

CSC 216 Course Syllabus

CSC 216 – Programming Concepts - Java

Section 001

FALL 2014

3 Credit Hours

Course Description

The second course in computing, intended for majors. Emphasis is placed on interpretation of inductive definitions (functions and data types); testing strategies; specification and implementation of finite-state machine; encapsulation; polymorphism; inheritance; class invariants; and resource management.

Learning Outcomes

Upon successful completion of this course, a student will be able to...

1. Describe the utility of inheritance, abstract classes, interfaces, and polymorphism in object-oriented systems, and design and implement programs which use these language features;
2. Identify the phases of a simple model of the software life cycle, and employ these phases in developing software;
3. Describe basic design modeling techniques, including UML class diagrams and simple design patterns (e.g., model/view/controller), and indicate how and when to use them;
4. Identify and compare the basic kinds of software testing, describe when to use each method, and design and implement test code;
5. Navigate and extract information from the Java API, and employ the Javadoc tool to construct internal documentation of source code;
6. Design and implement a finite state machine;
7. Identify when recursion is useful, and design and implement recursive algorithms and simple recursive data structures;
8. Construct and use a stack, queue, array-based list, and linked list.

Course Structure

Lecture and Exercises

Before and/or during each class, you will be presented with about two to five exercises that consist of conceptual questions or short programming exercises. You are encouraged, but not required, to work on these exercises with another class member. At least one member of the pair or team will need to have a laptop computer, or other electronic device, such as a tablet, that can submit answers via a Google form. You (and your partner) will be given credit for correct or mostly correct answers. The exercises provide the opportunity to explore recently covered materials individually or with peers. The exercises are submitted so the instructor can get a feel for the class' comprehension of materials in a timely manner. The exercises will be lightly evaluated by the teaching staff.

Each exercise will be scored out of 10 points. If you attempt the exercise you will receive at least a 5 (out of 10) on the exercise. The highest exercise score for a lecture will be your exercise grade for that lecture period. The **lowest five lecture scores** will be dropped, and the remaining scores will be averaged.

You must submit an answer for at least one exercise per class period to be counted as attending class for that day.

If you are absent from class, with an excused university absence, you will not be penalized for missing any exercises associated with the class.

Programming Projects

There are 3 programming projects this semester. Each project is broken into 2 parts that will be due approximately every one to two weeks. Part 1 will be a design and black box testing phase and Part 2 will be an implementation and unit testing phase. All project deliverables must be submitted electronically by the due date and follow the specified formats, submission instructions, and naming conventions. All projects (except Design Proposals and Rationales for Part 1) will be accepted up to 48 hours late through the appropriate submission system. You will lose 1 point every 2 hours the project is late, up to 24 points. No

submissions will be accepted after the 48 hour late window without a university excused absence. No late submissions will be accepted through email.

Each project write up will specify the specific submission instructions for the project.

For programming portions of the projects, use of the Eclipse Integrated Development Environment (IDE) is required.

For Part 2 of each project, all code for the project will be submitted to NC State's GitHub to an instructor provided repository. We will be using a continuous integration program, Jenkins, to automatically compile and test your program (both with your tests and the teaching staff tests) and provide style feedback. Your grade for Part 2 of that project will be calculated from the last GitHub submission you make before the deadline (even if Jenkins runs after the deadline for that submission) plus additional points for acceptance tests, FindBugs issues, and other related rubric items. The style deductions as derived from Jenkins feedback may be modified by the teaching staff when manually inspecting your comments.

Part 1 of all projects will be completed individually. Part 2 of Projects 1 and 2 will be developed individually or in an optional team of 2 or 3 at the instructor's discretion. Part 2 of Project 3 will be developed in a team of 2 or 3. Students will be eligible for participation in a team for a project only if they submit all deliverables for Part 1 of the project.

All programs are to be your own work (for paired and team assignments, all work is to be you and your assigned partner's or assigned team mates' own work). See the "Academic Integrity" section of the syllabus for further details. For each paired/team project, a peer evaluation will be required after the project's submission.

The projects are worth 45% of the final grade. The lowest project grade will count half as much as the two higher project grades. See the grading breakdown section of the syllabus for examples of calculating each individual project grade and the overall project grade.

