**CISC 181, Introduction to Computer Science, Fall 2016**

**Course Description:**

Principles of computer science illustrated and applied through programming in an object oriented language. Programming projects illustrate computational problems, styles and issues that arise in computer systems development and in all application areas of computation.

**Prerequisites:** Grade of C- or better in CISC108 or CISC106

**Co-requisites:** MATH221, or MATH241

**Instructor:** Mr. Bert R. Gibbons II, 101 Smith Hall, bgibbons@udel.edu

Office Hours: Monday afternoon, Smith Hall, room 201

Section 010 - Class Hours: Monday/Wednesday 5:30 – 6:45PM

Section 011 - Class Hours: Monday / Wednesday 7:00 – 8:15PM

Lab Sections 20, 21, 22, 30, 31, 32, all Tuesday

|  |  |  |
| --- | --- | --- |
| **TA Information** | **Lab Sections** | **Office Hours** |
| Lia Dawson  [ldawson@udel.edu](mailto:ldawson@udel.edu) | 20L 8:00 – 8:50AM | Thursday 5-7pm |
| Vinay Vizar  [vvazir@udel.edu](mailto:vvazir@udel.edu) | 21L 09:05 – 09:55AM | Tuesday 2:30-4:30pm |
| Hava Marneweck [hava@udel.edu](mailto:hava@udel.edu) | 22L 10:10 – 11:00 AM | Wednesday 10:30 - 12:30 |
| Mark Seda  [mseda@udel.edu](mailto:mseda@udel.edu) | 30L 11:15 – 12:05 PM | Thursday 2-4pm |
| Justin Irazabal [irazabal@udel.edu](mailto:irazabal@udel.edu) | 31L 12:20 – 1:10PM | Tuesday 11:30 - 1:30 |
| Vincent Fierro [vfierro@udel.edu](mailto:vfierro@udel.edu) | 32 1:25 – 2:15PM | Wednesday 1pm - 3pm |

**Textbook:** (optional, recommended resource) Liang, Introduction to Java Programming,

Comprehensive (10th Edition). Prentice Hall 2014.

<http://www.cs.armstrong.edu/liang/intro10e/>

ISBN: 978-0133813463

i>clicker (bookstore or online; must be this brand)

Assignment Schedule



**Class Schedule:** Two lecture sessions per week, 75 minutes per session

One lab session per week, 50 minutes per session

| **Week** | **Topics** | **Readings** |
| --- | --- | --- |
| 1 – 8/29 & 8/31 | Setup / getting started.  Variables / data types.  Scanner / importing external packages  Casting variables  Simple selection (if/then/else) | 1,2,3 |
| 2 – 9/7 | Loops (Do While, Until, For)  Character class (intro to wrappers)  String class  Operators / operations  Mutater / Variable scope  Simple Arrays  JavaDocs | 4,5,6,7 |
| 3 – 9/12 & 9/14 | Introduction to Object Oriented code  Classes  Constructors  Abstraction  Overriding  Overloading  Inheritance  Polymorphism  Interfaces  Abstract Classes / Methods  Exception Handling / Custom Error Classes | 8,9,10,11,14 |
| 4 – 9/19 & 9/21 | Annotations  Collections (generics)  Examples in OO (building on week 3)  Exception handling  Enums  Catch up from week 1-3 | 1-11,14 |
| 5– 9/26 & 9/28 | Junit  Comparators, Sorting with Comparators.  Complex objects | 20,21 |
| 6– 10/3 & 10/5 | More complex objects  JavaFX | 16, 20, 21 |
| 7– 10/10 & 10/12 | Midterm Review  10/12 is Midterm | All above chapters |
| 8– 10/17 & 10/19 | Multi-threading, parallel programming | 30 |
| 9– 10/24 & 10/26 | Socket programming / networking |  |
| 10– 10/31 & 11/2 | Database primer / JDBC | 21 |
| 11– 11/7 & 11/9 | Hibernate ORM, Oracle | 22 |
| 12– 11/14 & 11/16 |  |  |
| 13– 11/28 & 11/30 | JSP, JSF, Hibernate |  |
| 14– 12/5 & 12/7 | Web Services, course catch-up, Final review |  |

**Grading:**

|  |  |
| --- | --- |
| **Assignment** | **Weight** |
| Participation | 10% |
| Practice Sets | 20% |
| Discovery Labs | 20% |
| Quizzes | 20% |
| Midterm | 15% |
| Final Exam | 15% |

**Scale:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number | 100-95 | 95-90 | 90-87 | 87-83 | 83-80 | 80-77 | 77-73 | 73-70 | 70-67 | 67-63 | 63-60 | <60 |
| Letter | A | A- | B+ | B | B- | C+ | C | C- | D+ | D | D- | F |

**Lecture / Lab style:**

I consider CISC 181 a pivotal class. If you’ve never done OO coding before, it’s going to be a challenging class. I promise that if you make the effort, I’ll do my end and make sure you understand the material. If you leave this class without a fundamental understanding of how Java / OO works, you’re going to have a difficult career at UD in computer science.

