Course Syllabus

Professor and TA Contact Information:

Name	Kamran Z. Khan
Phone	214-280-7124 (Cell)
Office Location:	ECSS 4.403
Office Hours:	Tue/Thurs 5:30-7:00 pm. Email me for an apt first
Email:	kkhan@utdallas.edu
TA:	TBD

COURSE INFORMATION

Course Number	CS 2336 Section 502 Mon/Wed 5:30 – 6:45pm, ECSS 2.201
Course Title	Computer Science II
Credit Hours	3
Term:	Fall 2015

Course Pre-Requisites, co-requisites, and/or other restrictions

Prerequisites : CE 1337 or CS 1337 or TE 1337 with a grade of C or better.
Prerequisite or co-requisite: CE 2305 or CS 2305 or TE 2305 with a grade of C or better.
(Same as CE 2336 and TE 2336) (3-0)

Course Description: Further applications of programming techniques, introducing the fundamental concepts of data structures and algorithms. Topics include recursion, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), and algorithmic analysis. Includes comprehensive programming projects. Programming language of choice is Java.

Student Learning Objectives/Outcomes: After successful completion of this course, students will be able to implement different data structures using the Java programming language. They will be able use different data structures to program solutions to solve real problems. It will also help them understand algorithmic analysis and complexities.

- Ability to implement recursive algorithms
- Ability to implement linked lists, stacks, and queues
- Ability to implement a binary tree
- Ability to use hash tables and graphs
- Ability to understand algorithmic analysis
- Ability to create a comprehensive programming project

Required Textbooks and Materials:

Introduction to JAVA Programming, 10th Edition, Y. Daniel Liang, Prentice Hall. ISBN 978-0133761313

Additional material may be posted on eLearning

Important Dates:

August 24 Classes start
September 7 Labor Day
September 9 Census Date
September 18 Project 1 due
September 23 Exam 1
October 5 Project 2 due
October 28 Project 3 due
October 29 Last Day to Withdraw
November 2 Exam 2
November 18 Project 4 due
December 10 Project 5 due
TBA (Finals Week) Exam 3

These descriptions and timelines are subject to change at the discretion of the Professor.

University Policies

For all other University policies, please visit http://go.utdallas.edu/syllabus-policies

Date	Topic	Reading Assignment
8/24-26	Introduction to CS 2336 Review chapters 1-7	Read Chapters 1-7
8/31-9/2	Review chapters 1-7	Read Chapter 9
9/7-9	Objects and classes	Read Chapter 11
9/14-16	Inheritance and Polymorphism	Read Chapter 13 Project 1 Due
9/21-23	Abstract classes and Interfaces	Read Chapter 12
9/28-30	Exception Handling and Text I/O Exam 1 (Chapters 1-7, 9, 11, 12, 13)	
10/5-7	GUI Event Driven Programming	Project 2 Due Read Additional Material
10/12-14	Event Driven Programming Recursion Understanding of Algorithmic Analysis	Read Chapter 18 Additional Material
10/19-21	Lists, Stacks, Queues, and Priority Queues Lists, Stacks, Queues, and Priority Queues	Read Chapter 20
10/26-28	Implementation Lists, Stacks, Queues, Priority Queues	Read Chapter 24 Project 3 Due
11/2-4	Binary Search Tree Exam 2 (Chapters 18, Additional Material, 20, 24)	Read Chapter 25
11/9-11	Binary Search Tree AVL Trees	Read Chapter 26 Read Chapter 27
11/16-18	Hashing	Project 4 Due
11/23-25	Fall Break	
11/30-12/2	Graphs and Implementation Weighted Graphs and Applications	Read Chapter 28 Read Chapter 29
12/7-9	Weighted Graphs and Applications Exam 3 (25, 26, 27, 28, 29)	
12/10		Project 5 Due

DO YOU NEED ASSISTANCE?

E-mail:

Please include your course and section either in the subject or the body of your e-mail (preferably on the first line if not in the subject). This will allow me to address your e-mail as quickly as possible.

