

500+ questions with Explanations

# Java SE7 Associate Practice Exams

## 1Z0-803

With Exam Refresher



---

OCAJP Oracle© Certified Associate Java SE 7 Programmer  
Practice Exams

(Exam Code 1Z0-803)

©Hanumant Deshmukh  
[www.enthuware.com](http://www.enthuware.com)

# Preface

---

At Enthware, we have been training Students for various Java certifications for the past ten years. Our highly advanced mock exam simulator is a well respected study tool in terms of quality, quantity, price, and features. While it is a full blown desktop application that offers unparalleled features, we realize that it may not be possible for everybody to study while being tied to a regular PC. We have received numerous requests to provide the same content in an eBook format. This book is an attempt to help people access the same mock exams on their mobile devices.

# Table of Contents

---

1. [Introduction](#)
2. [Exam Objectives](#)
3. [Taking the Actual Exam](#)
4. [Sample](#)
  1. [Sample Questions](#)
  2. [Sample Questions \(Answered\)](#)
5. [Standard Tests](#)
  1. [Test 1](#)
  2. [Test 1 \(Answered\)](#)
  3. [Test 2](#)
  4. [Test 2 \(Answered\)](#)
  5. [Test 3](#)
  6. [Test 3 \(Answered\)](#)
  7. [Test 4](#)
  8. [Test 4 \(Answered\)](#)
  9. [Test 5](#)
  10. [Test 5 \(Answered\)](#)
  11. [Last Day Test \(Unique\)](#)
  12. [Last Day Test \(Unique\) \(Answered\)](#)
6. [Objective-wise Questions](#)
  1. [Constructors](#)
    1. [Constructors](#)
    2. [Constructors \(Answered\)](#)
  2. [Creating and Using Arrays](#)
    1. [Creating and Using Arrays](#)
    2. [Creating and Using Arrays \(Answered\)](#)
  3. [Encapsulation](#)
    1. [Encapsulation](#)
    2. [Encapsulation \(Answered\)](#)
  4. [Handling Exceptions](#)
    1. [Handling Exceptions](#)
    2. [Handling Exceptions \(Answered\)](#)
  5. [Java Basics](#)
    1. [Java Basics](#)
    2. [Java Basics \(Answered\)](#)
  6. [Java Basics - Garbage Collection](#)
    1. [Java Basics - Garbage Collection](#)
    2. [Java Basics - Garbage Collection \(Answered\)](#)
  7. [Overloading methods](#)
    1. [Overloading methods](#)
    2. [Overloading methods \(Answered\)](#)
  8. [Using Loop Constructs](#)
    1. [Using Loop Constructs](#)
    2. [Using Loop Constructs \(Answered\)](#)

9. [Using Operators and Decision Constructs](#)

---

  1. [Using Operators and Decision Constructs](#)
  2. [Using Operators and Decision Constructs \(Answered\)](#)
10. [Working with Inheritance](#)
  1. [Working with Inheritance](#)
  2. [Working with Inheritance \(Answered\)](#)
11. [Working with Java Data Types - String, StringBuilder](#)
  1. [Working with Java Data Types - String, StringBuilder](#)
  2. [Working with Java Data Types - String, StringBuilder \(Answered\)](#)
12. [Working with Java Data Types - Variables and Objects](#)
  1. [Working with Java Data Types - Variables and Objects](#)
  2. [Working with Java Data Types - Variables and Objects \(Answered\)](#)
13. [Working with Methods](#)
  1. [Working with Methods](#)
  2. [Working with Methods \(Answered\)](#)
14. [Working with Methods - Access Modifiers](#)
  1. [Working with Methods - Access Modifiers](#)
  2. [Working with Methods - Access Modifiers \(Answered\)](#)
7. [Exam Refresher](#)
8. [About the Author](#)

# Introduction

---

Oracle ® has significantly altered the Java Certification track with the release of Java SE 7. The entry level certification for Java Programmers is now broken up into two levels.

The first level is called Java Programmer - I, and confers an Associate Level certificate. The full name of this certification is *Oracle Certified Associate, Java SE 7 Programmer* (Exam Number: 1Z0-803). The second level is called, well, Java Programmer - II, and confers a Professional Level certificate. The full name of this certification is *Oracle Certified Professional, Java SE 7 Programmer* (Exam Number: 1Z0-804). One must first acquire the Associate Level certificate before going for the Professional Level. This book focuses on the Level 1 certification exam, i.e. OCA - Java SE 7 Programmer Certification.

## Who should use this book

This book is for OCA-JP SE7 certification aspirants. If you are a Java programmer with a couple of years of experience and if you are confident about your basic Java programming skills, you should take the mock exams in this book before attempting the real exam. The breadth of topics covered in this exam isn't much and the toughness level isn't too high. If you know basic Java programming, and are at least familiar with all the [exam objectives](#), you will sail through the exam. You don't need any specific Certification Study Guide for this exam. The only caveat is that the exam is really lengthy. You have to answer 90 questions in 2.5 hours. If you haven't taken a lot of online tests recently, you need to practice. This is exactly what this book is for. It will make you ready for the real exam in a couple of weeks.

