If you believe you need accommodations for a disability, please contact Services for Students with Disabilities (SSD). In accordance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990, qualified individuals with disabilities will not be discriminated against in any programs, or services available at Hofstra University. Individuals with disabilities are entitled to accommodations designed to facilitate full access to all programs and services. SSD is responsible for coordinating disability-related accommodations and will provide students with documented disabilities accommodation letters, as appropriate. Since accommodations may require early planning and are not retroactive, please contact SSD as soon as possible. All students are responsible for providing accommodation letters to each instructor and for discussing with him or her the specific accommodations needed and how they can be best implemented in each course.

For more information on services provided by the university and for submission of documentation, please contact the Services for Students with Disabilities, 212 Memorial Hall, 516-463-7075.

Academic Honesty

Plagiarism is a serious ethical and professional infraction. Hofstra's policy on academic honesty reads: "The academic community assumes that work of any kind [...] is done, entirely, and without assistance, by and only for the individual(s) whose name(s) it bears." Please refer to the "Procedure for Handling Violations of Academic Honesty by Undergraduate Students at Hofstra University" to be found at http://www.hofstra.edu/PDF/Senate_FPS_11.pdf , for details about what constitutes plagiarism, and Hofstra's procedures for handling violations.

Thank you for your patience in reading the whole syllabus and enjoy the class!!!