Tutorials

An important aspect of this course is software engineering skills. We're using several tools to facilitate learning software engineering skills, specifically Eclipse, JUnit, GitHub, Jenkins, FindBugs, PMD, CheckStyle, Subversion, and EcEmma. Several tutorials are provided to introduce these tools:

- **Intro to Eclipse:** Eclipse is an industrial-strength Integrated Development Environment (IDE) that incorporates many of the tools that we'll use this semester. The Intro to Eclipse Tutorial and associated program will count as 2% of your final grade and must be completed individually.
- **Eclipse Debugger:** The Eclipse Debugger is a powerful tool that can help you find faults in your code and that can help increase your understanding of a program's flow of control. The Eclipse Debugger Tutorial and associated program will count as 1% of your final grade and must be completed individually.
- **Testing Tutorial:** To facilitate white-box testing, we will be writing unit tests using JUnit and evaluating your program and unit tests through an automated continuous integration server, Jenkins. Additionally, we will evaluate the quality of your code using static analysis tools (CheckStyle, PMD, and FindBugs). A Testing Tutorial and associated program will be assigned to introduce you to the JUnit unit test framework, GitHub for version control, and Jenkins for automation of testing, static analysis tools, and code coverage tools (EcEmma – with the underlying Jacoco coverage tool). The Testing Tutorial will count at 2% of your final grade and must be completed individually.

Exams

There will be three exams in this course counting a total of 45% of your final grade. These exams will cover all materials (readings, lectures, projects, guest speakers, etc.) prior to the exam. All exams will be cumulative appropriate to the materials covered prior to the exam date.

Exams test each student's knowledge on course learning outcomes. Problems during the exam may build on a programming scenario. The exam may require writing a class or several methods of code, designing a system using UML, designing a finite state machine and providing the code, etc.

Time

You are expected to spend, on **average**, 6 to 12 hours per week outside of class preparing and working on assignments. In some weeks, especially those around project deadlines, you may spend more than 12 hours on course work. Please plan ahead and use your time wisely. Do NOT wait until the last minute to complete programming projects!!!

Course Policies

Computers and Electronic Devices

Students are encouraged to use computers and other electronic devices like tablets during class. The teaching staff asks that students respect their neighbors and keep their focus on course materials rather than games, FaceBook, etc. Electronic devices are required for submission of exercises.

You may not record the lecture without express written permission from the instructor.

Electronic Communication

The teaching staff looks forward to receiving emails and message board posts about any questions you have about the class, materials, exams, and assignments. Below are several rules for electronic communication.

Higher education provides you with a training ground prior to entry into the work environment for your chosen career. You will use many of the following rules of "netiquette" when you are communicating with colleagues, your supervisor, or clients once you are in the work world. Although many of the rules of etiquette for electronic communication will be similar in the work environment, we have some specific to this course.

Please observe the following etiquette when communicating with the teaching staff and your peers. The teaching staff receives many emails on a daily basis and the instructor teaches several courses. Please note that a member of the teaching staff will respond to an email or message board post within 24 hours on a business day and within 48 hours on a weekend or holiday. Most of the time, we will respond more quickly, but it is not guaranteed.

Also, before sending an email, try to find the answer to the question by using various references already available to you:

- If the question is related to class administration, check the syllabus
- If the question is related to recent information, check previous emails from the teaching staff
- If the question is project or exam related, check the message board to see if it has already been answered. Also, read your textbook.

For emails, please identify your course, section, and your name in the subject line (first and last name) along with the subject of the message. For example: "CSC216-002 Jenny Smith - Question about Project 1 Part 1".

Email should include a salutation to identify the recipients of the email. For example, begin an email to your instructor with a salutation such as "Hi Dr. Heckman," or "Dr. Heckman". For emails to the sup list, consider a salutation like "Greetings Teaching Staff,". You now have the attention of the email recipients.

The tone of the email message should be professional. Re-read your email before you press Send and make a judgment as to how you would respond if you were a recipient of the email you are planning to send.

If you have a question that is beyond the scope of an email, consider coming to office hours or scheduling an appointment with a member of the teaching staff.

If you have several questions or items, please number them for ease of reading. The response will also be easier to understand.

Please spell check and correct mechanical/grammar errors. Avoid emails written only in lowercase and lacking punctuation.

Close your email with your name.

If you have a general question about a homework, post your question to Piazza. If you have a question that is more specific, that involves snippets of code, or that involves a grade question, make a private Piazza post or email it to the sup list for your section: csc216-001-sup@wolfware.ncsu.edu.