I don’t waste class time with quizzes during class hours. I don’t see the utility with wasting an entire class watching sweat pour down your face while I monitor a stopwatch. Frankly speaking, the world doesn’t work like that. All the assignments are take-home, open note, open Google and except for the lab / project assignments, NOT group work. I’ll know if I’m getting the same code from two different students. Don’t take my leniency as a sign of weakness ☺

**There is one midterm, which is scheduled and proctored. Midterm is scheduled for 10/12 at 5pm. Mark it on your calendar.**

For the first four weeks, attendance is absolutely mandatory. If you miss one of the first few classes, you are guaranteed to play catch-up the rest of the term. Get some coffee, perk up and strap in.

The TAs will run the labs and grade the assignments. If there is a question on a particular grade, take it to the TA first, then to me.

**Disabilities:**

If you have a disability that requires special accommodation, please contact me by email during the first week of class.

**Participation:**

Your class participation grade is based (surprise!) on your participation in lecture. If you show up to every lecture and sit quietly and attentively, you can expect to get ONE out of five possible points. To get five points, politely ask and answer at least one question in every class. If you are unable to do this because of extreme shyness, see me during office hours in the first two weeks of the semester.

Your clicker grade is based on the number of times you respond out of the number of opportunities. It is not based on the quality of your answers, just the number, so please click. An exception to this will be students who persistently choose invalid responses. We will be using the i>clicker system for in class participation and as an immediate feedback mechanism. You should aim for an 80%+ participation rate to receive full credit.

**Practice Sets (PS):** All practice sets are assigned outside lab and are due the following Sunday evening (usually 6 days total to work on the assignment) at midnight. Lab attendance is required. **Practice sets need to be completed INDIVIDUALLY.** However, **you are free to consult other students to help complete the practice sets. Copying is not consulting -- keep in mind that each practice set is designed to cover basic material on which you will be quizzed and tested.** All practice set submissions must be submitted online using Sakai. No group submissions are allowed. Late items may be submitted to Sakai after the due date at a penalty of 15% until 9AM on Monday morning (only 9 hours after the due date). Note that ***NO LATE SUBMISSIONS WILL BE ACCEPTED AFTER THIS TIME***.

**Quizzes:** There will be a total of 8-10 Quizzes this semester. These quizzes are synchronized with the practice sets and discovery labs. If you successfully complete the assignments you should do well on the quizzes. Quizzes are take home, open note, open Google (sans cut/paste), but not group work. I don’t want to waste precious lecture time in class on quizzes; the honor code will be strictly enforced. Unexcused missed quizzes will count as zeros.

**Exams:** There will be two exams this semester: a midterm and a final. Exams are cumulative.

**Project (DL)**: The project this semester will be a smart phone application for the Google Android platform. In order to complete this project, you will complete 6 discovery labs that examine different advanced Computer Science topics. You will integrate the concepts from the discovery labs into your project and submit 3 checkpoints during the semester and then one final submission. Each checkpoint will be worth 5%, with the final submission worth 10%, for a total of 25% course grade for the project. Teams should be a student pair. Feedback collected from each group about individual participation will impact final project grade. Permission from the professor is required to work in a team of more than two students.

**Final Grade Rule:** *Your final grade cannot be more than 15 percentage points higher than your exam average.* This is to ensure mastery of fundamental skills. For example, if your exam average is a 75, you cannot receive higher than a 90 in the course. If your exam average is a 44, you cannot receive higher than a 59 in the course and will fail regardless of other completed work.

**Collaboration vs Cheating:**

Collaboration with anyone is ENCOURAGED for any in-class work. Collaboration is also encouraged for the practice sets. Collaboration of any kind is PROHIBITED during Exams.

A general rule for determining proper versus improper collaboration is the line of sight rule: when working on your program, can you see another student's work? Helping another student debug their problem set program is OKAY, but allowing another student to copy or transcribe your work is not okay.

Copying any other person's work (off the Internet, for example) without proper acknowledgment is plagiarism, a serious offense, and the one most common to computer science courses.  Anyone that aids another student with work that is expected to be done without collaboration is as guilty as the person who seeks help. Both will be prosecuted. It is strongly recommended that you familiarize yourself with the University's Policy of Academic Dishonesty.