

# Introduction to Programming in Java

## Course Text

There is no text for this course. All materials are included in the course fee.

## Course Description

This course introduces programming concepts to students using the language of their choice: C++, Python, or Java, while following a consistent course sequence, structure, and schedule of assessments.

The course teaches the core computer science concepts of variables, branching, loops, arrays/lists, and functions/methods. It also introduces object-oriented programming with classes and inheritance. The course covers use pointers and streams and teaches a variety of good coding practices, including iterative development, code formatting, and variable naming schemes.

## Course Objectives

After completing this course, students will be able to:

- Use standard input and output, and understand common syntax errors
- Declare and initialize variables with valid identifiers
- Develop programs that branch based on user input
- Combine loops and arrays/lists, and develop programs with multiple arrays/lists
- Write a function/method, then return from a function/method and parameterize a function
- Initialize class variables with class constructor
- Create derived and abstract classes
- Write a recursive function
- Use binary search, O notation, and algorithm analysis

## Course Prerequisites

It is suggested, though not required, that students take Pre-Calculus or its equivalent before enrolling in this course.

## Important Terms

In this course, different terms are used to designate tasks:

- **Participation Activities:** The Participation Activities in the textbook, which are interactive activities and form the core learning material. Student graded on completion of Participation Activities.
- **Challenge Activities:** The Challenge Activities in the textbook, which are small coding problems designed to bridge the gap between reading and labs. Each problem is auto-graded.
- **Labs:** A set of graded programming tasks. Students write a program to perform a defined task. Each lab targets a particular concept. All labs are graded automatically.
- **Midterm:** A graded online test. Fully-automated grading.
- **Final:** A graded and proctored online test. Fully-automated grading.

## Academic Integrity Statement

Academic integrity is the pursuit of scholarly activity in an honest, truthful and responsible manner. Violations of academic integrity include, but are not limited to, plagiarism, cheating, fabrication and academic misconduct. Failure to comply with the Academic Integrity Policy can result in a failure and/or zero on the attempted assignment/examination, a removal from the course, disqualification to enroll in future courses, and/or revocation of an academic transcript.

## Course Completion Policy

In order for a course to be considered complete, all required coursework must be attempted, submitted, and graded. Required coursework consists of graded assignments. Any Academic Integrity Policy violations may prevent a course from being considered complete.

## Course Evaluation Criteria

Your score provides a percentage score and letter grade for each course. A passing percentage is **70%** or higher.

There are a total of 1000 points in the course:

Topic	Assessment	Points Available
1	Chapter 1: Introduction to Java	29
2	Chapter 2: Variables / Assignments	29
3	Chapter 3: Branches	29
4	Chapter 4: Loops	29
5	Chapter 5: Arrays	29
6	Chapter 6: User-Defined Methods	29
7	Chapter 7: Objects and Classes	29

Topic	Assessment	Points Available
8	Chapter 8: Memory Management	29
9	Chapter 9: Input / Output	29
9	Midterm Exam	207
10	Chapter 10: Inheritance	29
11	Chapter 11: Recursion	29
12	Chapter 12: Exceptions	29
13	Chapter 13: Generics	29
14	Chapter 14: Collections	29
15	Chapter 15: GUI	29
16	Chapter 16: JavaFX	29
17	Chapter 17: Searching and Sorting Algorithms	29
18	Final Exam	300
Total		1000

## Course Topics and Objectives

Topic Number	Topic Title	Subtopics
1	Introduction to Java	<ul style="list-style-type: none"> <li>• Programming (general)</li> <li>• Programming basics</li> <li>• Comments and whitespace</li> <li>• Errors and warnings</li> <li>• Computers and programs (general)</li> <li>• Computer tour</li> <li>• Language history</li> <li>• Problem solving</li> <li>• Why programming</li> <li>• Why whitespace matters</li> <li>• Java example: Married-couple names</li> </ul>

