

<i>Exam Number: 1Z0-803</i>	<i>Duration: 140 minutes</i>
<i>Associated Certifications: <u>Oracle Certified Associate, Java SE 7 Programmer</u></i>	<i>Number of Questions: 90</i>
<i>Exam Product: Java SE,</i>	<i>Passing Score: 77%</i>
<i>Exam Price: Rs 8014</i>	<i>Validated Against: This exam has been validated against SE 7.</i>
	<i>format: Multiple Choice</i>

## **Exam Topics:**

### **Java Basics**

- Define the scope of variables
- Define the structure of a Java class
- Create executable Java applications with a main method
- Import other Java packages to make them accessible in your code

### **Working With Java Data Types**

- Declare and initialize variables
- Differentiate between object reference variables and primitive variables
- Read or write to object fields
- Explain an Object's Lifecycle (creation, "dereference" and garbage collection)
- Call methods on objects
- Manipulate data using the StringBuilder class and its methods
- Creating and manipulating Strings

### **Using Operators and Decision Constructs**

- Use Java operators
- Use parenthesis to override operator precedence
- Test equality between Strings and other objects using == and equals ()
- Create if and if/else constructs
- Use a switch statement

## Creating and Using Arrays

- Declare, instantiate, initialize and use a one-dimensional array
- Declare, instantiate, initialize and use multi-dimensional array
- Declare and use an ArrayList

## Using Loop Constructs

- 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

## Working with Methods and Encapsulation

- Create methods with arguments and return values
- Apply the static keyword to methods and fields
- Create an overloaded method
- Differentiate between default and user defined constructors
- Create and overload 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

## Working with Inheritance

- Implement inheritance
- Develop code that demonstrates the use of polymorphism
- 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

## Handling Exceptions

- Differentiate among checked exceptions, RuntimeExceptions and Errors
- Create a try-catch block and determine how exceptions alter normal program flow
- Describe what Exceptions are used for in Java
- Invoke a method that throws an exception
- Recognize common exception classes and categories