Grade Appeals

If at any time you feel an assignment was graded improperly, **write** a request for regrade and explain why you believe the assignment was graded improperly. First discuss the grade with the TA who graded the assignment. If you are still unsatisfied with the answer, submit the assignment to the instructor for a regrade. **All regrade requests must be submitted to the instructor no later than 2 weeks after the assignment was returned to you. Assignments returned within two weeks of the final exam must have all regrade requests submitted by the exam. Please talk with the TA who graded the assignment FIRST and have the written regrade explanation (an email or private Piazza post will satisfy the written regrade request requirement).**

Minimum Grade Requirements

In order to pass the course with a letter grade, you must have a 60+ average on the exams and you must have a 60+ for your programming assignments (where the Project Grade is calculated as described in the Project category, below).

In order to pass the course with a C or better, you must have a 65+ average on the exams and you must have a 65+ average on the overall programming assignments grade and tutorial grades.

Instructors

Dr. Sarah Heckman (sesmith5) - *Instructor*
Email: sarah_heckman@ncsu.edu
Web Page: <http://www4.ncsu.edu/~sesmith5>
Phone: 919-515-2042
Office Location: Engineering Building II 2297
Office Hours: M/T/W 1:00p-2:00p

Course Meetings

Lecture

Days: TH
Time: 2:20p – 3:25p
Campus: Centennial
Location: EBI 1011
This meeting is required.

Course Materials

Textbooks

Building Java Programs - *Reges and Stepp*
Edition: 3rd edition (you may be able to work with the 2nd edition, but find a friend with the 3rd to identify the differences)
ISBN: 0136091814
Web Link: <http://www.buildingjavaprograms.com>
Cost: \$121.60
This textbook is required.

Expenses

None.

Materials

None.

Requisites and Restrictions

Prerequisites

CSC116 with a C or better

Co-requisites

None.

Restrictions

None.

General Education Program (GEP) Information

GEP Category

This course does not fulfill a General Education Program category.

GEP Co-requisites

This course does not fulfill a General Education Program co-requisite.

Transportation

This course will not require students to provide their own transportation. Non-scheduled class time for field trips or out-of-class activities is NOT required for this class.

Safety & Risk Assumptions

None.

Grading

Grade Components

Component	Weight	Details
Projects	45	<p>There are three projects. Each project consists of 2 parts. Part 1 is a design and black box test planning phase. Part 2 is an implementation and testing phase.</p> <p>Part 1 of a project is worth 20% of the project grade. Part 2 of a project is worth 80% of the project grade. Therefore, if a student receives an 87 on Part 1 of a project and a 93 on Part 2 of the project, the student's grade for the project is:</p> $(87 * .2) + (93 * .8) = 17.4 + 74.4 = 91.8.$ <p>The lowest project grade is 20% of the overall project grade. The other two project grades are each worth 40% of the overall project grade. Therefore, if a student receives a 73 on Project 1, a 56 on Project 2, and a 95 on Project 3, the student's overall project grade is:</p> $(73 * .4) + (56 * .2) + (95 * .4) = 29.2 + 11.2 + 38 = 78.4$ <p>All Part 1s will be completed individually. For Part 2, Projects 1 and 2 will be completed individually or may have optional teams at the instructor's discretion. Project 3 will be completed on a team. You will only be placed on a team for Part 2 of a project if you complete Part 1 for that project.</p>
Intro to Eclipse Tutorial	2	Eclipse is an industrial-strength Integrated Development Environment (IDE) that incorporates many of the tools that we'll use this semester. The Intro to Eclipse Tutorial must be completed individually.
Eclipse Debugger Tutorial	1	The Eclipse Debugger is a powerful tool that can help you find faults in your code and that can help increase your understanding of a program's flow of control. The Eclipse Debugger Tutorial must be completed individually.
Testing Tutorial	2	To facilitate submission and white-box testing, we will be using GitHub and Jenkins. A Testing Tutorial and associated exercise will be assigned to introduce you to the JUnit unit test framework, GitHub, Jenkins, static analysis tools (CheckStyle, PMD, and FindBugs), and code coverage tools (EclEmma and Jacoco). The Testing Tutorial must be completed individually.
Exercises	5	Some exercises will be due before class others will be assigned and due during class. Exercises assigned during class, but not completed in class, will be due at 5pm the day they were assigned.
Exam 1	15	Exam 1 will cover material from approximately the first third of the course.
Exam 2	15	Exam 2 will cover material from approximately the first two-thirds of the course.
Exam 3	15	Exam 3 will cover all materials for the course.

