



Java Building Blocks

- Comments
- Main method
- Redundant Imports
- Naming Conflicts
- Package
- Constructors
- Instance Initializer Blocks
- Order of Initialization
- Primitive types
- Declaring Multiple Variables
- Identifiers
- Local Variables
- Instance and Class Variables
- Variable Scope
- Ordering Elements in a Class
- Garbage Collection

Operators and Statements

- Understanding Java Operators
- Numeric Promotion
- Logical Complement and Negation Operators
- Increment and Decrement Operators
- Casting Primitive Values
- Compound Assignment Operators
- Logical Operators
- Equality Operators
- The if Statement
- The if-then Statement (dangling else)
- Ternary Operator
- The switch statement
- Compile-time Constant Values
- The while Statement
- The do-while Statement
- The for Statement
- The for-each statement
- Adding Optional Labels
- The break statement
- The continue statement
- Unreachable and dead code

Core Java APIs

- Creating and Manipulating Strings

- Concatenation
- Immutability
- The String Pool
- Important String Methods
- Method Chaining
- StringBuilder Class
- Mutability and Chaining
- Creating a StringBuilder
- Important StringBuilder Methods
- StringBuilder vs. StringBuffer
- Understanding Equality
- Understanding Java Arrays
- Sorting Arrays
- Searching
- Creating a Multidimensional Array
- Using a Multidimensional Array
- Understanding an ArrayList
- ArrayList Methods
- Wrapper Classes
- Autoboxing
- Converting Between array and List
- Sorting list
- Working with Dates and Times
- Creating Dates and Times
- Manipulating Dates and Times
- Working with Periods
- Formatting Dates and Times

Methods and Encapsulation

- Designing Methods
- Access modifiers
- Optional Specifiers
- Return Type
- Method Name
- Working with Varargs
- Applying Access Modifiers
- Designing Static Methods and Fields
- Calling a Static Variable or Method
- Static vs Instance
- Static Imports
- Final Initialization
- Passing Data Among Methods
- Overloading Methods
- Overloading and Varargs
- Autoboxing



- Primitive and Reference Types
- Putting It All Together
- Creating Constructors
- Default Constructor
- Overloading Constructors
- Order of Initialization
- Encapsulating Data
- Creating Immutable Classes
- Writing Simple Lambdas
- Lambda Example
- Lambda Syntax
- Predicates

Class Design

- Introducing Class Inheritance
- Extending a Class
- Applying Class Access Modifiers
- Creating Java Objects
- Defining Constructors
- Understanding Compiler Enhancements
- Reviewing Constructor Rules
- Calling Constructors
- Calling Inherited Class Members
- `super()` vs `super`
- Overriding a Method
- Redeclaring private Methods
- Hiding Static Methods
- Overriding vs. Hiding Methods
- Creating final methods
- Inheriting Variables
- Hiding Variables
- Creating Abstract Classes
- Defining an Abstract Class
- Creating a Concrete Class
- Extending an Abstract Class
- Implementing Interfaces
- Defining an Interface
- Inheriting an Interface
- Classes, Interfaces, and Keywords
- Abstract Methods and Multiple Inheritance
- Interface Variables
- Default Interface Methods
- Default Methods and Multiple Inheritance
- Static Interface Methods

- Understanding Polymorphism
- Casting Objects
- Virtual Methods
- Polymorphic Parameters

Exceptions

- Understanding Exception Types
- Throwing an Exception
- Using a try Statement
- Adding a finally Block
- Catching Various Types of Exceptions
- Throwing a Second Exception
- Runtime Exceptions
- Checked Exceptions
- Errors
- Calling Methods That Throw Exceptions