Topic Number	Topic Title	Subtopics
2	Variables / Assignments	<ul style="list-style-type: none"> <li>• Variables and assignments (general)</li> <li>• Variables (int)</li> <li>• Identifiers</li> <li>• Arithmetic expressions (general)</li> <li>• Arithmetic expressions (int)</li> <li>• Example: Health data</li> <li>• Floating-point numbers (double)</li> <li>• Scientific notation for floating-point literals</li> <li>• Constant variables</li> <li>• Using math methods</li> <li>• Integer division and modulo</li> <li>• Type conversions</li> <li>• Binary</li> <li>• Characters</li> <li>• Strings</li> <li>• Integer overflow</li> <li>• Numeric data types</li> <li>• Random numbers</li> <li>• Reading API documentation</li> <li>• Debugging</li> <li>• Style guidelines</li> <li>• Java example: Salary calculation with variables</li> <li>• Java example: Married-couple names with variables</li> </ul>
3	Branches	<ul style="list-style-type: none"> <li>• If-else branches (general)</li> <li>• If-else</li> <li>• More if-else</li> <li>• Equality and relational operators</li> <li>• Detecting ranges (general)</li> <li>• Detecting ranges with if-else statements</li> <li>• Logical operators</li> <li>• Order of evaluation</li> <li>• Example: Toll calculation</li> <li>• Switch statements</li> <li>• Boolean data type</li> <li>• String comparisons</li> <li>• String access operations</li> <li>• Character operations</li> <li>• More string operations</li> <li>• Conditional expressions</li> <li>• Floating-point comparison</li> </ul>

Topic Number	Topic Title	Subtopics
		<ul style="list-style-type: none"> <li>• Short circuit evaluation</li> <li>• Java example: Salary calculation with branches</li> <li>• Java example: Search for name using branches</li> </ul>
4	Loops	<ul style="list-style-type: none"> <li>• Loops (general)</li> <li>• While loops</li> <li>• More while examples</li> <li>• For loops</li> <li>• More for loop examples</li> <li>• Loops and strings</li> <li>• Nested loops</li> <li>• Developing programs incrementally</li> <li>• Break and continue</li> <li>• Variable name scope</li> <li>• Enumerations</li> <li>• Java example: Salary calculation with loops</li> <li>• Java example: Domain name validation with loops</li> </ul>
5	Arrays	<ul style="list-style-type: none"> <li>• Array concept (general)</li> <li>• Arrays</li> <li>• Array iteration drill</li> <li>• Iterating through arrays</li> <li>• Multiple arrays</li> <li>• Swapping two variables (General)</li> <li>• Loop-modifying or copying/comparing arrays</li> <li>• Debugging example: Reversing an array</li> <li>• Two-dimensional arrays</li> <li>• Enhanced for loop: Arrays</li> <li>• Java example: Annual salary tax rate calculation with arrays</li> <li>• Java example: Domain name validation with arrays</li> </ul>

Topic Number	Topic Title	Subtopics
6	User-Defined Methods	<ul style="list-style-type: none"> <li>• User-defined method basics</li> <li>• Return</li> <li>• Reasons for defining methods</li> <li>• Methods with branches/loops</li> <li>• Unit testing (methods)</li> <li>• How methods work</li> <li>• Methods: Common errors</li> <li>• Array parameters</li> <li>• Scope of variable/method definitions</li> <li>• Method name overloading</li> <li>• Parameter error checking</li> <li>• Using Scanner in methods</li> <li>• Perfect size arrays</li> <li>• Oversize arrays</li> <li>• Methods with oversize arrays</li> <li>• Comparing perfect size and oversize arrays</li> <li>• Using references in methods</li> <li>• Returning arrays from methods</li> <li>• Common errors: Methods and arrays</li> <li>• Java documentation for methods</li> <li>• Java example: Salary calculation with methods</li> <li>• Java example: Domain name validation with methods</li> </ul>
7	Objects and Classes	<ul style="list-style-type: none"> <li>• Objects: Introduction</li> <li>• Using a class</li> <li>• Defining a class</li> <li>• Mutators, accessors, and private helpers</li> <li>• Initialization and constructors</li> <li>• Choosing classes to create</li> <li>• Defining main() in a programmer-defined class</li> <li>• Unit testing (classes)</li> <li>• Constructor overloading</li> <li>• Objects and references</li> <li>• The 'this' implicit parameter</li> <li>• Primitive and reference types</li> <li>• Wrapper class conversions</li> <li>• ArrayList</li> <li>• Classes and ArrayLists</li> <li>• ArrayList ADT</li> <li>• Java documentation for classes</li> </ul>

