Course Syllabus

Course Information
(course number, course title, term, any specific section title)

Course Prefix, Number, Section: **CS 2336.0U1 & CE 2336.0U1**
Course Title: **Computer Science II**
Term: **Summer 2023**
Class Time: MW 12:30pm - 2:45pm (ECSS 2.201)

Professor Contact Information

Professor	Richard Min, Ph.D. MBA, MS/CE, STM, M.Div
Email Address	rkm010300@utdallas.edu
Office Phone	972-883-4522
Office Location	ECSS 4.609
Office Hours	Wed 9:30am-11:30am (or via email, MS Teams) Online via email or MS Teams

Course Pre-requisites, Co-requisites, and/or Other Restrictions
(including required prior knowledge or skills)

CS/CE 1337 with a grade of C or better.

Prerequisite or Corequisite: CS/CE 2305 with a grade of C or better or (Data Science major and MATH 3315).

Credit cannot be received for both CS 2337 and CS/CE 2336.

Course Description

CS/CE 2336 - Computer Science II (3 semester credit hours) 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

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. After successful completion of this course, you should be able to:

1. Ability to implement recursive algorithms
2. Ability to implement linked lists, stacks, and queues
3. Ability to implement a binary tree
4. Ability to use hash tables and graphs
5. Ability to understand algorithmic analysis
6. Ability to create a comprehensive programming project
7. Ability to implement and use generics/templates

Required Textbooks and Materials

Introduction to Java Programming and Data Structures, Comprehensive, 11th edition, by Y. Daniel Liang. ISBN 9780134670942

Note. 12th or 13th edition (comprehensive edition) is also good for the textbook. You do not need to get this book but to subscribe zybooks to get online access of the textbook.

Zybooks for your subscription: to access your online textbook for your reading, exercises, labs, review quizzes, assignments & projects, etc. We will use zybooks platform to submit & auto-grade weekly assignments (lab, review quiz, etc.) and most activities. Your submissions will be tested against several test cases (similar to what you have also used for CS 1136/1336 courses).

To sign in for subscription, go to: http://learn.zybooks.com
1. Sign in or create an account (use your UTD email for sign in).
2. Enter zyBook code: **UTDALLASCE2336CS2336MinSummer2023**
3. Subscribe

Suggested Course Materials

1. Java IDE of your choice. There are several popular IDEs to develop Java programs. You are welcome to use any of them.

Download Software | Liang, Introduction to Java Programming and Data Structures, Comprehensive Version, 12/e (pearsoncmg.com)

https://media.pearsoncmg.com/ph/esm/ecs_liang_ijp_12/cw/#software

- Eclipse IDE https://www.eclipse.org/downloads/packages
- Netbeans IDE https://netbeans.apache.org
- Web based IDEs: https://replit.com/new/java, https://www.online-java.com
- IntelliJ Idea Community Edition https://www.jetbrains.com/idea
- VS Code https://code.visualstudio.com/docs/java/java-tutorial

2. Student Resources from Pearson Publisher for the textbook

Student Resources | Liang, Introduction to Java Programming and Data Structures, Comprehensive Version, 12/e (pearsoncmg.com)

https://media.pearsoncmg.com/ph/esm/ecs_liang_ijp_12/cw/

Please plan to view the video tutorials for Ch01 to Ch18 as supplementary materials.

VideoNotes | Liang, Introduction to Java Programming and Data Structures, Comprehensive Version, 12/e (pearsoncmg.com)

https://media.pearsoncmg.com/ph/esm/ecs_liang_ijp_12/cw/#videonotes

---

Assignments & Academic Calendar

(Topics, Reading Assignments, Due Dates, Exam Dates)

