CSC 106

Computer Science I Lab for Non-Majors
Spring 2018
Syllabus

Dr. Chen Huo
Office: MCT 150
Email: chuo@cs.ship.edu

Office Hours: Mon 1-3, Tue 3:30-4:30, Wed 10-11, Thu 10:30-11:30 (*and* by Appointment)

Required Texts:
Wellington and Wellington, A Developer's Approach to Learning Java. 2nd Edition.

Assignments:

Labs, Quizzes, and	92 – 100	A	80 – 81	B-
Homework: 50%	90 – 91	A-	77 – 79	C+
Exams: 30% (15% Each)	87 – 89	B+	70 – 79	C
Final Exam: 20%	82 – 86	B	60 – 69	D
			0 – 59	F

Course Material: The lectures will seek to reinforce what you learn from the textbook. However, there will be parts of each chapter that will not be directly covered in class that you will be expected to know and understand for the tests, quizzes, labs, programs, and homework.

Labs: You will be applying what we learn in class in a variety of labs. There are directions posted on D2L on how to properly turn in the code from a lab. In general, a lab is assigned on Friday (during our weekly meeting) and due on Wednesday of the following week.

Late Policy: All items turned in late have a 15% penalty accessed. After 24 hours, the assignment can no longer be turned in for credit. This is so because then I can go over the assignments in the following class if needed.

> *Example*: You turn in a project late that would normally be an 85. Since it is late you get a 72.25 (85 * 0.85).

Attendance: Attendance in this course is necessary to understand the material and is mandatory. If you must unavoidably miss a class, it is your responsibility to find out what was covered and what was assigned, and to get notes from a fellow student. We will regularly doing paper practice exercises in class to help reinforce the lab material.

The Learning Center: Is located on the main level of Lehman Library. Hours are Monday through Thursday from 9:00 am – 9:00 pm. If you find that you need additional help with this course, or any other course, please call x1420 to make an appointment with a tutor or learning specialist. You are of course more than welcome to ask me for help, this is just additional support provided to you by Shippensburg University.

Computer Science Department's Academic Dishonesty Policy[1]

The University's academic dishonesty polices are described in the Swataney, the student code of conduct. Specifically, that policy describes academic dishonesty as:

"...deceit or misrepresentation in attempting (successfully or unsuccessfully) to influence the grading policy or to obtain academic credit by a means that is not authorized by the course instructor or university policy. A breach of academic honesty is committed by students who give, as well as receive, unauthorized assistance in course and laboratory work and/or who purposefully evade, or assist other students in evading, the university's policy against academic dishonesty."[2]

The Computer Science Department agrees with the details of that policy, and expects that every student has read and understood that and the surrounding material. Most students understand what comprises the traditional forms of dishonesty, including: plagiarism, cheating on exams, or falsifying laboratory results. However, the types of assignments that are given in computer science courses frequently challenge the common perceptions of academic dishonesty. Assignments are given to assess student's abilities across a wide range of skills, including the ability to independently solve problems, collaborate to arrive at a solution, and use available tools to research existing solutions to problems. Academic dishonesty refers to any attempt to circumvent an instructor's ability to make that assessment.

Because of the wide range of assignments and different expectations for those assignments, it can be difficult for students to know what is acceptable behavior and what transgresses into academic dishonesty. In order to encourage students to help each other when appropriate, the following standard terminology was developed to describe the types of collaboration allowed on a given assignment. Course assignments will be categorized using this terminology, and if a student is unsure of what type of collaboration is allowed, they should consult the course instructor for confirmation.

There are four standard types of assignments:

Isolated Assignments: For these assignments the instructor will list the set of resources students are allowed to use. With the exception of those resources specified, students are not allowed to use anything (Internet, books, notes, etc.). In addition, this means students cannot discuss the assignment in any way with your peers.

Individual Assignments: The student is individually responsible for completing the work in these projects. Acceptable resources include textbook, notes, and any other resources made available by the course instructor. It is acceptable to use these resources, but it is unacceptable to simply copy material, even with appropriate attribution, from those resources.

It is acceptable for students to discuss strategy and possible solutions with their peers. It is unacceptable to exchange written or electronic materials. It is unacceptable to share sufficient information be shared to lead directly to a solution.
Case Study #1: Alice and Bob are working on a difficult question. Together they review material in the text and notes, discuss strategy, and propose possible partial solutions. At one point, they each reach an almost complete solution to the problem, but one is more efficient than the other. They continue to have a discussion about the merits of each. But, no one records those results and after the session, they each complete the solution on his/her own. They did not commit

[1] The policy is quoted here in its entirety, and ends with the section titled "Conclusion". It may also be found on the department website under the "Forms & Information" --> "Undergraduate" tab and at http://www.cs.ship.edu/forms/2012_AcademicDishonestyPolicy.pdf

[2] Shippensburg University, "Swataney," 2011-2012, pg. 120

academic dishonesty, because, even though they worked closely together, they both went back and independently derived their own solution, and they both have a deeper understanding of the material as a result.

