NYU CS9053 - Fall 2018

Introduction to Java

Instructor

Brian Langel

email	phone	office	hours
blangel@nyu.edu	(712) 266 3255	JABS 474 (the classroom)	by appointment (I'll generally be available prior to lecture)

Course Description

- An introduction to the Java programming language. See Lectures for topics covered.
- This is not an introduction to programming, data structures, algorithms or other computer science topics. It is expected that the student have experience in at least one programming language prior to taking this course. This course will cover Java specific solutions to common algorithms, data structures, concurrency problems and other computer science related topics. Although not a strict prerequisite, it is assumed that the student have taken undergraduate level courses in data structures and algorithms.
- The Java ecosystem is large and many topics will not be covered; including, *JDBC*, *EJB*, *Swing*, *JSF*, *JNI*, and *Java EE* topics

Textbook

- Core Java, volume one, 10th ed.; Cay Horstmann. **ISBN-13 978-0134177304**

Recommended Textbooks

Although these two books are not required they will be referenced extensively throughout the course. If you plan on programming in Java I highly recommend purchasing these books.

- Java Concurrency in Practice; Brian Goetz et al. **ISBN-13 978-0321349606**
- Effective Java, 2nd ed.; Joshua Bloch **ISBN-13 978-0321356680**

Java Version

- All homework and exams will be graded according to Java 8, i.e. JLS v8

Purpose

The goal of this course is to teach you a pragmatic understanding of the Java programming language. It will avoid the esoteric, the rarely used and the vestigial aspects of the language and the Java ecosystem at large (e.g., *Java EE*)

How to Succeed

- Attend lectures
  - Many topics / examples / questions which will appear on the exams and in the homework will be covered in lectures but not necessarily within the textbook.
- Participate in lecture by asking questions
  - Ask questions! It's the quickest way to learn. All questions will be answered and respected. If you do not feel comfortable asking questions (i.e., shy, feel inarticulate, etc) I still would like you to participate. To make this easier for you I've thought a lot about this and feel I have a good solution (see Participation).
- Study the language
  - Don't simply read the textbook / attend lectures. Study the concepts and ideas to get a firm understanding of them and why they are the way they are in the Java language in particular. For example, if you *know* about Hashtables don't assume you already understand them in Java. There are peculiarities to every language and learning those for Java is part of this course.
- Program with the language
  - Studying the language is not sufficient for Java or any other programming language. To actually become proficient in a language you must use it and use it extensively.
- Read others' code
  - In addition to studying and programming with a language, it is extremely important to read others' code in that language when learning it. There are common paradigms and principals that others have developed over time by virtual of working within the confines of the Java language. You can learn these patterns by reading others' code. Github is an amazing resource for this.

Lectures

Lecture	Date	Topic	Reading (chapters)
1	9/5	Introduction / Basics	1 & 2 & 13.1
2	9/12	Procedural Java	3
3	9/19	Objects	4
4	9/26	Inheritance	5 (*not 5.3 or 5.7*)
5	10/3	Interfaces / Nested & Inner Classes	6 (*not 6.3 or 6.5*)
6	10/10	Exceptions / Debugging / Annotations & Regular Expressions	7 & [supplemental]
-	10/17	**Midterm** (see Exams)	-
7	10/24	Generics	8 (*not 8.9*)
8	10/31	Collections	9 & 5.3
9	11/7	Concurrency	14 & Goetz (*not 14.11*)
10	11/14	Concurrency	14 & Goetz (*not 14.11*)
-	11/21	Thanksgiving Break	-
11	11/28	IO/NIO	[supplemental]
12	12/5	Functional Java (Lambdas / Streams / etc)	6.3 & [supplemental]
13	12/12	Libraries (Guava / Jackson) / Testing (Junit / Mockito) / IDEs / Patterns (Builder, Dependency Injection, etc)	
-	12/19	**Final** (see Exams)	-

Participation

Ask as many questions as you have. I would encourage you to ask the questions in class by raising your hand. However, I understand that you may not feel comfortable asking in front of a large group and so would prefer to ask your question anonymously. To facilitate as many questions in-class as possible, each lecture I will be opening a channel on tlk.io to allow anyone to ask questions anonymously. I will have the channel running throughout lecture and will answer questions as they are asked. The channel will be the same for each lecture; it is nyucs9053-fall2018.