Topic Number	Topic Title	Subtopics
		<ul style="list-style-type: none"> <li>Parameters of reference types</li> <li>Static fields and methods</li> <li>Using packages</li> <li>Java example: Salary calculation with classes</li> <li>Java example: Domain name availability with classes</li> </ul>
8	Memory Management	<ul style="list-style-type: none"> <li>Introduction to memory management</li> <li>A first linked list</li> <li>Memory regions: Heap/Stack</li> <li>Basic garbage collection</li> <li>Garbage collection and variable scope</li> <li>Java example: Employee list using ArrayLists</li> </ul>
9	Input / Output	<ul style="list-style-type: none"> <li>Output and input streams</li> <li>Output formatting</li> <li>Streams using Strings</li> <li>File input</li> <li>File output</li> </ul>
10	Inheritance	<ul style="list-style-type: none"> <li>Derived classes</li> <li>Access by members of derived classes</li> <li>Overriding member methods</li> <li>The Object class</li> <li>Polymorphism</li> <li>ArrayLists of Objects</li> <li>Abstract classes: Introduction (generic)</li> <li>Abstract classes</li> <li>Is-a versus has-a relationships</li> </ul>

Topic Number	Topic Title	Subtopics
		<ul style="list-style-type: none"> <li>• UML</li> <li>• Interfaces</li> <li>• Java example: Employees and instantiating from an abstract class</li> <li>• Java example: Employees and overriding class methods</li> </ul>
11	Recursion	<ul style="list-style-type: none"> <li>• Recursion: Introduction</li> <li>• Recursive methods</li> <li>• Recursive algorithm: Search</li> <li>• Adding output statements for debugging</li> <li>• Creating a recursive method</li> <li>• Recursive math methods</li> <li>• Recursive exploration of all possibilities</li> <li>• Stack overflow</li> <li>• Java example: Recursively output permutations</li> </ul>
12	Exceptions	<ul style="list-style-type: none"> <li>• Exception basics</li> <li>• Exceptions with methods</li> <li>• Multiple handlers</li> <li>• Exception handling in file input/output</li> <li>• Java example: Generate number format exception</li> </ul>
13	Generics	<ul style="list-style-type: none"> <li>• Comparable Interface: Sorting an ArrayList</li> <li>• Generic methods</li> <li>• Class generics</li> <li>• Java example: Map values using a generic method</li> </ul>
14	Collections	<ul style="list-style-type: none"> <li>• Enhanced for loop</li> <li>• Map: HashMap</li> <li>• Set: HashSet</li> <li>• List: LinkedList</li> <li>• Queue interface</li> <li>• Deque interface</li> </ul>
15	GUI	<ul style="list-style-type: none"> <li>• Basic graphics</li> <li>• Introduction to graphical user interfaces</li> </ul>



Topic Number	Topic Title	Subtopics
		<ul style="list-style-type: none"> <li>• Positioning GUI components using a GridBagLayout</li> <li>• GUI input and ActionListeners</li> <li>• GUI input with formatted text fields</li> <li>• GUI input with JSpinners</li> <li>• Displaying multi-line text in a JTextArea</li> <li>• Using tables in GUIs</li> <li>• Using sliders in GUIs</li> <li>• GUI tables, fields, and buttons: A seat reservation example</li> <li>• Reading files with a GUI</li> </ul>
16	JavaFX	<ul style="list-style-type: none"> <li>• Introduction to graphical user interfaces with JavaFX</li> <li>• Positioning GUI components using a GridPane</li> <li>• Input and event handlers</li> <li>• Basic graphics with JavaFX</li> </ul>
17	Searching and Sorting Algorithms	<ul style="list-style-type: none"> <li>• Searching and algorithms</li> <li>• Binary search</li> <li>• O notation</li> <li>• Algorithm analysis</li> <li>• Sorting: Introduction</li> <li>• Selection sort</li> <li>• Insertion sort</li> <li>• Quicksort</li> <li>• Merge sort</li> </ul>
18	Final Exam	<ul style="list-style-type: none"> <li>• Final Exam</li> </ul>

[Back to Top](#)