** **Note:** The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.

Week # MW 3-5:15pm	Topic & Java Prog 11ed Liang (12ed, 13ed) Topics & Reading	Reading Java Prog Liang	Assessment: 2 Tests & Weekly Activity via zybooks Zybooks Week#	Due Date Weekly Activity (zybooks): Friday of each Week Projects & Test1,2
01 – 5/24 W	Syllabus	ch03-ch05	Week1&2	See elearning for detail.
02 – ~~5/29 M~~ 5/31 W	Java Introduction ch03-ch08	ch06-ch08	Week3	The due date for Weekly activity items (& projects) are by end of Friday of each week. Please see elearning for detail.
03 – 6/05 M 6/07 W	Class & OOP ch09-ch10 Inheritance & Polymorphism ch11	ch09-ch11		
04 – 6/12 M 6/14 W	Exception ch12 Abstract class & Interface ch13	ch12-ch13	Week4&5	
05 – 6/19 M 6/21 W	Recursion ch18, Generics ch19	ch18-ch19	Week6&7, Week8	
06 – 6/26 M 6/28 W	List, Stack, Queue, Priority Queue ch20, ch24	ch20, ch24	Week9	
07 – 7/03 M 7/05 W	Algorithms ch22	ch22	Week14	Test1 7/05 at Test Center
08 – 7/10 M 7/12 W	Sorting ch23	ch23	Week10&11	
09 – 7/17 M 7/19 W	Binary Search Tree ch25	ch25	Week12	
10 – 7/24 M 7/26 W	Hashing ch27	ch27	Week13&14 Week14	Test2 8/02 at Test Center
11 – 7/31 M 8/02 W	Graph ch24 & ch29	ch24, ch29	Week15	
12 – 8/07 M	Last Class. Adv Topics			
8/09 W – 8/10 Th	Final Exam Week Midterm Grade due: 6/29/23 Final Grade due: 8/15/23			

Grading Policy

(including percentages for assignments, grade scale, etc.)

Letter grades will be assigned as follows:

97-100	A+	94-96	A	90-93	A-
87-89	B+	84-86	B	80-83	B-
77-79	C+	74-76	C	70-73	C-
67-69	D+	64-66	D	60-63	D-

Below 60 F		

Note 1. Each range shown above is inclusive and without any rounding-off. For example, 94-97 for grade A is for the score falling in the range between 94.000 and 96.999 inclusively. The grade of 93.999 is for A-.

Note 2. In elearning, "Running" total in your gradebook shows the current weighted grade based on your graded work only based on what you have submitted and graded. For example, if you have done only Test1, Assignment1, Weekly postings so far (but you have missed Test2 and missed Assignment2 totally), current total grade will be based on only those entries that you have submitted and done.

60% for 2 Tests (30% for each test). Each test will be taken at Test Center (for 1-hour or 2-hour examination). Time and detail of Test will be announced later in elearning. Each student should make a seat reservation prior to each test (as soon as possible) in the beginning of the semester (within the first two weeks). If you need a makeup test, please inform the instructor via email in the beginning of the semester. All exams are closed book and closed notes. Exams will focus more on concepts and less on details. Additional study-material or documentation will be provided to avoid the need for memorization as much as possible. We will likely take all the tests in the testing center as scheduled. You can expect to see a few coding/analysis questions, a few short answer questions and a few multiple-choice questions in each test. Instructor is responsible for grading all the tests.

Any make-up tests will be arranged and scheduled during the same week (usually Tuesdays prior to the actual test date) at the discretion of the instructor. There should be a valid reason for scheduling make-up tests & they need to be coordinated with the instructor, 1-2 weeks prior to the test date except for serious medical condition (with Doctor's or Hospital's certificate will be required as a valid proof. Without it, there will be 15% penalty for any makeup test after the scheduled test date). It is unlikely that curving will be used to boost the final grades. If the instructor decides to do it, only the test scores will be boosted, but the tests' contribution will be clipped at 60%. In other words, curving will NOT make up for the points lost in all other assignments. So, it is extremely important to complete them in timely manner.

40% for Weekly Activity items (via zybooks) & 2-3 Projects (Assignments). Each weekly activity may include weekly reading, weekly labs & review quizzes (via zybooks or elearning), and programming projects (and/or bonus activity if any). Note that for summer due to a tight schedule, we may have a reduced weekly activity items to fit for the 10-week schedule. Each weekly activity item is available by Monday 8am and is due by Friday 11:59pm of the week. No late submission is accepted. Please check elearning for detail.

