JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Syllabus
Computer Science EN.601.107
Introductory Programming in Java
Fall 2017
(3 credits, E)

Description
This course introduces the fundamental programming concepts and techniques in Java and is intended for all who plan to use computer programming in their studies and careers. Topics covered include control structures, arrays, functions, recursion, dynamic memory allocation, simple data structures, files, and structured program design. Elements of object-oriented design and programming are also introduced. Students without prior programming exposure are strongly advised to also take EN.601.108.
Course homework involves significant programming. Attendance and participation in class sessions are expected.

Prerequisites
Students should be comfortable using computers.

Instructor
Dr. Joanne Selinski, Associate Teaching Professor
joanne@cs.jhu.edu
http://www.cs.jhu.edu/~joanne/
Office: Malone 225, 410-516-4117
Office hours: see my webpage

Meetings
Mondays and Wednesdays from 1:30-2:45pm (Section 1) or 3-4:15pm (Sections 2 &3) in Shaffer 3.

Teaching Assistants
Head TA: Mariya Kazachkova
Course Assistants & Office hours to be posted on Piazza & held in Krieger 160.

Textbook There are many options and styles of reference for learning, and Java programming in particular. We've chosen an interactive text as the primary reference since it blends more traditional material with on-line exercises and demos to enhance learning.

- Required: Zyante learning system (https://my.zybooks.com): "Intro Programming in Java". This is an interactive textbook with integrated demos and exercises, and will be customized for our course. ($48, requires account set-up.) Students will be expected to cover sections on their own before class meetings, and might earn bonus points for timely exercise completion (requires signing up with your JHED login-based email). However, access to the materials ends a few weeks after the course end, and it doesn't have complete coverage of all course topics. (Supplemental materials will be provided as needed.)

- Optional: "Think Java" by Alan Downey and Chris Mayfield, 2016 - This text comes in a variety of formats as detailed on the book website: `http://greenteapress.com/wp/think-java/`, including pdf for download, an interactive web version, and print version available for purchase. It doesn't have coverage of all course topics. In addition to the free links available on the text website, you can read the print version for free through the JHU MSE Library collections (requires JHED login).
- Alternate: Dean & Dean, "Introduction to Programming with JAVA, A Problem Solving Approach", 2nd edition, McGraw-Hill, 2014 - This is a traditional print text that has been used in prior semesters at JHU. It is comprehensive, but must be bought through an on-line vendor such as Amazon, unless you can get it from other students. A copy will be placed on Reserves in the MSE Library.

Online Resources
Course materials, including lecture slides, readings, homework assignments, key dates, and other resources will be posted to the main course website: `cs.jhu.edu/~joanne/cs107`. We will also use Blackboard for assignment and exercise submission and grades. We will use Piazza for all discussions, questions, in class exercises and announcements.

Course Objectives
Upon successful completion of this course, you should be able to:

(1) Understand the basic concepts and principles of structured programming.
(2) Understand the basic concepts and principles of object oriented programming.
(3) Produce sample test cases, pseudocode or an incremental coding plan for a given programming problem statement.
(4) Be able to design, write and test a Java program to implement a working solution to a given problem specification.
(5) Understand the operation of simple data structures and algorithms.

Course Topics
We will cover basic concepts and techniques for programming including algorithms, variables, control structures (decision and looping), text files, method writing, class design, writing and usage, and arrays. We will also introduce recursion, sorting and searching algorithms and efficiency analysis. More advanced object oriented topics such as polymorphism and exceptions will be introduced as time permits. We will not cover any GUI material, however you are welcome to read and experiment on your own (just not in your assignments). A more detailed schedule will be posted on the course website.

Course Expectations & Grading
Students are expected to complete exercises and actively participate during class sessions. We will be taking an *active learning* approach to the course, where you learn by doing rather than listening to a lecture. To maximize the benefit of such hands-on work in this setting, it is crucial that students keep up with outside reading and come to class prepared to work on new material. Therefore, a portion of the course grade will be based on the participation effort students put into in-class exercises (ICE) as demonstrated through Piazza polls and submission of classwork to Blackboard as directed.

In addition, significant learning will come from completion of homework assignments (roughly 1 per week) which will be a mix of written exercises and programming assignments in Java. While these will not count towards your course grade, students may be asked to present their solutions in class. We will review solutions for selected homeworks, and individual feedback will be provided for others, providing informal assessment of individual skills. Students are encouraged to see course staff for help on these, and may collaborate with each other or use outside resources with proper citation.

Graded work in the course will be comprised of several more extensive programming projects, to be treated as take-home exams. These will assess the skills you have acquired through completion of the ungraded homeworks. Students may come to the course staff for help on these, but less help will be given than with the ungraded homeworks. These are individual projects and any collaboration will constitute an ethics violation.

Course testing will be comprised of in-class quizzes, an individual midterm and an individual final exam. All of these will be paper-based problems that involve concepts as well as tracing and writing snippets of actual program code. The complete grading breakdown is as follows:

- 5% - In-Class Exercises & Participation (85% attempted will be considered full credit)
- 30% - Projects (3, 10% each)
- 15% - Quizzes (4-5 in class, dates TBD)
- 20% - Midterm (in class, date TBD)
- 30% - Final Exam (2-5pm, Sunday 12/17 all sections)

Letter grades for the course will be subject to the instructor's evaluation of your overall class performance, and assigned based on a standard scale (no curve). Borderline grades may be bumped up for students who completed significant exercises in the text in a timely manner. All scores and grader feedback on your submissions will be available via Blackboard.

