

Spring 2020

CS 113-012: Introduction to Computer Science I (Revised for Remote Learning)

Junilda Spirollari

Follow this and additional works at: <https://digitalcommons.njit.edu/cs-syllabi>

Recommended Citation

Spirollari, Junilda, "CS 113-012: Introduction to Computer Science I (Revised for Remote Learning)" (2020). *Computer Science Syllabi*. 73.

<https://digitalcommons.njit.edu/cs-syllabi/73>

This Syllabus is brought to you for free and open access by the NJIT Syllabi at Digital Commons @ NJIT. It has been accepted for inclusion in Computer Science Syllabi by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

CS 113 - Introduction to Computer Science I

Course Syllabus, SPRING 2020

Overview

This course is a comprehensive introduction to the Java programming language teaching writing, testing and debugging of programs. The course has three major parts. The first part teaches fundamental programming techniques that use primitive data types, variables, assignments expressions and operators, control statements, arrays and files I/O. The second part covers testing and debugging and teaches students how to write programs that work reliably. The third part introduces object-oriented programming. The course guides students to the development of comprehensive Java applications.

Learning this material requires extensive hands-on practice. You should plan to spend twice as much time studying and working on problems outside of class, as you do in class.

Textbook

Java Software Solutions, 9th edition – John Lewis & William Loftus, ISBN-13: 9780134462028, Pearson. (One could use the older version, however it is his/her responsibility to make sure homework problems coincide with the ones assigned.)

Programming in Java with zyLabs – <http://learn.zybooks.com>

Prerequisites

CS100 – Roadmap to Computing or equivalent

Course Policies

Attendance is mandatory. A student who misses more than 5 classes will be dropped, without credit.

You must bring the textbook to class.

Moodle (<http://moodle.njit.edu/>) OR Canvas (<https://canvas.njit.edu/>) will be used for course discussion.

Homework must be submitted via Moodle/Canvas on the due date. They will not be accepted late except for special circumstances (such as jury duty or medical problem), for which you must provide documentation. All submitted work (including exams) must include your name and student ID.

Plagiarism will result in zero credit for the assignment and/or an XF grade in the course.

Cell phones must be turned off during class. During class time you may not play games, text, email, browse the web or engage in other activities that are not part of the class.

Students will be informed of any modifications of the syllabus during the semester.

Material covered

- Introduction to programming and Java programming language
- Data and Expressions
- Using Classes and Methods
- Decisions and Loops
- Arrays and File I/O
- Objects-Oriented Programming
 - a. Object-Oriented Design
 - b. Defining Classes and creating Objects
 - c. Defining methods
 - d. Inheritance
 - e. Polymorphism
- Recursion
- Exceptions

Learning Outcomes

Upon completing the course, the students will be expected to know and be able to use these elements to compute the solution to a problem:

- Understand the concept of classes and objects
- Design and implement own classes
- Create and use correctly object of different types
- Devise a sequence of steps (algorithm) that correctly solves a given problem.
- Write a program that implements the algorithm using:
 - A main set of java programming language elements (variables, syntax, keywords)
 - Data types (primitive and object data types including arrays)
 - Statements that perform input/output, control statements
 - Exception handling
- Understand inheritance and polymorphism and correctly use to solve complex problems
- Understand recursion and implement recursive methods

Evaluation

The evaluation will be based on the following course requirements:

- Homework 25%
- Midterms 1 and 2 20% each
- Final Exam 30%
- Misc. 5%

Midterm 2 Weight Distribution:

- 6% - multiple choice take-home quiz.
- 6% - challenge problems assigned on zyBooks
- 8% - lab assignments assigned on zyBooks

Final Exam Information

The final exam will take place Saturday, May 9th, 12:00pm – 12:30pm on Canvas. Lockdown Browser and Respondus Monitor will be used to deliver and proctor the exam.

Exam Policies

There are two common midterms: Monday, March 2 and Monday, April 6, 4:00-5:45pm. The final exam will take place Saturday, May 9th, 12:00pm – 12:30pm. **Be sure that you will be present for all of your final exams.**

You must bring a photo ID to all exams. Students with special needs are advised to make arrangements with Disability Services.

There are no makeup exams. If you miss a midterm because of a documented special circumstance you may receive an imputed grade based on the other midterm.

If you believe that you deserve more credit than you have been awarded on a particular exam problem, you may request, within 48 hours of the exam being returned, that it be regraded. Your entire exam will be regraded, which may result in points being added or subtracted.

Exams do not require any portable electronic devices, such as cell phones or calculators. In case such devices are brought to the exam room, **they must be left with the proctor for the duration of the exam.**

University Code on Academic Integrity

Every student should read the University Code on Academic Integrity (<http://www.njit.edu/academics/integrity.php>). All work that you represent as your own must, in fact, be your own. Work done by others must be given proper credit.

The following table shows the tentative agenda for the semester. It can be used to approximate how much time may be devoted to each topic and the corresponding reading from the suggested textbook.

Week (Approx)	Topic	Reading from book
	Introduction to programming and Java programming language	Ch. 1
1,2	Data and Expressions	Ch. 2
2	Using Classes & Objects,	Ch. 3
3	Writing Classes	Ch. 4
4	Writing Classes, Conditionals	Ch. 4, 5
5	Conditional & Loops	Ch. 5, 6
6	Object-Oriented Design (Testing & Debugging) + Review (Mid1)	Ch7
7	Arrays	Ch. 8
8	Arrays	Ch. 8
9	Recursion	Ch. 12
10	Recursion + Review (Mid2)	
11	Inheritance	Ch. 9
12	Polymorphism	Ch. 10
13	Exception Handling	Ch. 11
14	Review (Final)	Ch. 13