For your submision each activity item through elearning (each item in the week's activity folder) or via zybooks. More details on Weekly Activity, Requirement, and Submission steps will be given with eLearning. For some activity item (e.g., lab), TA may schedule a demo session and you are required to schedule your demo with TA (for 5-10 minutes) and do your demo to TA in a week after the due date. If you have any conflict for the demo schedule, please check with TA to arrange your demo to be done via MS Teams, etc.

Note. If you need a quick response for an urgent issue or concern, you may send an email to instructor or TA directly via email to get a quicker response and/or immediate attention.

An instructor who believes a student has committed an act of **plagiarism** should take appropriate action, which includes the issuing of a “penalty grade” (that is, F for the course) for academic dishonesty. For any "minor" plagiarism charge, the maximum letter grade for the course would be B+ or lower.

Course & Instructor Policies

(make-up exams, extra credit, late work, special assignments, class attendance, classroom citizenship, etc.)

Instructor is responsible for grading all the tests. TA will be responsible for grading all weekly activity items (any assignment or project). So, contact the TA directly for any grading related discrepancies (and cc to the instructor). It is not possible to give a detailed feedback for each weekly activity item or test question due to large number of students enrolled in our classes. If you need more details/clarification, you are encouraged to meet the TA/instructor via online during office hours & get personal attention. Do not rely on email alone to get the full response. If you are stuck with your assignment, it is better to turn in what you have and send us email. We will revise your submission and give some guidance. Your next submission will override the previous submission - TA will always grade the latest submission for each project. You can use email to get help for weekly assignments. Include the detailed problem description & applicable error messages, zip all your source files and include it with your email too. Do not just say "my program does not work" and expect us to figure out everything - you need to help us to help you efficiently. We expect to complete grading assignments (projects), weekly activities or quizzes, and tests in a week or so. However, when the schedule gets too busy, it can be as long as 2 weeks before the grades are assigned. It is the students' responsibility to review the grade details when they become available and follow up for clarifications if needed.

Make-up exams

Any make-up tests will be scheduled during the same week (usually one or two days prior to the actual test date) at the discretion of the instructor. There should be a valid reason for scheduling make-up tests and they need to be coordinated with the instructor, 1-2 weeks prior to the test date except for serious medical condition (with Doctor's or Hospital's certificate will be required as a valid proof). Without any valid reason, there will be 15% penalty for any makeup test scheduled after the scheduled test date.

Extra Credit

Please check for any extra or bonus credit work posted in elearning.

Late Work

No late submission of any work is accepted unless stated otherwise.

Special Assignments

Not Applicable or check with Instructor.

Classroom Citizenship

Please review the UTD policy and guideline on Student behavior and conduct, academic honesty and integrity in https://www.utdallas.edu/conduct/integrity/ and UTD BAIT team in https://www.utd.edu/conduct/bait/

Also note that all the course materials are only for your individual and personal use and for this course. Do not share or redistribute any of the course materials in any form or means with other. Do not make any of the course materials available via Internet or web site (e.g., git or github, tutorial or quizlet site).

Class Participation & Attendance.

For in-class course, your attendance and class participation is required.
(For online course, it will be tracked and assessed via your weekly activity items and tests. These items will be counted as your weekly class-participation via elearning and online).

Off-campus Instruction and Course Activities

(Below is a description of any travel and/or risk-related activity associated with this course.)
Not Applicable

Comet Creed

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

Academic Support Resources

The information contained in the following link lists the University's academic support resources for all students.

Please see http://go.utdallas.edu/academic-support-resources.

UT Dallas Syllabus Policies and Procedures

The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus.

Please go to http://go.utdallas.edu/syllabus-policies for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.

