# CSS143: Programming Methodology

## Course Syllabus

## Autumn 2017

|  |  |
| --- | --- |
| Course: Computer Programming II | Year: 2017   Quarter: Autumn |
| CSS 143 A | SLN 23291 |
| Instructor: Lesley Kalmin | Instructor email: [lkalmin@uw.edu](mailto:lkalmin@uw.edu) |
| Class Time: T/Th 1:15-3:15 pm | Course Meets: UW2 005 |
| Office Hours: Truly House, up the hill | Office Hours: Monday 1-2pm,  Thursday 10:30-11:30am.  Available other times – just ask! |

**Course Catalog Description**: Transition from basic programming skills to a rigorous process of software development. Familiarization with higher level programming techniques (recursion, generic programming, stacks, queues, trees, searching, and sorting). Emphasizes connection between algorithmic thought and implementation. Must be concurrently enrolled in CSSKL 162.

**CSS Requirements & Prerequisites:**Prerequisite: minimum grade of 2.7 in CSS 161, CSS 142, or CSE 142; co-requisite: CSSSKL 143.

**Required Textbook:**Walter Savitch, Absolute Java, newest edition, Addison-Wesley, 2007-2014. Any additional required readings and topics will also be posted on the online on the website above.

**Grading:**

Quizzes 5%

Assignments and Lab work 30%

Midterm exams (2 of them) 40%

Comprehensive Final exam 25%

The UW grading system is described [in detail in the](http://www.washington.edu/students/gencat/front/Grading_Sys.html) [Student Guide](http://www.washington.edu/students/gencat/front/Grading_Sys.html). All graded work in this class is scored on points.  All the points in a category (e.g., HW, RQ and RA, etc.) are added up and that total is converted to a percentage. The category weighting is then then applied to the category percentage, which then becomes part of your final grade. Thus, small point value assignments will be worth less in terms of its category than large point value assignments.

Your final grade, as expressed in grade points, will be calculated from your percentage grade assuming 60% is equivalent to a grade of 0.7 and 97% is equivalent to a grade of 4.0. Grades higher than 97% receive a grade of 4.0 and grades lower than 60% receive a grade of 0.0.

**Policies:**

Work is to be done independently unless directed otherwise; collaboration of work is NOT acceptable. You may discuss the problem statement with each other and help debug, but any actual work to be turned in, must be done without collaboration. This class is run by honor code. By taking this class, you agree that you will not collaborate inappropriately on any work. Should you violate

the honor code. This violation will be taken seriously. Student conduct code:

<http://www.uwb.edu/academic/policies/Academic_Conduct.xhtml>

Homework will general be assigned on Thursdays and due the following Wednesday at midnight. Late assignments receive a grade of zero (unless we have spoken about the circumstances and prior arrangements have been made). Computer use during lecture is limited to class activities. No social networks, email, games, etc. You are not allowed to display any images on your screen during lecture as it is distracting to your neighbors trying to learn. And no other technology, no cell phones, etc. unless used for class (i.e. PollEverywhere). Conversations should be kept to a minimum as a courtesy to other students that are trying to learn.

There will be a short weekly quiz on Tuesdays on the material from the previous week. It will be approximately 10 minutes of multiple choice questions. I will drop the lowest quiz score at the end of the quarter.

**Homework (HW):** More integrative projects that solve a problem/do a task. (As opposed to labs which tend to be more focused on practicing one element of programming.) Please remember that learning programming cannot be crammed in. (In contrast, once you have learned how to program, you can write a program in an all-night frenzy if you want.) You must start the assignments early and see me promptly if you get stuck. Otherwise, you will likely go around in circles wasting time and energy.

Programs that compile and run: Because everyone has different computers, it is difficult to make sure we can execute your code on our machines.  Thus, it is your responsibility to make sure the following two conditions are met for all programs you turn in:

* It runs and produces expected results.
* It has no environment-specific settings (e.g., absolute paths), unless otherwise permitted in the assignment description

**Course goals:**

Students develop competencies associated with problem‐solving, design, testing, programming, and management techniques. Besides learning programming fundamentals, you will see common applications and consider these applications in society. Good software engineering and algorithm analysis techniques are used throughout. As with most technical courses, besides ability and motivation, it takes time to learn and master the subject.