GitHub Usage

This class will use GitHub extensively. All lectures, links to homework assignments and discussion will happen on the GitHub class repository; To read more about the benefits to this for you as a student, read here and here

You must notify me of your GitHub user id. As soon as you do, I will verify that you're enrolled in the class, and then give you access to the class's repository. The repository is https://github.com/NYU-CS9053/Fall-2018. If you do not have access you will get a 404 message or be prompted to login. This means either; * you haven't given me your GitHub user id * I have not yet added you * or you are not logged-in with that user id.

Ensure you have access to this repository **ASAP (as soon as possible)**

Note, homework you complete and push to GitHub must not be copied elsewhere online. The homework are copy-righted material for this class. I will revoke access if you do this and submit you to NYU for code of conduct violation.

Homework

There will be 11 homework assignments. I will throw out the lowest score. This allows for some cushion in case you are unavailable to work on an assignment, are late, get a bad grade or for whatever reason do not finish an assignment.

Homework assignments will be posted immediately after lecture. They will be due at **5pm** the day of the next lecture (in general, consult the Grading Schedule for any changes)

The process for viewing and submitting homework assignments is: * I will make the assignment available immediately after the lecture * You will accept the link to the homework (posted in `homework/weekXX` where XX is the assignment number) * You will clone the homework repo and work locally * You must make all commits **before 5pm** of the due date (i.e., the day of the following lecture, you'll have about one week to complete the assignment) * You must email me the final commit id **before 5pm** of the date (run `git rev-parse HEAD` to get this id) * The TA and I will review the commit you emailed me (provided it was emailed prior to 5pm of the due date) and any previous commits you made. * You must then push your local commits to your online repository, otherwise I will not be able to see them and consequently grade them. This must be done within 24 hours of the due date.

Late submissions are treated as 0 Submitting an assignment late will not be tolerated in any circumstance. This includes any commit performed after 5pm of the due date or any commit performed before 5pm but after the latest commit id you emailed me prior to 5pm of the due date.

Important If any part of this process does not make sense please let me know **ASAP (as soon as possible)** via e-mail or in the first lecture.

See Grading for dates and overall grade percentage.

Exams

There will be 2 exams in total.

- **Midterm** - this is an in-class exam. It will be written (no computer usage).
- **Final** - this is an in-class exam. It will be slightly longer than the midterm and will also be written (no computer usage).

See Grading for dates and overall grade percentage.

Grading

Activity	Date Due	Overall Grade Percentage
Homework 1	9/12 @ 5 pm	5%
Homework 2	9/19 @ 5 pm	5%
Homework 3	9/26 @ 5 pm	5%
Homework 4	10/3 @ 5 pm	5%
Homework 5	10/10 @ 5 pm	5%
Midterm	10/17	20%
Homework 6	10/24 @ 5 pm	5%
Homework 7	10/31 @ 5 pm	5%
Homework 8	11/7 @ 5 pm	5%
Homework 9	11/14 @ 5 pm	5%
Homework 10	11/28 @ 5 pm	5%
Homework 11	12/5 @ 5 pm	5%
Final	12/19	30%

The lowest scoring homework assignment will not be counted.

Tips for Homework

- Your code must compile and be tested to work. If it does not compile/work it is better to leave a comment explaining as much of what you've done to try to remedy as you may receive partial credit.
- Comment your code. Be descriptive yet succinct.
- Follow as closely as possible the Java Code Conventions. Mainly:
  - Class names begin with an upper case character
  - Constants (`static final`) variables are all upper case.
  - Methods and variable names should be camel-case (i.e., begin with a lower case and then use an upper case character to distinguish second and subsequent words; e.g., `processRecords`)
- Use descriptive names for variables, methods & classes.

Cheating / Copying Code

Any form of cheating or using others' code will not be tolerated. All work must be original. If two students hand in essentially the same code then **both** students will receive 0 for that assignment and also for another assignment (their highest scoring) and may also face further disciplinary action from NYU, as they will be reported to the authorities, including the CSE department's student records, as described in the University's Student Code. Furthermore, the School of Engineering encourages academic excellence in an environment that promotes honesty, integrity and fairness. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Please see the school's policy on academic dishonesty here.