Java Prog 11ed by Liang (12ed, 13ed) – Table of Contents

1. Introduction to Computers, Programs, and Java™
2. Elementary Programming
3. Selections
4. Mathematical Functions, Characters, and Strings
5. Loops
6. Methods
7. Single-Dimensional Arrays
8. Multidimensional Arrays
9. Objects and Classes
10. Object-Oriented Thinking
11. Inheritance and Polymorphism
12. Exception Handling and Text I/O
13. Abstract Classes and Interfaces
14. JavaFX Basics
15. Event-Driven Programming and Animations
16. JavaFX UI Controls and Multimedia
17. Binary I/O
18. Recursion
19. Generics
20. Lists, Stacks, Queues, and Priority Queues
21. Sets and Maps
22. Developing Efficient Algorithms
23. Sorting
24. Implementing Lists, Stacks, Queues, and Priority Queues
25. Binary Search Trees
26. AVL Trees
27. Hashing
28. Graphs and Applications
29. Weighted Graphs and Applications
30. Aggregate Operations for Collection Streams

Bonus Chapters 31–44 are available from the Companion Website at www.pearsonhighered.com/liang:

31. Advanced JavaFX and FXML
32. Multithreading and Parallel Programming
33. Networking
34. Java Database Programming
35. Advanced Database Programming
36. Internationalization
37. Servlets
38. JavaServer Pages
39. JavaServer Faces
40. RMI
41. Web Services
42. 2-4 Trees and B-Trees
43. Red-Black Trees
44. Testing Using Junit

APPENDIXES

1. Java Keywords and Reserved Words
2. The ASCII Character Set
3. Operator Precedence Chart
4. Java Modifiers
5. Special Floating-Point Values
6. Number Systems
7. Bitwise Operations
8. Regular Expressions
9. Enumerated Types
10. The Big-O, Big-Omega, and Big-Theta Notations

Introduction to Java Programming and Data Structures, Comprehensive Version, 12ed by Y. Daniel Liang

VideoNotes | Liang, Introduction to Java Programming and Data Structures, Comprehensive Version, 12/e (pearsoncmg.com)
https://media.pearsoncmg.com/ph/esm/ecs_liang_ijp_12/cw/#videonotes

Video tutorials illustrating key concepts from the text.

Chapter 1: Introduction to Computers, Programs, and Java蒂
- Your First Java Program(opens new window)
- Compile and Run a Java Program(opens new window)
- NetBeans Brief Tutorial(opens new window)
- Eclipse Brief Tutorial(opens new window)

Chapter 2: Elementary Programming
- Obtain Input(opens new window)
- Use Operators / and %(opens new window)
- Software Development Process(opens new window)
- Compute Loan Payments(opens new window)
- Compute BMI(opens new window)

Chapter 3: Selections
- Program Addition Quiz(opens new window)
- Program Subtraction Quiz(opens new window)
- Use Multi-way if-else Statements(opens new window)
- Sort Three Integers(opens new window)
- Check Point Location(opens new window)

Chapter 4: Functions, Characters, and Strings
- Introduce Math Functions(opens new window)
- Introduce Strings and Objects(opens new window)
- Convert Hex to Decimal(opens new window)
- Compute Great Circle Distance(opens new window)
- Convert Hex to Binary(opens new window)

Chapter 5: Loops
- Use while loop(opens new window)
- Guess a Number(opens new window)
- Multiple Subtraction Quiz(opens new window)
- Use do-while loop(opens new window)
- Minimize Numeric Errors(opens new window)
- Display Loan Schedule(opens new window)
- Sum a Series(opens new window)

Chapter 6: Methods
- Define/Invoke Max Method(opens new window)
- Use Void Method(opens new window)
- Modularize Code(opens new window)
- Stepwise Refinement(opens new window)
- Reverse an Integer(opens new window)
- Compute π(opens new window)

Chapter 7: Single-Dimensional Arrays
- Random Shuffling(opens new window)
- Deck of Cards(opens new window)
- Selection Sort(opens new window)
- Command-Line Arguments(opens new window)
- Coupon Collector's Problem(opens new window)
- Consecutive Four(opens new window)

Chapter 8: Multidimensional Arrays
- Find the Row with the Largest Sum(opens new window)
- Grade Multiple-Choice Test(opens new window)
- Sudoku(opens new window)
- Multiply Two Matrices(opens new window)
- Even Number of 1s(opens new window)

Chapter 9: Objects and Classes
- Define Classes and Create Objects(opens new window)
- Static vs. Instance(opens new window)
- Data Field Encapsulation(opens new window)
- The this Keyword, Part 1(opens new window)

- The this Keyword, Part 2(opens new window)
- The Fan Class(opens new window)