Help Desk:

For help with issues regarding your computer, UTD maintains a walk-in help desk. Visit their Web site for details: http://www.utdallas.edu/ir/helpdesk/

WHAT DO I NEED DURING LECTURE?

- **Notebook –** You are going to take a lot of notes and have quizzes. Bring paper to write on.
- **Writing Tool** – pen, pencil, crayon, etc. It's hard to take notes without one.
- **Textbook**: Introduction to JAVA Programming, 10th Edition, Y. Daniel Liang, Prentice Hall. ISBN 978-0133761313

WHAT DO WE EXPECT OF EACH STUDENT

- **Ask for help.** Email me or stop by during office hours. I want you to succeed. I would rather point you in the right direction so that you can complete an assignment instead of you remaining quiet and getting a failing grade.
- **Take responsibility for your education.** I am not the type of teacher that lectures with slides and expects students to memorize. I will teach by creating program examples in class. I will give challenging assignments to push students toward learning the intricacies of Java. Part of being a professional is learning how to teach yourself. I am going to guide each student through the topics of the semester, but a significant portion of what students take with them to the next class will be things that they learned on their own.
- **Attend every class.** Not only might you miss essential words of wisdom, you might miss a quiz as well.
- **Make mistakes!** This is how you learn how to do something. Don't be discouraged when something goes wrong; it will enlighten you on what not to do in a similar situation.

- **Read the chapter before the corresponding lecture** (see class schedule below)**.** I use class time to write programs that help illustrate the topics mentioned in each chapter. If a student doesn't have minimal knowledge of the concepts that will be covered for that chapter (which are gained by reading the chapter), it will be harder to get a deeper connection to what we are accomplishing in class.

- **Bring your textbook to class.** I know it is heavy and you would rather leave it at home to collect dust, but your book wants to be a part of your academic experience.

- **Arrive to class on time and remain in class until dismissed.** Arriving late and leaving early cause disruptions to the other students in the class and to me. Should you need to leave early for a valid reason, please notify me in advance and sit near the door to limit the disruption. Repeat offenders will be penalized by replacing a previous quiz grade with a zero.

- **No computers in class.** I know, this sounds like crazy talk to say no computers in a computer science classroom. I have seen grades improve by about 10% in classes where I do not allow computers. Without a computer in front of them, students are more engaged during class time. Many students like coding along with me in class, however, this becomes a detriment to the student. While students are copying the code I am writing in class, they are not concentrating on the logic or details behind the code. It is the logic and ideas behind the statements that are more important. All code that is written in class will be posted in eLearning after it is completed.

- **Don't sleep in class.** Let's be honest; fatigue from late night gaming and/or study sessions and it is super easy to doze off. Fight it off.

- **Silence all cell phones.** As much as we all might like to sing along to a One Direction ringtone, it causes a distraction in class and distractions eat away time that you paid for to get an education.

WHAT EACH STUDENT SHOULD EXPECT

- **An open environment dedicated to learning.** I want students to feel free to voice their opinions. Oftentimes as we code in class, I will ask students what they would do in a certain situation. I want each student to feel as if he/she can speak freely and also be open for other students to discuss that idea, even if that means that some students will disagree.

- **Class commitment of 12-15 hours a week.** Some students may put in considerably more time every week, especially if students wait until the last minute to work on a project. Students should be prepared to tackle multiple course-related activities each week (e.g. reading the textbook, studying for quizzes/exams, practicing programming, etc.)

- **A quiz could be given at any time.** Quizzes will be given to measure how well you understand the information from each chapter. It is each student's responsibility to be prepared. Quizzes will be based on chapter readings, examples from lecture and/or exercises from the book and will primarily involve coding.

- **Exams focused on application.** I do not create run-of-the-mill multiple choice exams that ask students to regurgitate things from memory. The exams are completely different than anything you have had in any other class (unless you've had me for a previous class). I expect you to apply the knowledge you have learned to the situations on the test. Questions on the test are designed to make sure that you understand what you are doing rather than regurgitating an example from your notes or the textbook.