Case Study #2: Alice and Bob are working on another difficult problem. Bob knows what he needs to do to finish the code, but he cannot get the syntax right. He shows his code to Alice. She attempts to prompt Bob's memory, but cannot lead him to the solution. Alice just wants to finish her own work, so she grabs Bob's keyboard and types in the missing code. As a result of Alice directly contributing to Bob's solution, they have both committed academic dishonesty.

Open Assignments:
Open Assignments are often large, and challenging projects that require a significant amount of research to complete. It is acceptable for students to use a wide array of texts, notes, research and professional literature, and even the web. Students must give proper attribution to their sources (e.g. citations in a text document, source code comments, *etc.*).

Students may not use reference materials that lead them directly to a significant portion of the solution to their problem. The clearest way to state this policy is that students are committing academic dishonesty when, as a result of their use of a source, they are able to submit a solution that they do not understand.

Case Study #3: Alice is working an open-ended project. She uses Google to search for information about the problem. She happens to arrive at a website that has an example of a program that is almost identical to the one she is supposed to write, but with just a few small changes. She studies the example, but does not copy the code. Later, she begins writing a solution to the problem, and is able to recall general details about the solution she saw. She was influenced by the solution, but she took the time to learn from example, and she has a deeper understanding of the solution as a result. She cites her sources. Alice did not commit academic dishonesty, and she demonstrated that she was able to use the available resources wisely.

Case Study #4: Charles is working on a difficult assignment. He searches the Internet, and he finds a method the author states will work. He doesn't understand how it works, but he copies the code, provides a citation to the original source, and turns it in. When questioned about the assignment, Charles cannot describe how the code works and has learned nothing from the assignment. As a result of circumventing the purpose of the assignment, namely to learn about multi-threaded programming, he has committed academic dishonesty. The important point in this case study is that the academic dishonesty was not plagiarism, it was that he purposefully evaded the intent of the assignment.

Case Study #5: Dave is stuck on an assignment. He goes to his professor, but he is still unable to solve the problem. He goes to an online chat site and poses his question to the group. Helpful experts give him the code for a complete solution to his problem which he turns in. Dave has committed academic dishonesty, because he has relied on another person to solve the problem for him, circumventing the purpose of the assignment.

Collaborative Assignments:
When working on a collaborative assignment, students are expected to work together to arrive at a solution. By agreeing to work with a partner, a student is explicitly agreeing to provide their fair share of the effort to solving the problem. Failure to participate while taking credit for an assignment is a form of academic dishonesty. Within a working group, members should follow the rules that apply for *Open Assignments*, and communication between groups should follow the rules that apply for *Individual Assignments.*

Case Study #7: Alice, Bob, and Charlie are working on a semester project. The team met to lay out their strategy and to discuss what each of them needed to do. They agreed to meet every Wednesday evening. Charlie frequently misses these meetings, or when he does show up, he doesn’t contribute anything meaningful. The code that he does submit doesn’t work and requires significant revision by Alice and Bob. Just before the deadline, Alice pulls a couple of last-minute late night sessions to make up for Charlie’s lack of contribution. When they submit the project, they include Charlie’s name in the submission. They do not notify the professor of the problems with Charlie. They have all committed academic dishonesty. Charlie took credit for others’ work. Alice and Bob assisted him in the dishonesty. They should have communicated the problems with the professor and followed the resulting instructions.

Conclusion

While this policy is designed to help students understand what is and is not academic dishonesty, for any given situation, students are encouraged to ask their professor for guidance if they are unsure of the appropriateness of a potential situation.

Programming Assignments: Any programming assignments are expected to include comments explaining the code. There should be a comment explaining the purpose of each class, variable, and method you create. Each class should list who it was written by, any methods that you got from someone else should have that person listed as the author (e.g. No plagiarizing of work.). Programs are expected to be well formatted (e.g. all the code should not be in main unless specified in the assignment). A template that you can load into Eclipse (the editor we will be using to write the Java programs) will be provided. All programs are to be formatted as shown below:

```
/**
 * Comment stating the purpose of the class.
 * Author: <Who wrote this here>
 */
public class Example
{
    private int[] myArray = { 1, 2, 3, 4, 5, 6 };  // What I store

    private int[] emptyArray = new int[] {}; // What I store

    /**
     * Comment stating what this method does.
     */
    public Example()
    {
        <Your code here>
    }

    /**
     * Comment stating what this method does.
     */
    public void bar(int p)
    {
        for (int i = 0; i < 10; i++)
        {
             <Your code here>
        }

        if ( p > 0)
        {
            <Your code here>
        }
        else
        {
            <Your code here>
        }
    }

    /**
     * Comment stating what this method does.
     */
    public void foo()
    {
        <Your code here>
    }
}
```