Chapter 10: Object-Oriented Thinking
- The Loan Class(opens new window)
- The BMI Class(opens new window)
- The StackOfInteger Class(opens new window)
- Process Large Numbers(opens new window)
- The String Class(opens new window)
- The MyPoint Class(opens new window)

Chapter 11: Inheritance and Polymorphism
- Geometric Class Hierarchy(opens new window)
- Polymorphism and Dynamic Binding Demo(opens new window)
- New Account Class(opens new window)
- ArrayList

Chapter 12: Exception Handling and Text I/O
- Exception-Handling Advantages(opens new window)
- Create Custom Exception Classes(opens new window)
- Write and Read Data(opens new window)
- HexFormatException(opens new window)

Chapter 13: Abstract Classes and Interfaces
- Abstract GeometricObject Class(opens new window)
- Calendar and GregorianCalendar Classes(opens new window)
- The Concept of Interface(opens new window)

Chapter 14: JavaFX Basics
- Getting started with JavaFX(opens new window)
- Understand Property Binding(opens new window)
- Use Image and ImageView(opens new window)
- Use Layout Panes(opens new window)
- Use Shapes(opens new window)
- Display a Tic-Tac-Toe Board(opens new window)
- Display a Bar Chart(opens new window)

Chapter 15: Event-Driven Programming and Animations
- Handler and Its Registration(opens new window)
- Anonymous Handler(opens new window)
- Move Message Using the Mouse(opens new window)
- Animate a Rising Flag(opens new window)
- Flashing Text(opens new window)
- Simple Calculator(opens new window)
- Check Mouse Point Location(opens new window)
- Display a Running Fan(opens new window)

Chapter 16: JavaFX UI Controls and Multimedia
- Use ListView(opens new window)
- Use Slider(opens new window)
- Tic-Tac-Toe, Part 1(opens new window)
- Tic-Tac-Toe, Part 2(opens new window)
- Tic-Tac-Toe, Part 3(opens new window)
- Use Media, MediaPlayer, and MediaView(opens new window)
- Use Radio Buttons and Text Fields(opens new window)
- Set Fonts(opens new window)

Chapter 17: Binary I/O
- Copy File(opens new window)
- Object I/O(opens new window)
- Split a Large File(opens new window)

Chapter 18: Recursion
- Binary Search(opens new window)
- Directory Size(opens new window)
- Search a string in a directory(opens new window)
- Recursive tree

Zybooks cs2366 Table of contents by Week (http:// http://learn.zybooks.com)

1. Projects
2. Weeks 1 & 2: Introduction to Java
3. Weeks 1 & 2: Variables / Assignments
4. Weeks 1 & 2: Streams
5. Weeks 1 & 2: Branches
6. Weeks 1 & 2: Loops
7. Weeks 1 & 2: User-Defined Methods
8. Weeks 1 & 2: Recursion
9. Weeks 1 & 2: Arrays
10. Week 3: Characters / Strings

11. Week 3: Objects and Classes
12. Weeks 4 & 5: Inheritance and Polymorphism
13. Week 6: Arraylists
14. Week 6 and 7: Linked Lists
15. Week 7: Generics
16. Week 7: Exceptions
17. Week 8: Abstract Classes and Interfaces
18. Week 9: Stacks, and Queues
19. Weeks 10 & 11: Trees
20. Week 12: Graphs
21. Weeks 13 & 14: Hash Tables
22. Week 14: Introduction to Data Structures and Algorithms
23. Week 15: Searching and Sorting Algorithms

24. Memory Management
25. Collections
26. GUI Optional
27. JavaFX Optional
28. Additional Material Optional

29. Searching and Algorithm Analysis
30. Sorting Algorithms
31. Balanced Trees Optional
32. Heaps and Treaps
33. Algorithms
34. B-trees
35. Sets
36. My First Class
37. Coding Practice Problems
38. Additional Labs: Variables / Assignments
39. Additional Labs: Branches
40. Additional Labs: Loops
41. Additional Labs: User-Defined Methods
42. Additional Labs: Objects and Classes
43. Additional Labs: Recursion
44. Additional Labs: Introduction to Java
45. Additional Labs: Arrays
46. Additional Labs: Memory Management