- **A simulated professional experience.** The projects in this class require you to exercise strategies found in "the real world". Your logic for a project may force you to learn new techniques that haven't been discussed in class. You will have to perform code maintenance and improve the efficiency of previously written code. These things offer a small taste of how life might be once you graduate and are given large sums of money by a company seeking your skills.

- **A deep understanding of Java.** Upon completion of this course you will have more knowledge of Java than you ever expected. You should have peace of mind moving on in your program because you will be fully prepared to tackle what the next course in the sequence will throw at you.

THE INFORMATION YOU REALLY CARE ABOUT

Grading Scale:

98-100 A+	88-89 B+	78-79 C+	68-69 D+	Below 60 F
92-97 A	82-87 B	72-77 C	62-67 D	
90-91 A-	80-81 B-	70-71 C-	60-61 D	

Grade Components: Programming Projects (8-9): 30%
Exams (3): 60%
Quizzes/homework: 10%

General Grade Information:

All grades will be available in eLearning.

I do not curve grades. Assignments are combined into categories so that a low grade for one item will not destroy your grade. There are also opportunities provided to help students who may have done poorly on an assignment or exam.

Grade Disputes:

All grade disputes must be reported within 1 week and resolved within 2 weeks of the grade in question being posted in eLearning.

I am responsible for grading your exams. If you have questions regarding your exam, please contact me. Please note that due to FERPA, I cannot discuss grades via e-mail.

Quizzes, homework and projects will be graded by a TA. Please address any grading concerns you have regarding these grades with the TA. **If you email the TA to dispute a grade, please copy me on the email so that I am aware of the situation and can make sure it is resolved.**

Late Assignments:

Homework is due at the beginning of class and **will not be accepted late**. If your assignment is not submitted at the time of collection in class, it is late and will not be accepted. Please arrive to class on time in order to submit your homework. Homework is not accepted via e-mail unless I have approved the submission.
Projects will be accepted late at the penalty of 5% per hour late (rounded up) for up to 12 hours past the due date/time.

Projects:

Projects will be programming assignments that supplement recently discussed topics and should be completed in two to three weeks.

Projects are individual endeavors and students are not to work in groups on any project. Students are permitted (and I openly encourage students) to discuss the project. Feel free to share ideas on the logic, but **DO NOT SHARE ANY CODE.** All projects will be submitted in eLearning and will be compared for originality. Any projects that are approximate or identical copies will be reported to the Judicial Affairs Committee and I will accept their decision in regards to the grade.

Programming assignments will be graded on a 100 point basis. Not only will your project be graded on proper execution, but also things like efficiency, implementation and documentation. Keep in mind that you always want to write code that is easy to understand and is also easy to maintain. Fewer lines do not necessarily mean a better program. Please use comments liberally.

Homework:

Mini-projects are short coding assignments that can be done in the course of a few hours. These assignments will generally be due the next class meeting.

Quizzes:

Quizzes may be given in class and are generally unannounced. **No make-up quizzes will be given.** Quizzes missed for an excusable reason (with valid documentation) will be exempted. The exemption of a quiz is at the sole discretion of the instructor.

Exams:

Exams will cover chapters as listed below in the tentative course schedule below. Exams will include a variety of question types including multiple choice, short answer and programming exercises. Students are expected to be able to apply knowledge from earlier chapters in conjunction with the tested chapters. Exams are not created to make you feel smart; they are designed for students to demonstrate their understanding of the concepts.

An exam should not be missed except for the most extreme circumstances (such as hospitalization or death of an immediate family member). A make-up exam may be given to students with a valid reason (and documentation) for missing the exam. Otherwise, the missed exam grade will be zero. The allowance of a make-up exam is at the sole discretion of the instructor. Make-up exams must be completed within 48 hours of the date and time of the exam.