CS 113 - Introduction to Computer Science I
Course Syllabus, Spring 2015

Format

The format has lecture and lab components (recitations):

- One lecture plus one recitation per week.
- All students attend the same lecture class
- Students partitioned into smaller groups attend recitation

Lectures + Recitation

Lecture Class Meeting Time: Tuesday: 10:00 am – 11:25 pm
Room: CKB 204
Instructor: Dr. Narain Gehani

There are 3 recitation sections attached to the above lecture class. You will be in one of them:

- **Recitation Section 002**: Monday: 1:00 pm – 2:25 pm
  Room: GITC 2315A (Michael Baltrush)

- **Recitation Section 004**: Thursday: 4:00 pm – 5:25 pm
  Room: GITC 2400 (Michael Baltrush)

- **Recitation Section 006**: Thursday: 10:00 am – 11:25 pm
  Room: GITC 2400 (Junilda Spirollari)

Contact Information

Instructor: Dr. Narain Gehani
Office: GITC 4304
Email: gehani@njit.edu
Office Hours: Tuesday: 11:30 am – 1:00 pm

TA Contact Information

TA: TBA
Office: TBA
Email: TBA
Office Hours: TBA

Overview

This course is a comprehensive introduction to the Java programming language teaching writing, testing and debugging of programs. Topics covered fundamental Java object-oriented programming. Topics covered are primitive data types, variables, assignments expressions and operators, control statements, recursion, design and use of classes, arrays, and I/O. Other topics covered are testing and debugging and writing programs that work reliably. The course guides students to the development of comprehensive Java applications.

Textbook

Java Software Solutions, 8/E – John Lewis & William Loftus, ISBN-13: 978-0133594959, Addison-Wesley.

Please ignore the graphics portions at the end of each chapter. Graphics are not part of this course. Similar is the case with applets.

Reference Material

Java's strength comes from the large number of libraries. Language is relative simple – but made complex from the large library facilities For details of Java library facilities, please refer to

`docs.oracle.com/javase/8/docs/api/overview-summary.html`

Programming Environment

We will use jGRASP, an integrated development environment (IDE) that supports Java, for program development. This environment is supported at NJITs lab classrooms.

To acquire this environment for personal use, you can download it from

`www.jgrasp.org`

You will also need to install to install Java development kit (JDK). Follow Java (JDK) download instructions on the jGRASP download page – go to Oracle website and download Java.

`www.oracle.com/technetwork/java/javase/downloads`

You can also develop and run Java programs by accessing the Java compiler (`javac`) and interpreter (`java`) directly from the Windows *Command Window*. You need to edit Windows Environment variables PATH to **point** to the BIN directory in the Java software folder.

Prerequisites

CS100 – Roadmap to Computing or equivalent

Course Policies

- Attendance is mandatory.
- Moodle (moodle.njit.edu/) will be used for course communication. Please keep checking Moodle.
- **Homework assignments must be submitted in hard copy**.
- Homework assignments 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.
- Students will be informed of any modifications of the syllabus during the semester.

Material to be Covered

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

Course Goals

Upon completing the course, the students would accomplished the following:

1. Learn how to use core Java facilities with a focus on problem solving
2. Learn how to define classes (objects) and use them to write programs
3. Be able to write non-trivial Java programs.

.

Learning Outcomes

Students will be able to work with a customer (someone who wants a software/program to be developed) to understand the problem to be solved, design and write the (Java) program working by themselves.

Specifically, having determined the program to be written, students will be able to write Java programs that include programming concepts / Java facilities declarations, conditional and loop statements, define classes, methods (including recursive methods), handle errors, and debug/test programs for correct behavior.

Students will also be prepared with Java programming skills (as above) for advanced courses.

What do the Assignments Accomplish?

By doing assignments,

1. a student will develop problem solving expertise,
2. write Java programs putting to practice the programming concepts / Java facilities learned,
3. learn object-oriented programing, and
4. learn about software requirements and testing.

Writing Java programs will involve the use of declarations, conditional and loop statements, define classes (including inheritance), methods (including recursive methods), handle errors, and debugging / testing programs for correct behavior.