Homework Grading Policy

Every homework will be evaluated under the following policy:

- Style (1 - 5) - 10%
  - This is related to how your name classes, variables and how well you follow the Java Code Conventions
- Immutability (0 or 5) - 10%
  - This is whether you program with immutable data. If all data is immutable you get a 5 if one or more portions of your code uses mutable data you get a 0 for this portion of the grade.
- Repeating Past Mistakes (0 or 5) - 10%
  - This is whether or not you fix past mistakes going forward. E.g. if you used mutable data in the past homework and I commented about this and you continue to use mutable data you will get a 0 for this portion of the grade.
- Git Usage (0 or 5) - 10%
  - This is whether you properly use Git/GitHub. You should not have merge conflicts submitted in your homework. You should not submit `.class` files or IDE files.
  - NOTE, this is separate from submitting code on time. Only code prior to 5pm of the deadline (with an emailed commit id) will be reviewed and if you do not push your code for review within 24hr of the deadline you will get a 0 for the entire homework.
- Organization (1 - 5) - 20%
  - This is how well you organize your code. Is it readable and maintainable?
- Correctness (1 - 5) - 40%
  - This is whether your code fulfills the specifications of the homework. E.g. does it compile? Does it work? Does it pass test cases if present? Etc.

Illness / Personal Matters

If you have any illness or personal matter which affects your performance or ability to complete homework or take an exam please reach out to Deanna Rayment. She is the Coordinator of Student Advocacy, Compliance, and Student Affairs at Tandon and is best in a position to assist. She will coordinate with me if need be. She can be reached via email at deanna.rayment@nyu.edu and by phone at 646-997-3046.

Learning Needs / Moses Center Statement of Disability

If you are student with a disability who is requesting accommodations, please contact New York University's Moses Center for Students with Disabilities (CSD) at 212-998-4980 or mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found here. The Moses Center is located at 726 Broadway on the 2nd and 3rd floors.

Office Hours

If you want to meet me prior to class please email me so that I can ensure I'm available. If you do not email me in advance I may be early enough prior to lecture to have a meaningful conversation. Additionally, if meeting prior to lecture does not work for you I will make myself available via Google Hangout (video call). Just send me an email and we can coordinate a time that will work for both of us.

Supplemental Reading

In addition to the Textbook and the Recommended Textbooks I'd also suggest you read the following:

- Core Java, volume two, Advanced Features, 10th ed.; Cay Horstmann and Gary Cornell. **ISBN-13 978-0134177298**
  - We will be referencing the second volume in lectures 6 and 12
  - Many of the features (like JDBC, etc) we will not get into but this is still a good book to own and reference
- The Java Programming Language, 4th ed; Ken Arnold, James Gosling & David Holmes; **ISBN-13 978-0321349804**
  - Great introduction to the language and surprisingly approachable
- Java Puzzlers; Joshua Bloch & Neal Gafter **ISBN-13 978-0321336781**
  - Fun read and the first lecture will include a couple examples of which for illustrative purposes
  - Many of the puzzles deal with very esoteric aspects of the language but many of them are also gotchyas of which Java programmers should be aware
- The Well-Grounded Java Developer: Vital techniques of Java 7 and polyglot programming; Benjamin J Evans & Martijn Verburg **ISBN-13 978-1617290060**
  - Geared toward Java 7 in particular. I'd recommend you read this after taking this class. It will reinforce a lot of what you should learn in this class.
- Java in A Nutshell, 5th ed; David Flanagan **ISBN-13 978-0596007737**
  - Decent overview of the language. Not as good as *Core Java* but still worth reading

Online Resources

- Java 8 Documentation - http://docs.oracle.com/javase/8/docs/api/
  - Mostly will be referenced by you via *Google* searches but still good to browse through proactively to understand how **Javadoc** structures documentation in general
- Java Tutorials - http://docs.oracle.com/javase/tutorial
  - Fairly good tutorials that are practical and pointed. I would recommend doing most of these.
- The Java Language Specification - http://docs.oracle.com/javase/specs
  - For a rainy day...
- Angelika Langer's FAQ on Java Generics - http://www.angelikalanger.com/GenericsFAQ/JavaGenericsFAQ.html
  - Extremely great FAQ about generics added in Java 5. This is my go-to when I'm twisting my head around edge cases with generics.
- John Skerlig's Java for C++ Programmers
  - A great reference point for those coming to Java from C++. It's written by an NYU professor who has taught CS9053 for years.
- StackOverflow - http://stackoverflow.com
  - The best place to find and post questions to Java problems. I'm guessing you'll be spending a lot of time on this site while learning Java.