Data Structures

CSCI-UA.9102, Fall 2019

Instructor:	Nadim Kobeissi	**Office Hours:**	4:30pm – 5:00pm.
Email:	nk76@nyu.edu	**Office Number:**	Room 4.11.

1 Course Information

- *Course number and section:* CSCI-UA9102.
- *Course title:* Data Structures.
- *Course description:* Organizing and managing large quantities of data using computer programs is increasingly essential to all scientific and engineering disciplines. Students learn how to design data structures for representing information in computer memory, emphasizing abstract data types and their implementation, and designing algorithms using these representations. Topics include recursion, asymptotic analysis of algorithms, lists, stacks, queues, trees, hashing, priority queues, and graph data structures. This course is taught using **Java** programming language, and assumes that students have had at least one semester course in Java programming language (Introduction to Computer Science course). All programming assignments are given in Java and students need to be able to write fairly involved programs from the very first project. In addition to Java, many course materials will also be provided in the **Go** programming language. Any usage of Go is optional.
- *Prerequistes:* Passing CSCI-UA.0101 with a grade of C or better. You are expected to know and remember the material from CSCI-UA.0101 course. If you took the course a few semesters ago and/or do not remember parts of the material, start reviewing it during the summer/winter break. If you took an equivalent of this course at a different school, you need to make sure that you are familiar with Java. We assume you know enough Java to write fairly large programs right at the beginning of the semester.
- *Class meeting days and times:* Tuesdays and Thursdays, 5:45pm – 7:00pm. Room 4.11.
- *Recitations:* Thursdays, 7:15pm – 8:45pm. Room 4.11.
- *Term dates:* September 2, 2019 until December 13, 2019.
- *Course website:* `https://computerscience.paris/structures`.

2 Course Overview and Goals

Upon completion of this course, students will have a strong understanding of:

- Java features: abstract classes, interfaces, inheritance, polymorphism, generics, iterators, exception handling, file I/O.
- Core concepts: recursion, worst case asymptotic running time analysis, abstract data types, lists, stacks and queues, trees (binary and binary search trees, AVL trees), sorting algorithms, hashing and hash tables, and priority queues.

3 Course Requirements

- *Class participation:* You are expected to attend all classes. Missing a class can entail missing on important material. Discourse is encouraged during classes, but not mandatory.
- *Assigned readings:* Every lecture will be accompanied by outside readings that expand on what is discussed in class or present the same material in a different way. Neither the readings nor the lectures are a replacement for each other; deeply understanding the material will likely require attendance as well as reading. It is possible to read before or after class, depending on your learning style.
- *Problem sets:* Three problem sets will be assigned as homework. Problem sets must be submitted online *before* the start of class on the day that they are marked as due.
- *Recitations: "Recitations"* (which are essentially practical lab sessions) are scheduled once a week.
- *Exams:* A midterm exam and a final exam will be organized as part of this course.

4 Grading of Assignments

The grade for this course will be determined according to the following formula:
Class participation (10%), recitations (20%), problem sets (20%), midterm exam (25%) and final exam (25%).

Letter Grade	**Points**	**Description**
A	94	Outstanding
A-	90	Excellent
B+	87	Very Good
B	84	Good
B-	80	Satisfactory
C+	77	Above Average
C	74	Average
C-	70	Below Average
D+	67	Unsatisfactory
D	65	Low Pass
F	64	Fail

5 Course Schedule

Part 1: Java Topics . . . . ≈ 2 sessions

1.1: Documentation, Conventions
1.2: Inheritance, Polymorphism
Data Structures and Algorithms in Java, Chapters 1, 2

Part 2: Abstract Data Types . . . . ≈ 4 sessions

2.1: List Abstract Data Types
2.2: Array-Based Implementation of List ADTs
2.3: Referenced-Based Implementation of List ADTs
2.4: Linked Lists: Code Examples
Data Structures and Algorithms in Java, Chapter 3

Part 3: Recursion . . . . ≈ 5 sessions

3.1: Recursion: Definition, Base Case, Recursive Case
3.2: More on Recursion
3.3: Advanced Recursion with Backtracking
3.4: Midterm Exam Review
Midterm Exam
Data Structures and Algorithms in Java, Chapter 5

Part 4: Stack and Queue ADTs . . . . ≈ 3 sessions

4.1: Algorithmic Performance Analysis (Big O Notation)
4.2: Stack ADTs, Array and Reference-Based Implementation
4.3: Queue ADTs, Array and Reference-Based Implementation
Data Structures and Algorithms in Java, Chapters 4, 6

Part 5: Trees . . . . ≈ 6 sessions

5.1: Introduction to Trees
5.2: Binary Trees, Tree Traversal
5.3: Binary Search Trees: Algorithms and Implementations
5.4: Binary Search Trees: Removing an Element
5.5: Adelson-Velsky and Landis (AVL) Binary Search Trees
5.6: Caring for Your Binary Search Tree
Data Structures and Algorithms in Java, Chapters 8, 11.1, 11.2, 11.3

Part 6: Sorting . . . . ≈ 6 sessions

6.1: Quick Sort
6.2: Priority Queues and Heaps
6.3: Use of Heaps for Sorting
6.4: Hash Tables
6.5: Final Exam Review
Final Exam
Data Structures and Algorithms in Java, Chapters 12.1, 12.2, 9, 9.4, 10.1, 10.2

6 Required Textbooks and Materials

Aside from the textbooks and materials, students will also require their own personal computer for various parts of this course. Windows, Linux and Mac computers are all suitable.