In-Class Exercises.
Most days during the course, you will be expected to work on In-Class Exercises (ICEs) to help you learn and practice course concepts. Some of these will require you to work with a partner. These exercises are scheduled during class time so that the instructor and CAs can provide hands-on assistance in a timely manner. Some of these exercises will require pencil-and-paper, some you can type in a word document, others will require you to write and execute code. Most days you'll need to use a charged laptop with the course software installed.
Though these exercises are not graded for correctness, you will earn credit towards your final course grade through your serious efforts to complete these exercises. Full credit for participation will be awarded for seriously attempting 85% of the exercises. Before leaving class, you must submit your work to Blackboard to earn credit for that day's ICE. Even if you have not completed the entire exercise, you should hand in what you have. You are encouraged to work through any incomplete portions of the problem set after class ends, though you are not required to hand it in later. ICE submission on Blackboard will only be open during class times, not later in the day.

Attendance. All students are generally expected to attend all meetings of this course, and actively participate in all course meetings. Out of respect for your fellow students, be sure to arrive on time each day, and turn your mobile phone completely off - there should be no beeps or buzzes, and absolutely no texting! We also ask that you sit in every other row so that the staff can circulate through the room easily each day to help with exercises. If you miss a classroom meeting for any reason, you are responsible for the material presented, which is usually posted on the main course website.

Homework Assignment Logistics. The programming assignments in this course will require you to design and write Java programs that compile with a standard Java (version 8) compiler. Programs will be evaluated not only on correctness, but also on style and good programming techniques. Please see the course website for specific assignment requirements.

You are strongly encouraged to use the jGRASP environment for program development in this course. Each student should have access to the Krieger 160 campus computer lab, where a standard Java compiler and the jGRASP development environment are installed. Be sure that you do not store your program files on public machines - doing so will be considered a serious violation of the Ethics Code. Instead, it is recommended that you use a flash drive or secure private cloud-based file service (such as Dropbox) so you can work on your programs from any computer, and will have a back-up if your

computer crashes. Alternatively, you are free to do your work on your own computer. Please see the course web site for software download information.

Additionally, please note carefully the following policies:

- Homework submissions will take place only via Blackboard (`blackboard.jhu.edu`).
- *No credit will be given for code that does not compile.* This means that all components of an assignment must compile together, or you will not receive any credit for it.
- *No late homework will be accepted.* If your code is not working perfectly by the deadline for the assignment, turn it in as-is, with detailed comments describing which parts are complete, and which still need work. Be sure that what you turn in compiles, to make it possible to receive feedback and partial credit. Exceptions for illness will be given only by the instructor (not by any teaching assistants or course assistants). Exceptions for poor planning will not be given.

In light of the above policies, you are strongly encouraged to adopt an *incremental coding* style, making small changes to your code one at a time, so that you always have a version of your program which compiles, meaning you will always have something to turn in, even if it is not 100% complete. You are also reminded to exercise discipline in backing up your work.
If you have trouble or need extra help, don't hesitate to attend office hours or contact a member of the course staff through Piazza. Please don't wait until you're hopelessly behind.

Optional 601.108 Lab Course Students without a programming background may wish to concurrently enroll in 601.108 Introduction to Programming Lab (Java). This one-credit course meets once weekly for three hours, and provides students the opportunity for hands-on programming practice in a supervised setting on concepts covered in the 601.107 lecture course. Lab exercises are generally planned to synchronize with the lecture course, so that often, completing one week's lab exercise will be good preparation for the 601.107 homework assignment due the following week. Those students who do not enroll in the 601.108 lab course may also benefit from attempting the lab exercises on their own, so links to the lab exercises will be available from the lab website: http://cs.jhu.edu/~joanne/cs108.

Academic Support - Garland 300
The following free academic support services are provided by the university for this course:

- PILOT Learning - Peer Led Team Learning: Students are organized into small study teams who meet weekly to collaborate on faculty-developed problems-sets. A trained student leader acts as captain and facilitates the weekly meetings.
  Contact: Mrs. Ariane Kelly - 410-516-4648; email ariane.kelly@jhu.edu
- Learning Den - Small Group Tutoring: Small groups consist of a maximum of six students from the same course headed by one tutor. Visit the website `http://academicsupport.jhu.edu` to register online.
  Contact: 410-516-8216; email tutoring@jhu.edu

Ethics
The strength of the university depends on academic and personal integrity. In this course, you must be honest and truthful, abiding by the *Computer Science Academic Integrity Policy*:

> Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.

Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.

Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else's solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.

If your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.

Falsifying program output or results is prohibited.

Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.

On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]". Your course instructors will let you know where to find copies of old exams, if they are available.

In addition, the specific ethics guidelines for this course are:

(1) In the completion of in-class exercises and ungraded homework assignments you may collaborate with other students and/or use outside resources for help in completing them, although outside sources are not recommended. Any work that is not entirely yours, including collaboration with other students or help from other sources must be acknowledged in your submission.

(2) In the completion of graded projects, you are permitted to discuss the instructions with other students for clarification. However, you may not discuss any substance of your solution with anyone except course staff, whether that be algorithms, test cases, component design, or actual code.

Report any violations you witness to the instructor.
You can find more information about university misconduct policies on the web at these sites:

- Undergraduates: `e-catalog.jhu.edu/undergrad-students/student-life-policies/`
- Graduate students: `e-catalog.jhu.edu/grad-students/graduate-specific-policies/`

Students with Disabilities

Any student with a disability who may need accommodations in this class must obtain an accommodation letter from Student Disability Services, 385 Garland, (410) 516–4720, `studentdisabilityservices@jhu.edu`. Additionally, please arrange an necessary accommodations such as extra time on quizzes or tests directly with Joanne in advance of each such event.

ABET Outcomes

- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution (b)
- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs (c)
- An ability to use current techniques, skills, and tools necessary for computing practice (i)

- An ability to apply design and development principles in the construction of software systems of varying complexity (k)