SENNOVATE
Service Innovation

# Java Programming, Java SE 8 (JDK 8)

(Duration: 64 Hours)

*Prepares you for Oracle Certified Associate (OCA) Exam - 1Z0-808*

## An introduction to Java programming

- An overview of Java
- Java timeline
- Java editions
- How Java compares to C++ and C#
- Types of Java applications
- Two types of desktop applications
- Two types of web applications
- Mobile apps
- An introduction to Java development
- The code for a console application
- How Java compiles and interprets code
- Introduction to IDEs for Java development
- An introduction to Eclipse
- How to select a workspace
- How to import a project into a workspace
- How to open a file in the code editor
- How to compile and run a project
- How to enter input for a console application
- How to work with two or more projects
- How to remove a project from a workspace

## How to start writing Java code

- Basic coding skills
- How to code a class
- How to code a main method

- How to code statements
- How to code comments
- How to print output to the console
- How to use Eclipse to work with a new project
- How to create a new project
- How to create a new class
- How to work with Java source code and files
- How to use the code completion feature
- How to detect and correct syntax errors
- How to work with numbers
- How to declare and initialize variables
- How to assign values to a variable
- How to code arithmetic expressions
- How to work with strings
- How to declare and initialize String variables
- How to join strings
- How to include special characters in strings
- The Code Tester application
- The user interface
- The CodeTesterApp class

**How to use classes and methods**

- How to work with classes, objects, and methods
- How to import classes
- How to create an object from a class
- How to call a method from an object
- How to call a method from a class
- How to view the documentation for the Java API
- How to work with the console
- How to use the Scanner class to get input
- How to convert strings to numbers

- A class that reads input from the console
- How to convert numbers to formatted strings
- A class that prints formatted numbers to the console
- How to code simple control statements
- How to compare numbers
- How to compare strings
- How to code a while loop
- How to code an if/else statement
- The Line Item application
- The user interface
- The code
- The Future Value application
- The user interface
- The code

**How to code your own classes and methods**

- An introduction to classes
- How encapsulation works
- The relationship between a class and its objects
- How to work with a class that defines an object
- The Product class
- How to code instance variables
- How to code constructors
- How to code methods
- How to create an object from a class
- How to call the methods of an object
- How to work with static fields and methods
- The ProductDB class
- How to code and call static fields and methods
- When to use static fields and methods
- The Product Viewer application

- The user interface
- The ProductApp class
- More skills for working with classes and methods
- Reference types compared to primitive types
- How to overload methods
- How to use the this keyword
- The Product class with overloading

**How to structure an object-oriented application**

- How to use the three-tier architecture
- How the three-tier architecture works
- How to work with packages
- How to use Eclipse to work with packages
- The Line Item application
- The user interface
- The class diagram
- The LineItem class
- The LineItemApp class

**How to test and debug an application**

- Basic skills for testing and debugging
- Typical test phases
- The three types of errors
- Common Java errors
- How to determine the cause of an error
- A simple way to trace code execution
- How to use Eclipse to debug an application
- How to set and remove breakpoints
- How to step through code
- How to inspect variables

- How to inspect the stack trace

**How to work with primitive types and operators**

- Basic skills for working with data
- The eight primitive data types
- How to declare and initialize variables
- How to declare and initialize constants
- How to code arithmetic expressions
- How to use the binary operators
- How to use the unary operators
- How to use the compound assignment operators
- How to work with the order of precedence
- How to work with casting
- How to use Java classes to work with numbers
- How to use the Math class
- How to use the BigDecimal class
- How to fix rounding errors
- The Invoice application
- The user interface
- The code

**How to code control statements**

- How to code Boolean expressions
- How to compare primitive data types
- How to use the logical operators
- How to code if/else and switch statements
- How to code if/else statements
- How to code switch statements
- A new if/else statement for the Invoice application

- How to code loops
- How to code while loops
- How to code do-while loops
- How to code for loops
- How to code break and continue statements
- How to code try/catch statements
- How exceptions work
- How to catch exceptions
- The Future Value application
- The user interface
- The code

**How to work with strings**

- How to work with the String class
- How to create strings
- How to join strings
- How to append data to a string
- How to compare strings
- How to work with string indexes
- How to modify strings
- How to work with the StringBuilder class
- How to create a StringBuilder object
- How to append data to a string
- How to modify strings
- The Product Lister application
- The user interface
- The StringUtil class
- The Main class

**How to work with arrays**

- Essential skills for working with arrays
- How to create an array
- How to assign values to the elements of an array
- How to use for loops with arrays
- How to use enhanced for loops with arrays
- How to work with two-dimensional arrays
- How to use the Arrays class
- How to fill an array
- How to sort an array
- How to search an array
- How to create a reference to an array
- How to copy an array
- How to compare two arrays
- The Month Selector application
- The user interface
- The Main class

**How to work with inheritance**

- An introduction to inheritance
- How inheritance works
- How the Object class works
- Basic skills for working with inheritance
- How to create a superclass
- How to create a subclass
- How polymorphism works
- The Product application
- The console
- The Product, Book, and Software classes
- The ProductDB class
- The ProductApp class
- More skills for working with inheritance

