The Java™ Tutorials

Trail: Bonus
Lesson: Preparation for Java Programmer Language Certification

The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases.

Java SE 8 Programmer I Exam

This page maps sections in the Java Tutorials to topics covered in the Java SE 8 Programmer I exam. This exam is associated with the Oracle Certified Associate, Java SE 8 Programmer certificate. The topics covered in this exam are:

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

Section 1: Java Basics

Item 1: Define the scope of variables.

- Variables

Item 2: Define the structure of a Java class.

- A Closer Look at the "Hello World!" Application
- Classes

Item 3: Create executable Java applications with a main method; run a Java program from the command line; produce console output.

- "Hello World!" for the NetBeans IDE
- "Hello World!" for Microsoft Windows
- "Hello World!" for Solaris OS and Linux
- A Closer Look at the "Hello World!" Application

Item 4: Import other Java packages to make them accessible in your code.

- Creating and Using Packages
- Using Package Members

Item 5: Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation, etc.

- About the Java Technology
- Object-Oriented Programming Concepts

Section 2: Working with Java Data Types

Item 1: Declare and initialize variables (including casting of primitive data types).

- Variables
- Initializing Fields

Item 2: Differentiate between object reference variables and primitive variables.

- Primitive Data Types
- The Numbers Classes

Item 3: Know how to read or write to object fields.

- Inheritance

- Declaring Member Variables
- Creating Objects
- Using Objects

Item 4: Explain an object's lifecycle (creation, "dereference by reassignment" and garbage collection).

- Objects
- Creating Objects
- Using Objects

Section 3: Using Operators and Decision Constructs

Item 1: Use Java operators; use parentheses to override operator precedence.

- Operators
- Assignment, Arithmetic, and Unary Operators
- Equality, Relational, and Conditional Operators
- Bitwise and Bit Shift Operators
- Expressions, Statements, and Blocks

Item 2: Test equality between strings and other objects using `==` and `equals()`.

- Object as a Superclass

Item 3: Create and use `if`, `if-else`, and ternary constructs.

- The if-then and if-then-else Statements
- Equality, Relational, and Conditional Operators

Item 4: Use a `switch` statement.

- The switch Statement

Section 4: Creating and Using Arrays

Item 1: Declare, instantiate, initialize and use a one-dimensional array.

- Arrays

Item 2: Declare, instantiate, initialize and use a multi-dimensional array.

- Arrays

Section 5: Using Loop Constructs

Item 1: Create and use `while` loops.

- The while and do-while Statements

Item 2: Create and use `for` loops including the enhanced for loop.

- The for Statement

Item 3: Create and use `do-while` loops.

- The while and do-while Statements

Item 4: Compare loop constructs.

- Summary of Control Flow Statements

Item 5: Use `break` and `continue`.

- Branching Statements

Section 6: Working with Methods and Encapsulation

Item 1: Create methods with arguments and return values, including overloaded methods.

- Returning a Value from a Method
- Defining Methods

Item 2: Apply the `static` keyword to methods and fields.

- Variables
- Understanding Class Members
- Default Methods

Item 3: Create an overloaded method; differentiate between default and user defined constructors.

- Defining Methods
- Providing Constructors for Your Classes

Item 4: Apply access modifiers.

- Controlling Access to Members of a Class

Item 5: Apply encapsulation principles to a class.

- Inheritance
- Inner Class Example
- Nested Classes

Item 6: Determine the effect upon object references and primitive values when they are passed into methods that change the values.

- Passing Information to a Method or a Constructor

Section 7: Working with Inheritance

Item 1: Describe inheritance and its benefits.

- Inheritance
- Overriding and Hiding Methods

Item 2: 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.

- Polymorphism
- Creating Objects
- Using Objects

Item 3: Determine when casting is necessary.

- Inheritance

Item 4: Use `super` and `this` to access objects and constructors.

- Using the Keyword super
- Using the this Keyword

Item 5: Use abstract classes and interfaces.

- Abstract Methods and Classes
- Defining an Interface
- Implementing an Interface

Section 8: Handling Exceptions

Item 1: Differentiate among checked exceptions, `RuntimeException`, and `Error`.

- The Catch or Specify Requirement

Item 2: Create a `try-catch` block and determine how exceptions alter normal program flow.

- Catching and Handling Exceptions
- The try Block
- The catch Blocks

Item 3: Describe the advantages of exception handling .

- What Is an Exception?
- Advantages of Exceptions

Item 4: Create and invoke a method that throws an exception.

- Catching Exceptions

Item 5: Recognize common exception classes and categories (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException).

Section 9: Working with Selected classes from the Java API

Item 1: Manipulate data using the `StringBuilder` class and its methods.

- The StringBuilder Class
- Summary of Characters and Strings

Item 2: Create and manipulate strings.

- Strings
- Converting Between Numbers and Strings
- Comparing Strings and Portions of Strings
- Manipulating Characters in a String

Item 3: Create and manipulate calendar data using classes from `java.time.LocalDateTime`, `java.time.LocalDate`, `java.time.LocalTime`, `java.time.format.DateTimeFormatter`, `java.time.Period`.

- Date and Time Classes
- Date Classes
- Parsing and Formatting
- Period and Duration

Item 4: Declare and use an `ArrayList` of a given type.

- The List Interface
- List Implementations

Item 5: Write a simple Lambda expression that consumes a Lambda Predicate expression.

- Lambda Expressions
- Aggregate Operations

Previous page: Preparation for Java Programmer Language Certification
Next page: Java SE 8 Programmer II Exam