



AU420 Introduction to Java Programing

Instructor Information:	Yuanmao Ye Home Institution: East China University of Science and Technology Email: ymye@ecust.edu.cn Office Hours: Determined by Instructor		
Term:	June 27, 2022 - July 22, 2022	Credits:	4 units
Class Hours:	Monday through Friday, 120 minutes per teaching day		
Discussion Sessions:	2.5 hours each week, conducted by teaching assistant(s)		
Total Contact Hours:	66 contact hours (1 contact hour = 45 mins, 3000 mins in total)		
Required Texts (with ISBN):	Liang, Y. Daniel, Introduction to Java Programming and Data Structures, Comprehensive Version ©2020 Pearson/Prentice-Hall: ISBN-10: 0-13-652023-5, ISBN-13: 978-0-13-652023-8		
Prerequisite:	MA077 Linear Algebra		



Course Overview

This course offers an introduction to computer science and computer programming in Java. With emphasis on object-oriented programming practice and problem-solving skills, the course presents a balanced coverage of Java language basics, programming concepts and techniques, standard algorithms and applications.

Learning Outcomes

On successful completion of the course, the student should be able to

1. Explain what software development is, the unique features, history, and appropriate uses of Java, and identify Java learning and reference resources.
2. Understand the use of programming tools such as editors, compilers, API(Java Doc) documentation, and IDEs, to enter, compile, package, and execute small Java programs.
3. Understand the fundamentals of Java programming, design, testing, packaging, and documentation, and perform related tasks.
4. Describe and demonstrate basic Java programming concepts, including numeric data types (and their uses, operations, and pitfalls), expressions and operators, variables and constants, scope, packages, control structures, input and output, and other procedural programming elements.
5. Discuss and use object-oriented programming concepts including classes, objects, abstraction, encapsulation, generics, inheritance, polymorphism, abstract classes and methods, interfaces and exceptions for error handling.
6. Select and appropriately apply some elements of the standard Java library, including strings, collections, arrays, dates and times, math utility classes, and other utility classes.
7. Identify and discuss the basic Java memory model elements, including the object references, and garbage collection.
8. Understand the basic concepts of sorting, searching and recursion.



Grading Policy

Quizzes	15%
Participation & Professional Classroom Conduct	5%
Programming Projects	20%
Midterm Exam	25%
Final Exam	35%

Grading Scale

Number grade	Letter grade	GPA
90-100	A	4.0
85-89	A-	3.7
80-84	B+	3.3
75-79	B	3.0
70-74	B-	2.7
67-69	C+	2.3
65-66	C	2.0
62-64	C-	1.7
60-61	D	1.0
≤59	F (Failure)	0

Class Attendance and Participation: You are expected to attend each class session. You will be responsible for all material covered during any absence. Your active participation in class discussions is very much encouraged and it will definitely enhance the learning process.

Reading and Preparation: You are expected to read the assigned chapters/sections before the class session in which they will be discussed.

Programming Projects: Programming projects will be assigned during the term in order to facilitate your understanding of the material, and will be due on a designated date and time. They will be collected and graded. Projects will be graded based on their correctness and completeness, the efficiency and presentation. In general late projects will not be accepted for credit.

Quizzes: A number of quizzes will be given throughout the summer course

Exams: There will be a midterm exam and a final exam. Exams will be comprised of questions and problems based upon textbook, topics covered in the class.

Academic Dishonesty

Acts of academic dishonesty include

- unauthorized assistance on an examination;
- falsification or invention of data;
- unauthorized collaboration on an academic exercise;
- plagiarism;



- e) misappropriation of resource materials;
- f) any unauthorized access of an instructor's files or computer account; or
- g) any other serious violation of academic integrity as established by the instructor.

Notes

You may find the following link useful:

<https://docs.oracle.com/javase/specs/jls/se17/html/index.html> (Java Language Specifications)



Class Schedule

Date	Lecture
Day 1	Course introduction. Java overview. Computer, Compiler basics, JDK tools (and installation), edit-compile-run cycle. First Java program, console output.
Day 2	OOPS, programming basics (comments, use of white-space, program style, main method, common errors and messages), Declaring variables, identifier naming rules. Data types. Differences between Java primitive numeric types and object references.
Day 3	Common logic errors: over- and under- flow, floating point comparison, rounding errors, and object comparison. Expressions. Converting types (cast). Auto-boxing, wrapper classes, object assignment, and defining immutable classes. Using static methods of java.lang.Math and java.util.Random. Unicode.
Day 4	Control structures: block statements, if statement, selection operator, switch statement. Boolean expressions, operators.
Day 5	Review comparing integer, floating point, and object values (especially String). The null value. Quiz1
Day 6	Loops: while, do, and for. The break and continue statements.
Day 7	Mathematical Functions, Characters, and Strings
Day 8	Methods (a.k.a. functions): calling, returning values, passing parameters. Pass by reference versus pass by value. Method signatures and overloading. Getters and setters (accessors and mutators) methods, for JavaBean standard. Design guidelines for methods: structured programming, testing, and good comments.
Day 9	Quiz2. Arrays: declaring, using, initializing. for-each loops with arrays.
Day 10	2 dimension arrays, and ArrayLists.
Day 11	Using partially filled arrays. linear search, sorting, binary search, complexity and “big O” notation. Copying arrays with clone and System.arraycopy.
Day 12	Midterm Exam
Day 13	Object-orientation, design and implementation of classes. Abstraction and encapsulation. Object properties (a.k.a. fields, attributes, columns, instance variables) and class properties. Adding a test driver to classes.
Day 14	Scope and lifetime: local, instance, and class variables. Access modifiers. Pass by value and Pass by reference. Array of Objects.
Day 15	OOP concepts: Inheritance, super, polymorphism (overriding, early and late binding), instanceof, super.



Day 16	Quiz3. Interfaces, abstract classes and methods.
Day 17	Exception handling
Day 18	Basic IO and recursion.
Day 19	Summary
Day 20	Final Exam