

# Fast Track to Core Java Programming for OO Developers (C++, C#)

**Days:** 4



**Prerequisites:** To ensure a smooth learning experience and maximize the benefits of attending this course, you should have prior hands-on programming experience in another OO programming language such as C# or C++. This course is not for non-OO developers, or non-developers.

**Audience:** This introductory-level course is an excellent fit for developers, software engineers, and IT professionals who have hands-on prior object-oriented programming experience (as in C++, C#, etc.) and are seeking to broaden their skills into Java programming.

**Description:** This engaging hands-on course helps experienced object-oriented developers (coming from C++, C#, etc.) quickly get up and running with programming in Java, using the latest coding skills. You'll exit the course well-prepared to tackle real-world programming tasks, equipped with an in-depth knowledge of Java and a practical skill set to create robust, efficient, and secure Java applications.

Working in a skills-immersive, hands-on learning environment led by our expert instructor, you'll explore the intricacies of the Java platform and the Java Standard Edition, working seamlessly with the Java Development Kit (JDK), and using the development environment (IDE) proficiently. The course emphasizes best practices in OO development, reinforcing core object-oriented principles such as inheritance, polymorphism, interfaces, and abstract classes, allowing you to design software applications, develop back-end systems, or contribute to enterprise-level projects.

The course centers around applying these skills to real-world challenges, ensuring you can leverage them in your role or project right after class. Throughout the course you'll build applications, create subclasses, and develop custom exception handling. You'll also gain proficiency in using Java's arrays, records, packages, and utility classes. The course also delves into the advanced features of Java programming, including working with generics, understanding functional programming with lambda expressions, and using the Collection API. You'll be proficient in the JDK, be able to navigate your course IDE, and will be able to develop complex Java programs.

You'll exit this course equipped with the skills required to tackle a variety of programming challenges and create efficient, robust Java applications with ease.

**Course Objectives:** This course combines engaging instructor-led presentations and useful demonstrations with valuable hands-on labs and engaging group activities. Throughout the course you'll:

- Master the fundamentals of the Java platform and the Java Standard Edition. This foundation will empower you to make the most of these tools, and confidently navigate the Java ecosystem.
- Get hands-on with the Java Development Kit (JDK) and become comfortable with the IDE / development environment. These are key tools that will become part of your everyday programming toolbox.
- Understand and apply essential object-oriented principles such as inheritance, polymorphism, interfaces, and abstract classes. These are the building blocks of effective Java programming.
- Engage in practical, real-world programming challenges through hands-on lab sessions. You'll become skilled in using arrays, records, packages, and utility classes, enhancing your ability to build robust and efficient Java applications.
- Dive into advanced Java programming concepts, exploring the use of generics, lambda expressions, functional interfaces, and the Collection API. This will equip you to tackle complex programming tasks and take your coding skills to the next level.
- Develop a deep understanding of exception handling in Java. Learn how to define your own exceptions, handle multiple exceptions, and use try/catch blocks. This will help you to create more reliable and fault-tolerant code.

Baton Rouge | Lafayette | New Orleans

[www.lantecctc.com](http://www.lantecctc.com)

# Fast Track to Core Java

## Programming for OO Developers

### (C++, C#)

- Explore and understand the Java Modular System and its impacts on accessibility and visibility. You'll become adept at executing Java applications, defining modules, and dealing with dependencies.
- Gain proficiency in working with collections in Java, which includes learning about the different collection implementations (Set, List, and Queue), using iterators, and sorting collections. This will enable you to manage data effectively in your Java programs.
- Explore specific Java 17 features that are covered in the course including: Switch Expressions, Text blocks, Pattern matching for instance of, Introduce records as carrier of immutable data

#### OUTLINE:

##### LESSON 1: THE JAVA PLATFORM

- Introduce the Java Platform
- Explore the Java Standard Edition
- Discuss the lifecycle of a Java Program
- Explain the responsibilities of the JVM
- Executing Java programs
- Garbage Collection
- Documentation and Code Reuse

##### LESSON 2: USING THE JDK

- Explain the JDK's file structure
- Use the command line compiler to compile a Java class
- Use the command line Java interpreter to run a Java application class

##### LESSON 3: THE ECLIPSE PARADIGM

- Become more familiar with Eclipse workbench concepts
- Explore the paradigm used by Eclipse, consisting of editors, views and perspectives in detail
- Introduce some commonly used views
- Explain Perspectives

## Getting Started with Java

##### LESSON 4: WRITING A SIMPLE CLASS

- Write a Java class that does not explicitly extend another class
- Define instance variables for a Java class
- Create object instances
- Primitives vs Object References
- Implement a main method to create an instance of the defined class
- Java keywords and reserved words

##### LESSON 5: ADDING METHODS TO THE CLASS

- Write a class with accessor methods to read and write instance variables
- Write a constructor to initialize an instance with data
- Write a constructor that calls other constructors of the class to benefit from code reuse
- Use the keyword to distinguish local variables from instance variables

