

» Course Overview

This course introduces students to the world of Java SE8. Students will get an insight into the fundamentals of Java programming. Over 9 modules, students will learn everything from absolute basics like learning about Java class, variables, and how to run a Java program, to handling arrays and exceptions. The course contains guided tutorials, do-it-yourself projects, and great resources that will help students practice and learn how to program in Java. After completion of this course, you will be prepared to take either the Java SE 8 Oracle Certified Associate (OCA) certification exam or the Information Technology Specialist in Java certification exam.

» Course Outline by Module

Module 1	Introduction to Coding and Java	Module 6	Working with Methods and Encapsulation
Module 2	Java Basics	Module 7	Working with Inheritance
Module 3	Working With Java Data Types	Module 8	Handling Exceptions
Module 4	Using Operators and Decision Constructs and Creating and Using Arrays	Module 9	Working with Selected classes from the Java API
Module 5	Using Loop Constructs		

» Module Overview and Learning Objectives

| Module 1. Introduction to Coding and Java

In this module, you will get an introduction to coding and the basics of Java. Learn to use compilers and IDEs and how to install Java. You will also learn how to think in code and understand some key concepts about algorithms, problems, and flowcharts in Java.

Learning Objectives: In this module, students will:

- Learn how to think in code and find solutions to basic mathematical problems
- Understand algorithms and flowcharts
- Understand the basics of Java, including how to use compilers and interpreters, text editor, IDE, CLI, and GUI
- Write a basic Hello World program in Java and understand key Java terminology

| **Module 2. Java Basics**

In this module, we will discuss the scope of variables and the structure of a Java class. You will learn how to create and execute programs in Java, as well as import other packages into your Java program. Finally, we will discuss some key features that are unique to Java programming.

Learning Objectives: In this module, students will:

- Define the scope of variables
- Define the structure of a Java class
- Create executable Java applications with a main method; run a Java program from the command line; produce console output
- Import other Java packages to make them accessible in your code
- Compare and contrast the features and components of Java such as platform independence, object orientation, encapsulation, etc.

| **Module 3. Working With Java Data Types**

This module will introduce you to the various data types in Java. You will learn all about working with different types of variables and the differences between them. You will also learn about objects, their lifecycle, and why they are such an important core component of any Java program. Finally, you will also learn about different types of wrapper classes.

Learning Objectives: In this module, students will:

- Declare and initialize variables (including the casting of primitive data types)
- Differentiate between object reference variables and primitive variables
- Know how to read or write to object fields
- Explain an Object's Lifecycle (creation, "dereference by reassignment" and garbage collection)
- Develop code that uses wrapper classes such as Boolean, Double, and Integer

| **Module 4. Using Operators and Decision Constructs and Creating and Using Arrays**

In this module, you will learn about the different operators that are used to perform various tasks in Java. You will learn how you can compare strings with other objects and use conditional statements to moderate the flow of content. Finally, you will learn all about one-dimensional and multi-dimensional arrays in Java.

Learning Objectives: In this module, students will:

- Use Java operators; use parentheses to override operator precedence
- Test equality between Strings and other objects using `==` and `equals ()`
- Create `if` and `if/else` and ternary constructs
- Use a `switch` statement
- Declare, instantiate, initialize and use a one-dimensional array
- Declare, instantiate, initialize and use multi-dimensional arrays

| **Module 5. Using Loop Constructs**

Java programs can be very complex and long. There are many instances where one or a set of steps need to be performed multiple times in a program. Looping constructs are used in Java to help control the flow of the program. They do this by simplifying repetitive tasks. There are many different types of looping construct statements that you can create in Java. In this module, you will learn all about `for`, enhanced `for`, `while`, and `do-while` statements, which are all loop statements. You will also learn about `break` and `continue` statements, which are jump statements used to alter the flow of loop statements.

Learning Objectives: In this module, students will:

- Create and use while loops
- Create and use for loops including the enhanced for loop
- Create and use do/while loops
- Compare loop constructs
- Use break and continue

| **Module 6. Working with Methods and Encapsulation**

This module will cover different ways in which you can work with methods and encapsulation in your Java programs. You will learn about creating and using overloaded methods with arguments and return values. You will also learn how the static keyword is used across various elements of a program. You will learn how overloading also applies to constructors, and how you can use access modifiers to adjust the visibility and privacy of data. Finally, we will discuss key concepts of encapsulation and its key advantages.

Learning Objectives: In this module, students will:

- Create methods with arguments and return values; including overloaded methods
- Apply the static keyword to methods and fields
- Create and overload constructors; differentiate between default and user-defined constructors
- Apply access modifiers
- Apply encapsulation principles to a class
- Determine the effect upon object references and primitive values when they are passed into methods that change the values

| Module 7. Working with Inheritance

In this module, you will learn about the concept of inheritance and the different types of inheritance in Java. Next, you will learn about polymorphism and how to use different types of polymorphism to develop code that overrides methods. You will gain an understanding of which situations call for the use of casting, and also about using keywords like super and this to access objects and constructors. Finally, you will learn how to use abstract classes and interfaces.

Learning Objectives: In this module, students will:

- Describe inheritance and its benefits
- Develop code that makes use of polymorphism; develop code that overrides methods; differentiate between the type of a reference and the type of an object
- Determine when casting is necessary
- Use super and this to access objects and constructors
- Use abstract classes and interfaces

| Module 8. Handling Exceptions

In this module, we will analyze the different types of exceptions in Java. Different types of exceptions affect the program flow in different ways and are used to achieve different results. You will also learn how to execute a try-catch block to modify how a program flows in Java. You will learn about the importance of handling exceptions and the correct way to do so by creating and invoking methods that throw exceptions. Java also has a number of exception classes that we will cover in this module.

Learning Objectives: In this module, students will:

- Differentiate among checked exceptions, unchecked exceptions, and Errors
- Create a try-catch block and determine how exceptions alter normal program flow
- Describe the advantages of Exception handling
- Create and invoke a method that throws an exception
- Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)

| **Module 9. Working with Selected classes from the Java API**

In this module, you will learn to work with strings in Java. You will learn about manipulating data using the `StringBuilder` class. We'll also discuss how you can use different calendar functions to modify data. We'll cover how you can create arrays to manage data and finally, we will discuss lambda expressions and how they can help you create shorter, more efficient code.

Learning Objectives: In this module, students will:

- Create and manipulate Strings
- Manipulate data using the `StringBuilder` class and its methods
- Create and manipulate calendar data using classes from `java.time.LocalDateTime`, `time.LocalDate`, `java.time.LocalTime`, `java.time.format.DateTimeFormatter`, `java.time.Period`
- Declare and use an `ArrayList` of a given type
- Write a simple Lambda expression that consumes a Lambda Predicate expression