- How to cast objects
- How to compare objects
- How to work with the abstract and final keywords
- How to work with the abstract keyword
- How to work with the final keyword

**How to work with interfaces**

- An introduction to interfaces
- A simple interface
- Interfaces compared to abstract classes
- Basic skills for working with interfaces
- How to code an interface
- How to implement an interface
- How to inherit a class and implement an interface
- How to use an interface as a parameter
- How to use inheritance with interfaces
- New features for working with interfaces
- How to work with default methods
- How to work with static methods
- The Product Viewer application
- The console
- The ProductReader interface
- The ProductDB class
- The ProductApp class

**How to work with inner classes, enumerations, and documentation**

- How to work with inner classes
- An introduction to GUI programming
- How to code an inner class
- How to code an anonymous class

- How to work with enumerations
- How to declare an enumeration
- How to use an enumeration
- How to enhance an enumeration
- How to work with static imports
- How to document a class
- How to add javadoc comments to a class
- How to use HTML and javadoc tags in javadoc comments
- How to use Eclipse to generate documentation
- How to view the documentation

**How to work with collections, generics, and lambdas**

- An introduction to Java collections
- A comparison of arrays and collections
- An overview of the Java collection framework
- An introduction to generics
- How to use the ArrayList class
- How to create an array list
- How to add and get elements
- How to replace, remove, and search for elements
- How to store primitive types in an array list
- The Invoice application
- The user interface
- The Invoice class
- The InvoiceApp class
- How to work with lambda expressions
- An introduction to lambdas
- A method that doesn't use a lambda expression
- A method that uses a lambda expressions
- How to use the Predicate interface

**How to work with dates and times**

- An introduction to date/time APIs
- The date/time API prior to Java 8
- The date/time API for Java 8 and later
- How to use the new date/time API
- How to create date and time objects
- How to get date and time parts
- How to compare dates and times
- How to adjust date/time objects
- How to add or subtract a period of time
- How to format dates and times
- An Invoice class that includes an invoice date

**How to handle exceptions**

- An introduction to exceptions
- The exception hierarchy
- How exceptions are propagated
- How to work with exceptions
- How to use the try statement
- How to use the try-with-resources statement
- How to use the methods of an exception
- How to use a multi-catch block
- How to use the throws clause
- How to use the throw statement
- How to work with custom exception classes
- How to create your own exception class
- How to use exception chaining

**How to work with file I/O**

- An introduction to directories and files

- A package for working with directories and files
- Code examples that work with directories and files
- An introduction to file input and output
- How files and streams work
- A file I/O example
- How to work with I/O exceptions
- How to work with text files
- How to connect a character output stream to a file
- How to write to a text file
- How to connect a character input stream to a file
- How to read from a text file
- A class that works with a text file
- The Product Manager application
- The console
- The Main class

**How to work with threads**

- An introduction to threads
- How threads work
- Typical uses for threads
- Classes and interfaces for working with threads
- The life cycle of a thread
- Two ways to create threads
- Constructors and methods of the Thread class
- How to extend the Thread class
- How to implement the Runnable interface
- How to synchronize threads
- How to use synchronized methods
- When to use synchronized methods

**How to work with a MySQL database**

- How a relational database is organized
- How a table is organized
- How the tables in a database are related
- How the columns in a table are defined
- An introduction to MySQL
- What MySQL provides
- Ways to interact with MySQL
- How to open a database connection
- How to enter and execute a SQL statement
- A SQL script that creates a database
- How to drop, create, and select a database
- How to create a table and insert data
- How to create a user and grant privileges
- The SQL statements for data manipulation
- How to select data from a table
- How to insert, update, and delete rows

**How to use JDBC to work with databases**

- How to work with JDBC
- An introduction to database drivers
- How to connect to a database
- How to return a result set and move the cursor through it
- How to get data from a result set
- How to insert, update, and delete data
- How to work with prepared statements
- Two classes for working with databases
- The DBUtil class
- The ProductDB class
- Code that uses the ProductDB class

**How to develop a GUI with Swing (part 1)**

- An introduction to GUI programming
- A summary of GUI toolkits
- The inheritance hierarchy for Swing components
- How to create a GUI that handles events
- How to display a frame
- How to add a panel to a frame
- How to add buttons to a panel
- How to handle a button event
- How to work with layout managers
- A summary of layout managers
- How to use the FlowLayout manager
- How to use the BorderLayout manager
- How to work with tables
- How to create a model for a table
- The ProductTableModel class
- How to create a table
- How to get the selected row or rows
- How to add scrollbars to a table
- How to work with built-in dialog boxes
- How to display a message
- How to confirm an operation
- The Product Manager frame
- The user interface
- The ProductManagerFrame class

**How to develop a GUI with Swing (part 2)**

- How to work with labels and text fields
- How to work with labels
- How to work with text fields

- How to use the GridBagLayout manager
- An introduction to the GridBagLayout manager
- How to lay out components in a grid
- How to add padding
- How to avoid a common pitfall
- How to code a data entry form
- How to create a custom dialog
- How to pass data between a dialog and its parent
- The Product form
- The user interface
- The ProductForm class
- Two methods that use the ProductForm class
- How to use threads with Swing
- A common problem
- How to solve the problem