6.1 Textbooks

- Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, *Data Structures and Algorithms in Java, 6th Edition (International Student Version)*, Wiley, 2014. ISBN: 978-1-118-80857-3.

- OpenDSA, a free online reference: `https://opendsa-server.cs.vt.edu/ODSA/Books/CS2/html/`

- GoDS, a free online reference: `https://github.com/emirpasic/gods`

6.2 Development Environment

All students are required to download and install Visual Studio Code (https://code.visualstudio.com/), a free code editor and programming environment for Windows, Linux and Mac computers. You will also need to download and install a Java Development Kit (JDK) and to configure Visual Studio Code to use your JDK (see *"Writing Java with Visual Studio Code"*: https://code.visualstudio.com/docs/java/java-tutorial) It is mandatory that you be able to run a simple Java "Hello World" program on your computer before the first class session.

Gopher! In addition to Java, many course materials will also be provided in the Go programming language. Any usage of Go is optional. You can install Go from the Golang.org website. For help with configuring Visual Studio Code for Go development, check out *"Go with Visual Studio Code"*: https://code.visualstudio.com/docs/languages/go.

7 Resources

- *Access your course materials:* NYU Classes (https://nyu.edu/its/classes).
- *Databases, journal articles and more:* Bobst Library (https://library.nyu.edu).
- *Assistance with strengthening your writing:* NYU Writing Center (https://nyu.mywconline.com).
- *Obtain 24/7 technology assistance:* IT Help Desk (https://nyu.edu/it/servicedesk).

8 Attendance and Tardiness

Study abroad at Global Academic Centers is an academically intensive and immersive experience in which students from a wide range of backgrounds exchange ideas in discussion-based seminars. Learning in such an environment depends on the active participation of all students. And since classes typically meet once or twice a week, even a single absence can cause a student to miss a significant portion of a course. To ensure the integrity of this academic experience, class attendance at the centers is mandatory, and unexcused absences will be penalized with a two percent deduction from the students final course grade for every week's worth of classes missed. Students are responsible for making up any work missed due to absence. Repeated absences in a course may result in harsher penalties including failure.

- Unexcused absences will be penalized with a 2% deduction from the students final course grade.
- Absences are excused only for illness, religious observance, and emergencies.
- *Illness:* For a single absence, students may be required to provide a doctors note, at the discretion of the Associate Director of Academics. In the case of two consecutive absences, students must provide a doctors note. Exams, quizzes, and presentations will not be made up without a doctors note.
- *Religious observance:* Students observing a religious holiday during regularly scheduled class time are entitled to miss class without any penalty to their grade. This is for the holiday only and does not include the days of travel that may come before and/or after the holiday. Students must notify their instructor and the Academic Office in writing via email one week in advance before being absent for this purpose. If exams, quizzes, and presentations are scheduled on a holiday a student will observe, the Associate Director, in coordination with the instructor, will reschedule them.
- *Contact your professor:* if you are unable to attend class, you are required to email your professors directly to notify them.
- *Late assignment:* Late submission or work will be accepted only with justifiable reasons of health or family emergency.

9 Academic Honesty

At NYU, a commitment to excellence, fairness, honesty, and respect within and outside the classroom is essential to maintaining the integrity of our community. Plagiarism is defined as presenting others' work without adequate acknowledgement of its source, as though it were ones own. Plagiarism is a form of fraud. We all stand on the shoulders of others, and we must give credit to the creators of the works that we incorporate into products that we call our own. Some examples of plagiarism:

- A sequence of words incorporated without quotation marks or an unacknowledged passage paraphrased from another's work.
- The use of ideas, sound recordings, computer data or images created by others as though it were ones own.
- Submitting evaluations of group members work for an assigned group project which misrepresent the work that was performed by another group member.
- Altering or forging academic documents, including but not limited to admissions materials, academic records, grade reports, add/drop forms, course registration forms, etc.

Furthermore, my courses have a **zero tolerance policy for cheating**. Any instance of cheating will result in an immediate, non-negotiable grade of 0 on the pertinent assignment and a report to the university faculty:

- Your code has to be your own. No copying code (or rewriting it line by line based on someone else's code) will be tolerated.
- Any sharing of any answers on any assignment is considered cheating.
- Coaching another student by helping them writing their answers line by line is also cheating.
- Copying answers or code from the Internet or hiring someone to write your answers for you is cheating.

Explaining how to use systems or tools and helping others with high-level design issues is not cheating.

For further information, students are encouraged to check NYU's Academic Integrity Policy.

10 Disability Disclosure Statement

Academic accommodations are available for students with disabilities. Please contact the Moses Center for Students with Disabilities (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.

11 About Your Instructor

My name is Nadim and it is my distinct privilege to be your instructor for this course. I am a researcher with a focus on applied cryptography, protocol analysis and formal verification. In designing and deploying real-world cryptographic systems in the public and private sector, I have always attempted to combine both theoretical and applied approaches to cryptography. I received my Ph.D. after doing research at the Institut National de Recherche en Informatique et Automatique (INRIA) in Paris (accredited by École Normale Superieure) and have published peer-reviewed research focusing on applied cryptography and automated protocol verification. I have also maintained several open source projects and have been involved in digital privacy issues.