Expect to spend an *average* of 10 to 15 hours a week outside of class time for this course; some of you may spend more time, some less time.

**Assignments:**

− Follow any special directions for turn‐in given on an assignment. Always turn in an electronic copy of just your source code **(only .java files)** using canvas. The turn‐in area is at

<https://canvas.uw.edu/courses/1174261>

− Your code should compile and run correctly.

− Syntax errors and run‐time errors with not much output yield a low grade. Run‐time errors or incorrect

answers will result in a significant number of points being deducted from your grade. In other words,

CHECK YOUR ANSWERS!!!!!!

Otherwise, you will be graded on documentation (clarity and completeness), style (indentation and use of blank lines/spaces), meaningful identifier names, organization of your program (modularity/design), efficiency (no useless, unnecessary, or unnecessarily complicated code), output (clarity and format), the overall readability, and following directions.

**Message/discussion board:** You can post (and answer) questions on the discussion board at [https://canvas.uw.edu/courses/1174261/discussion\_topics.](https://canvas.uw.edu/courses/1174261/discussion_topics)

**Access and Accommodations:** Your experience in this class is important to me. If you have already established accommodations with Disability Resources for Students (DRS), please communicate your approved accommodations to me at your earliest convenience so we can discuss your needs in this course.

If you have not yet established services through DRS, but have a temporary health condition or permanent disability that requires accommodations (conditions include but not limited to; mental health, attention-related, learning, vision, hearing, physical or health impacts), you are welcome to contact DRS at 425-352-5307 or [uwbdrs@uw.edu](mailto:uwbdrs@uw.edu). DRS offers resources and coordinates reasonable accommodations for students with disabilities and/or temporary health conditions. Reasonable accommodations are established through an interactive process between you, your instructor(s), and DRS. It is the policy and practice of the University of Washington to create inclusive and accessible learning environments consistent with federal and state law.

**For Our Veterans:** If you are a student who has served in our nation’s military forces, thank you for your service. I hope that you feel comfortable enough to confidentially self-identify yourself to me so I can help you make a successful transition from the military to higher education.

**The weekly schedule** (subject to modification during the quarter):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Date** | **Reading** | **Topic** | **Assignment (date assigned, due following Wednesday)** |
| 1 | 28-Sep | Ch. 1, 2, 3 | CSS142 review | Fractions V1 |
| 2 | 3-Oct | 4.1-4.2 | Class definitions, encapsulation | Quiz 1 |
|  | 5-Oct | 4.3-4.4 | Overloading, constructors | Fractions V2 |
| 3 | 10-Oct | Ch. 5.1, 14.1 | Static methods and variables, ArrayLists | Quiz 2 |
|  | 12-Oct | Ch. 6 | Arrays | Data Structures: ArrayLists, stacks, queues |
| 4 | 17-Oct | Ch. 9 | Exception handling | Quiz 3 |
|  | 19-Oct | Midterm 1 |  | Classes & Privacy Leaks |
| 5 | 24-Oct | Ch. 5.2 - 5.4 | References and packages |  |
|  | 26-Oct | Ch. 7 | Inheritance | Exceptions and Shape Inheritance |
| 6 | 30-Oct | Ch. 8 | Polymorphism |  |
|  | 2-Nov | Ch. 11 | Recursion | Recursion |
| 7 | 7-Nov | Ch. 13.1, 14.2 | Interfaces and generics |  |
|  | 9-Nov | Midterm 2 |  | Classes & Interfaces |
| 8 | 14-Nov | Ch. 10 | File I/O |  |
|  | 16-Nov | Ch. 15.1, 15.3, 15.4 | Linked lists - hash - trees | Linked Lists, stacks and queues |
| 9 | 21-Nov | Ch 15.5, 15.7 | Linked lists - hash - trees |  |
|  | 23-Nov | TBD | Sorting |  |
| 10 | 28-Nov |  | Sorting, Big-O |  |
|  | 30-Nov | Ch. 16 | Collections - maps - iterators | Summary assignment |
| 11 | 5-Dec |  | Catch up, Swing(?) |  |
|  | 7-Dec |  | Review for final |  |
|  | 14-Dec | Final |  |  |