## Essential Java Programming

##### LESSON 6: LANGUAGE STATEMENTS

- Arithmetic operators
- Operators to increment and decrement numbers
- Comparison operators
- Logical operators
- Return type of comparison and logical operators
- Use for loops
- Switch Expressions
- Switch Expressions and yield

##### LESSON 7: USING STRINGS AND TEXT BLOCKS

- Create an instance of the String class
- Test if two strings are equal
- Perform a case-insensitive equality test
- Contrast String, StringBuffer, and StringBuilder
- Compact Strings
- Text Blocks
- Unicode support

##### LESSON 8: FIELDS AND VARIABLES

- Discuss Block Scoping Rules

# Fast Track to Core Java

## Programming for OO Developers

### (C++, C#)

- Distinguish between instance variables and method variables within a method
- Explain the difference between the terms field and variable
- List the default values for instance variables
- Final and Static fields and methods

#### LESSON 9: SPECIALIZING IN A SUBCLASS

- Constructing a class that extends another class
- Implementing equals and toString
- Writing constructors that pass initialization data to parent constructor
- Using instanceof to verify type of an object reference
- Pattern matching for instanceof
- Overriding subclass methods
- Safely casting references to a more refined type

## Java Developer's Toolbox

#### LESSON 10: USING ARRAYS

- Declaring an array reference
- Allocating an array
- Initializing the entries in an array
- Writing methods with a variable number of arguments

#### LESSON 11: RECORDS

- Data objects in Java
- Introduce records as carrier of immutable data
- Defining records
- The Canonical constructor
- Compact constructors

#### LESSON 12: JAVA PACKAGES AND VISIBILITY

- Use the package keyword to define a class within a specific package
- Discuss levels of accessibility/visibility
- Using the import keyword to declare references to classes in a specific package
- Using the standard type naming conventions

- Visibility in the Java Modular System
- Correctly executing a Java application class
- The Java Modular System
- Defining Modules

#### LESSON 13: UTILITY CLASSES

- Introduce the wrapper classes
- Explain Autoboxing and Unboxing
- Converting String representations of primitive numbers into their primitive types
- Defining Enumerations
- Using static imports
- Deprecating classes and methods

## Object Oriented Development

#### LESSON 14: INHERITANCE AND POLYMORPHISM

- Write a subclass with a method that overrides a method in the superclass
- Group objects by their common supertype
- Utilize polymorphism
- Cast a supertype reference to a valid subtype reference
- Use the final keyword on methods and classes to prevent overriding

#### LESSON 15: INTERFACES AND ABSTRACT CLASSES

- Define supertype contracts using abstract classes
- Implement concrete classes based on abstract classes
- Define supertype contracts using interfaces
- Implement concrete classes based on interfaces
- Explain advantage of interfaces over abstract classes
- Explain advantage of abstract classes over interfaces

## Exception Handling

#### LESSON 16: INTRODUCTION TO EXCEPTION HANDLING

- Introduce the Exception architecture

# Fast Track to Core Java

## Programming for OO Developers

### (C++, C#)

- Defining a try/catch blocks
- Checked vs Unchecked exceptions

#### LESSON 17: EXCEPTIONS

- Defining your own application exceptions
- Automatic closure of resources
- Suppressed exceptions
- Handling multiple exceptions in one catch
- Enhanced try-with-resources
- Helpful NullPointerException(s)

## Advanced Java Programming

#### LESSON 18: BUILDING JAVA APPLICATIONS

- Explain the steps involved in building applications
- Define the build process
- Introduce build scripts
- Explain the standard folder layout
- Resolving project dependencies
- Tutorial: Importing code Using Maven

#### LESSON 19: INTRODUCTION TO GENERICS

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls To Generics
- When Generics Should Be Used

#### LESSON 20: INTRODUCING LAMBDA EXPRESSIONS AND FUNCTIONAL INTERFACES

- Understanding the concept of functional programming
- Understanding functional interfaces
- Lambda's and type inference

#### LESSON 21: WORKING WITH LAMBDA EXPRESSIONS

- Writing lambda expressions
- Explore the difference between anonymous classes and lambda expressions
- Consumer, Predicate and Function interfaces

## Working with Collections

#### LESSON 22: COLLECTIONS

- Provide an overview of the Collection API
- Review the different collection implementations (Set, List and Queue)
- Explore how generics are used with collections
- Examine iterators for working with collections

#### LESSON 23: USING COLLECTIONS

- Collection Sorting
- Comparators
- Using the Right Collection
- Lambda expressions in Collections

## Bonus Topics / Time Permitting

#### LESSON 24: SEALED CLASSES

- Introduce sealed classes
- The sealed and permits modifier
- Sealed interfaces
- Sealed classes and pattern matching

#### LESSON 25: STREAMS

- Understanding the problem with collections in Java
- Thinking of program solutions in a declarative way
- Use the Stream API to process collections of data
- Understand the difference between intermediate and terminal stream operations
- Filtering elements from a Stream
- Finding element(s) within a Stream
- Collecting the elements from a Stream into a List

#### LESSON 26: COLLECTORS

- Using different ways to collect the items from a Stream
- Grouping elements within a stream
- Gathering statistics about numeric property of elements in a stream