Letter Grades

This Course uses Standard NCSU Letter Grading:

97 ≤	A+	< 100
93 ≤	A	< 97
90 ≤	A-	< 93
87 ≤	B+	< 90
83 ≤	B	< 87
80 ≤	B-	< 83
77 ≤	C+	< 80
73 ≤	C	< 77
70 ≤	C-	< 73
67 ≤	D+	< 70
63 ≤	D	< 67
60 ≤	D-	< 63
0 ≤	F	< 60

Requirements for Credit-Only (S/U) Grading

In order to receive a grade of S, students are required to take all exams and quizzes, complete all assignments, and earn a grade of C- or better. Conversion from letter grading to credit only (S/U) grading is subject to university deadlines. Refer to the Registration and Records calendar for deadlines related to grading. For more details refer to <http://policies.ncsu.edu/regulation/reg-02-20-15>.

Requirements for Auditors (AU)

Information about and requirements for auditing a course can be found at <http://policies.ncsu.edu/regulation/reg-02-20-04>.

The grade of "AU" will be awarded to students who take all exams and earn a 60% or higher average on all of the exams. Auditors are required to meet with the instructor during the first two weeks of the course.

Policies on Incomplete Grades

If an extended deadline is not authorized by the instructor or department, an unfinished incomplete grade will automatically change to an F after either (a) the end of the next regular semester in which the student is enrolled (not including summer sessions), or (b) the end of 12 months if the student is not enrolled, whichever is shorter. Incompletes that change to F will count as an attempted course on transcripts. The burden of fulfilling an incomplete grade is the responsibility of the student. The university policy on incomplete grades is located at <http://policies.ncsu.edu/regulation/reg-02-50-3>.

Late Assignments

There is a 48 hour late window for tutorial and programming project submissions, except for Design Proposals. You will lose 1 point for every 2 hours the project is late, up to 24 points. No submissions will be accepted after the 48 hour late window without a university excused absence.

Exercises will not be accepted late. You will not receive credit for an exercise if the timestamp is later than 5pm on the day the exercise was assigned.

No late submissions will be accepted through email.

Attendance Policy

For complete attendance and excused absence policies, please see <http://policies.ncsu.edu/regulation/reg-02-20-03>

Attendance Policy

Attendance to lecture is mandatory!

Absences Policy

Excused absences are defined in the NC State Academic Policy on Attendance Regulations (<http://policies.ncsu.edu/regulation/reg-02-20-03>). **Documentation of the absence is required to excuse an absence.**

- Exam makeups will only be given with a documented excused absence.
- Project extensions will only be given with a documented excused absence. If the project solution has already been released (in the case of teaching staff designs) an alternative assignment may be given.

- Exercise waivers will only be given with a documented excused absence.

All anticipated absences must be presented to the instructor no later than one week before the absence. All emergency absences must be turned in no later than one week after the student's return date. All other absences will be unexcused.

A maximum of 4 class periods per semester may be missed due to excused absences. Any number of excused absences beyond four will only be allowed with special permission of the instructor.

Makeup Work Policy

All projects and exams must be made up within one to two weeks of the absence and the timeframe will be determined through discussion between the instructor and student. If a project has moved forward in such a way that the missed project cannot be completed, the instructor may request the student to complete an alternative assignment. No exercises will be made up.

Additional Excuses Policy

None.

Academic Integrity

Academic Integrity

Students are required to comply with the university policy on academic integrity found in the Code of Student Conduct found at <http://policies.ncsu.edu/policy/pol-11-35-01>

All work that you turn in for grading must be your own! This means that all work must be an independent and individual creation by you or in the case of paired/team assignments, all work must be an independent and individual creation by you and your assigned partner or assigned teammates. Any attempt to gain an unfair advantage in grading, whether for yourself or another, is a violation of academic integrity. You may only work on an assignment with another student(s) in the class if explicitly stated in the assignment.

Students who cheat on a homework, exercise, or exam will receive a -100 for the assignment!!!

Cheating is worse than not turning in the assignment. All cases of academic misconduct will be reported to the Office of Student Conduct. A first offense will place the student on *Academic Probation* for the remainder of their academic career. A student's status on *Academic Probation* **may** affect financial aid and be reported to groups that request the information from the Office of Student Conduct, like Park Scholars, ROTC, graduate schools, employers, etc.

The Computer Science department uses software that detects cheating violations for programming projects. Do not use other student's code, do not share your code, do not copy or use code from someone who took the class X semesters ago, do not use code from online.