Performance

Assignments and the exams aim to assess

1. Java programming skills in the context of the use of Java facilities (as mentioned above) to solve problems, and
2. understanding of programming concepts.

Evaluation

Evaluation (final grade) will be based on the following items:

• Prerequisite Quiz	1%
• Attendance	5%
• Homework	30%
• Midterm Exam	31%
• Final Exam	33%

Please note that scores entered on Moodle are the raw scores for each individual item – not allocated as per the above percentages.

Note: In case the prerequisite quiz is not given, then the Midterm Exam will count for 32%.

Exam Policies

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

Only one midterm.

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 final exam.

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, and all such devices must be put away and turned off during the exam. Cell phones must be on silent and cannot be answered during the exam.

University Code on Academic Integrity

Read the University Code on Academic Integrity (njit.edu/academics/integrity.php). All work that you represent as your own must be your own. Work done by others must be given proper credit.

Tentative Weekly Coverage of Material

The following table shows approximately how much time may be devoted to each topic. **Actual class lectures may vary in pace and order**. Recitations will supplement lectures. Students should also supplement learning by reading in the book topics covered in class. A reading list is given following the table:

Week (Approx)	**Lecture**	**Recitation**
1	**Introduction to programming and Java programming language**	
2	**Basics of Java** programs	
3	**Software Development using Stepwise Refinement**	
3	**Nuts & Bolts of Java** programs …	
4	**Nuts & Bolts of Java** programs …	
5	**More Software Development using Stepwise Refinement**	
5	**Printing output**	
6-7	**Java Statements**	

7	**Strings**	
8	**More Software Development using Stepwise Refinement**	
9	**Arrays + Program Development using Stepwise Refinement**	
9	**Object-Oriented Programming**	
10	**Object-Oriented Programming**	
10	**Printing Output – details about printf**	
11	**Testing & Debugging + Passing Arguments**	
11	**Data structures**	
12	**More about classes – Inheritance, Abstract Classes**	
13	**Recursion, Searching, Sorting**	
14	**Exceptions**	
14	**More Data Structures**	

Some Important Dates

Exam Dates

Midterm	Tuesday March 10, 2015. **Only one midterm – in lecture class – no common exam midterms.**
Final	TBD by the Registrar

Assignment Due Dates

Assignment	**Due date is your first recitation after the date listed below.** **Hand in and pick up graded assignment in recitations.**
1	January 21
2	February 10
3	February 24
4	March 10
5	March 31
6	April 14
7	April 31

Readings From

Java Software Solutions by Lewis & Loftus – 7/E

as Supplement for Topics Covered in Lectures

- Java Programming Language: Pages 26 – 36.
- Program Development: Pages 36 – 44.
- Object-Oriented Programming: Pages 44 – 49.
- Strings: Pages 58 – 65.
- Variables + Assignment: Pages 65 – 70.
- Primitive Data Types: 71 – 75.
- Expressions: Pages 75 – 83.
- Data Conversion: Pages 83 – 87.
- Interactive Programs: Pages 87 – 92.
- Creating objects: Pages 114 – 118
- String Class: Pages 118 – 122
- Random & Math Classes: 126 – 132
- Wrapper Classes: 141 –144
- Class Die: Pages 162 – 167
- Encapsulation + Visibility Modifiers: Pages 169 – 172
- Methods (includes constructors): Pages 172 – 182
- Boolean expressions & if statements: Pages 210 – 229
- Loops: Pages 230 – 241
- Switch statement & conditional operator: Pages 270 – 275
- Do statement & for loops: Pages 275 – 284
- static variables & methods: Pages 305 – 309
- Interfaces: Pages 322 – 327
- Method Parameters & Overloading: Pages 338 – 345
- Testing: Pages 345 – 349
- Arrays: Pages 380 – 392
- Arrays of Objects: 392 – 401
- Command-line Arguments: Pages 402 – 404
- Variable Length Parameter Lists: Pages 404 – 408
- Two Dimensional Arrays: Pages 408 – 412
- Inheritance: Pages 443 – 453
- Sorting: Pages 504 – 513
- Searching: Pages 513 –519
- Exceptions: Pages 537 – 548
- Recursion: Pages584 – 601
- Queues & Stacks: Pages 627 – 631