The only people that you MAY receive help from are your instructor, the TA(s) for CSC216, and for paired/team assignments, you may receive help from your assigned partner or your assigned teammates. For exercises, you may work with any of your neighbors that are physically present in class. You may use any of the resources provided by the teaching staff on the course website.

You MAY also reference your textbook, the textbook website, the Java API, and other third party APIs as appropriate for an assignment (for example, you may use the JUnit API to help you with writing JUnit tests).

You MAY NOT receive help from anyone or anything else.

Examples of Cheating (this list is NOT exhaustive):

- It is cheating to give any student access to any of your work which you have completed for individual class assignments.
- It is cheating AND plagiarism to use another person's work and claim it as your own. You are expected to complete all assignments on your own, unless otherwise specified in the assignment.
- It is cheating to interfere with another student's use of computing resources or to circumvent system security.
- It is cheating to email, ftp, post on the Internet, bulletin boards, message boards, etc. your work for others to obtain. Do NOT use sites that allow you to "anonymously" post code. Those sites are searchable, and others may find your code (like the teaching staff).
- It is cheating to ask or pay another person or persons to complete an assignment for you.
- It is cheating AND plagiarism to decompile any compiled code and use the decompiled source code as your own. You may also break the law by decompiling code.
- It is cheating AND plagiarism to use code that you find online.

- It is cheating to give another student access to your account (NC State account or others that you use for university work) or to give them your account password.
- It is cheating for you and another student to work collaboratively on an assignment, unless otherwise specified by the assignment.
- to circumvent the intention of the assignment and/or the automated grading system (e.g., by hardcoding test case solutions).

Examples of NOT Cheating (this list is NOT exhaustive):

- Using the code from the class website (with citations in the comments).
- Using code from other programs YOU wrote.
- Using code from other programs that YOU and a partner wrote as part of assigned exercises.
- Help from TAs or instructor (with citations in the comments).
- Using code from the textbook or textbook website (with citations in the comments).

Example Citations

```
/* (In method or class level comments)
```

```
* I received help from Dr. Heckman on date during her office hours. We discussed X.
*/
```

```
/*
* The code for this method is based on Exercise Y that I completed with Z on date.
*/
```

Protecting Yourself

- Do not leave papers lying around your workstation.
- Do not dispose of important papers in the lab recycling bins and trash cans until after the assignment is graded.
- Do not give out your password.
- Do not leave your workstation unattended or forget to log yourself out.
- Do not leave your laptop unattended.
- Do not give other students access to any of your workspace or email them any code.
- Do not give other students access to your course materials or your personal computer.
- Do not email, ftp, or post your code on the Internet, message boards, etc.
- Keep all copies of final and intermediate work until after the assignment is graded.
- Keep all graded assignments until after you receive the final grade for the course.
- Do not discuss implementation details of the assignment with your peers.

Forum Use

The forum is available to ask questions about assignments and tests. **Do NOT post any code to the forum!** The teaching staff reserves the right to edit any student's forum post for inappropriate content.

Academic Honesty

See <http://policies.ncsu.edu/policy/pol-11-35-01> for a detailed explanation of academic honesty.

Honor Pledge

Your name on any test or assignment **or** the electronic submission of an assignment through Moodle or other class courseware system indicates "I have neither given nor received unauthorized aid on this test or assignment."

Electronically-Hosted Course Components

Students may be required to disclose personally identifiable information to other students in the course, via electronic tools like email or web-postings, where relevant to the course. Examples include online discussions of class topics, and posting of student coursework. All students are expected to respect the privacy of each other by not sharing or using such information outside the course.

Electronically-hosted Components: The following materials are electronically-hosted for use by students through a combination of Moodle, Wolfware Classic, Google Docs (through NC State), NC State's GitHub, Jenkins (hosted by NC State), Piazza: lecture notes, message boards, electronic submission of assignments, electronic submission of exercises.

Accommodations for Disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, student must register with the Disability Services Office (<http://www.ncsu.edu/dso>), 919-515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation at <http://policies.ncsu.edu/regulation/reg-02-20-01>.

Support Fellow Students in Distress

As members of the NC State Wolfpack community, we each share a personal responsibility to express concern for one another and to ensure that this classroom and the campus as a whole remains a safe environment for learning. Occasionally, you may come across a fellow classmate whose personal behavior concerns or worries you. When this is the case, I would encourage you to report this behavior to the NC State Students of Concern website: <http://studentsofconcern.ncsu.edu/>. Although you can report anonymously, it is preferred that you share your contact information so they can follow-up with you personally.

Non-Discrimination Policy

NC State University provides equality of opportunity in education and employment for all students and employees. Accordingly, NC State affirms its commitment to maintain a work environment for all employees and an academic environment for all students that is free from all forms of discrimination. Discrimination based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or NC State University policy and will not be tolerated. Harassment of any person (either in the form of quid pro quo or creation of a hostile environment) based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation also is a violation of state and federal law and/or NC State University policy and will not be tolerated. Retaliation against any person who complains about discrimination is also prohibited. NC State's policies and regulations covering discrimination, harassment, and retaliation may be accessed at <http://policies.ncsu.edu/policy/pol-04-25-05> or http://www.ncsu.edu/equal_op/. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the Office for Equal Opportunity (OEO) at 919-515-3148.

Course Schedule

NOTE: The course schedule is subject to change.

Lecture — OO Review and Composition — 8/21/2014

OO Review and Composition

Lecture — Inheritance and Polymorphism — 8/26/2014

Inheritance and Polymorphism

Lecture — Abstract Classes and Interfaces — 8/28/2014

Abstract Classes and Interfaces

Lecture — Testing — 9/2/2014

Testing

Lecture — Coverage and Static Analysis — 9/4/2014

Coverage and Static Analysis

Lecture — Design and Patterns — 9/9/2014

Design and Patterns

Lecture — Software Lifecycle and Task Planning — 9/11/2014

Software Lifecycle and Task Planning

Lecture — Exceptions and Libraries — 9/16/2014

Exceptions and Libraries

Lecture — Java Collections — 9/18/2014

Lecture — Array Lists 1 — 9/23/2014

Array Lists 1

Lecture — Array Lists 2 — 9/25/2014

Array Lists 2

Lecture — Exam 1 — 9/30/2014

Exam 1 will cover all material discussed in class, readings, or assignments prior to the exam date.

Lecture — Linked Lists 1 — 10/2/2014

Linked Lists 1

Lecture — Linked Lists 2 — 10/7/2014

Linked Lists 2

Lecture — Inspections — 10/14/2014

Inspections

Lecture — Iterator and Inner Classes — 10/16/2014

Iterator and Inner Classes

Lecture — Stacks — 10/21/2014

Stacks

Lecture — Queues — 10/23/2014

Queues

Lecture — Exam 2 — 10/28/2014

Exam 2 will cover all material discussed in class, readings, or assignments prior to the exam date.

Lecture — Finite State Machines — 10/30/2014

Finite State Machines

Lecture — Finite State Machines and the State Pattern — 11/4/2014

Finite State Machines and the State Pattern

Lecture — Recursion — 11/6/2014

Recursion

Lecture — Recursion and Lists — 11/11/2014

Recursion and Lists

Lecture — GUIs — 11/13/2014

GUIs

Lecture — GUI Events — 11/18/2014

GUI Events

Lecture — Searching — 11/20/2014

Searching

Lecture — Binary Search Trees — 11/25/2014

Binary Search Trees

Lecture — Sorting — 12/2/2014

Sorting

Intro to Eclipse Tutorial — 9/2/2014

Due by 11:45pm. Late deadline is 48 hours later.

Eclipse Debugger Tutorial — 9/9/2014

Due by 11:45pm. Late deadline is 48 hours later.

Testing Tutorial — 9/16/2014

Due by 11:45pm. Late deadline is 48 hours later.

Project 1 Part 1 — 9/23/2014

Due by 11:45pm. Late deadline is 48 hours later for the Black Box Test Plan ONLY. The Design Document may NOT be submitted late due to release of the teaching staff design the morning following the deadline.

Project 1 Part 2 — 10/7/2014

Due by 11:45pm. Late deadline is 48 hours later.

Project 2 Part 1 — 10/21/2014

Due by 11:45pm. Late deadline is 48 hours later for the Black Box Test Plan ONLY. The Design Document may NOT be submitted late due to release of the teaching staff design the morning following the deadline.

Project 2 Part 2 — 11/4/2014

Due by 11:45pm. Late deadline is 48 hours later.

Project 3 Part 1 — 11/18/2014

Due by 11:45pm. Late deadline is 48 hours later for the Black Box Test Plan ONLY. The Design Document may NOT be submitted late due to release of the teaching staff design the morning following the deadline.

Project 3 Part 2 — 12/2/2014

Due by 11:45pm. Late deadline is 48 hours later.

Final Exam — 12/9/2014

Final Exam in EBI 1011 from 1pm - 4pm. The final exam is cumulative.