If you are a Java beginner, you should use this book as a supplement to which ever regular Java programming book you are going through. This book is not a tutorial or a guide and it is not meant to replace a regular Java book. It is not meant to teach you the basics of Java programming. You should use this book to check how well you are learning the concepts by answering questions for any given exam objective. For example, if you are done studying the topic of Constructors from another book, you should attempt the questions given in this book on this topic and check how much you've learnt. If you have already gone through a book, you may attempt a complete mock exam and see how you score. If you pass the mock exam, you may proceed with the next mock exam otherwise, you need to go back to your regular Java programming book and read up on the topics on which you scored less.

## How to prepare for OCA-JP

As mentioned before, if you are a Java programmer with a couple of years of experience, you don't need any specific Certification Study Guide. You can start with the mock exams in this book straight away.

If you are a complete beginner, you should first go through any Java book for beginners. Write a lot of short and simple programs to understand the concepts. Check out the [exam objectives](#) and read up on these topics from any book or online tutorial. We recommend [Thinking in Java](#) by **Bruce Eckel**. It is, in our opinion, the best book for learning Java. The first seven chapters are free and a must read for a Java beginner. Remember, the exam isn't tough in terms of the complexity of the questions. Once you are comfortable with all the topics, you may start with the first Standard Test.

---

## **A note about Old SCJA Resources**

While the name of the new certification is similar to the name of the previous entry level certification called SCJA, which is short for "Sun Certified Java Associate for Java SE5/SE6", they have little in common. The exam objectives, style of questions, the toughness level of questions, number of questions, and time limit, are all different. So if you have a book for the old version of the certification, be careful not to get distracted by topics such as UML, EJB, JSP/Servlet, JDBC, HTML, and Swing.

## **How is this book organized?**

This book contains full sized mock exams that mimic the style and toughness level of the real exam. All such mock exams are under Standard Tests.

Every question is also categorized under the exam objective that it covers. So if you are following a book and want to reinforce your understanding about a topic, you may attempt to answer questions on that particular topic.

### **Duplicate Questions**

Standard Tests and Objective wise sets contain the same questions. So, depending on your mode of preparation, you should either attempt Standard Tests or Objective wise questions. If you have already attempted Objective wise questions, taking Standard Tests is of no use. You would have seen all the questions and your score will not be a real measure of your preparation.

**Last Day Test**, however, is an exception. We have made this test completely unique. Questions in this test are not included in Objective wise sets and so even if you have attempted all the objective wise questions, you may still attempt this test.

### **Taking the mock exams**

You should start with the first Standard Test. Your score on this test will give you a fair idea of how well you are prepared for the exam. Ideally, you should score more than 77%\* on this exam before moving on to the next exam. We have included a lot of reading material with the questions and you should go through the detailed explanation for each question...even for questions that you've answered correctly. Your objective should be to improve your score on the topics on which you scored less in this test.

If you fail in a standard test, you should not move on to the next test. Instead, first read up on the topics in which you failed from any book, write some sample programs to reinforce the concepts, and then attempt the next test.

\*Oracle has recently changed the passing marks from 75% to 77% and it may change the passing marks again at any time. So it is a good idea to check the current passing percentage at the time of your exam.

At the end of your preparation, you should attempt the "Last Day Test". If you pass this test, you are ready for the real exam. Most of our users have scored 10% higher on the real exam.

---

That is all there is to this book. Happy Learning!

**-Hanumant Deshmukh and the rest of Enthware Team.**

P.S.1 If you have any doubt or feedback about any question, just click on the question id at the top of the question to see any discussion associated with that question on [Enthware discussion forum](#). If it hasn't been discussed before, feel free to post a message and we will try our best to help.

P.S.2 If you like this book, please do [leave a feedback here](#). This will motivate us to create ebooks for other certifications as well.



# Exam Objectives

---

The following are the exam objectives as of this writing. Oracle may tweak the objectives at any time, so please verify the current objectives published at [OCA-JP Certification Page at Oracle](#).

## 1. 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

## 2. 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
- Call methods on objects
- Manipulate data using the StringBuilder class and its methods
- Create and manipulate strings

## 3. Using Operators and Decision Constructs

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

## 4. 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

## 5. 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

## 6. 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

## 7. 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

## 8. **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

## Taking the Actual Exam

---

The exam is conducted by [Pearson VUE](#). You may pay for and schedule the exam online through their website.

### Type of Questions

All the questions in the exam are multiple choice questions and every question tells you how many option you have to select. There are no drag and drop or fill in the blanks type questions.

### Testing Software

The testing application is fairly straight forward. You can mark the questions, move forward and backward while answering the questions, change your answers, review questions, and finally submit the answers for evaluation. Before starting the test, it allows you to get acclimatized by presenting you with a test containing dummy questions. This dummy test does not eat up the time from the actual test so it is a good idea to use this feature and make yourself comfortable with the testing environment before starting the real test.

Overall, it is not something that you need to lose your sleep over. However, if you have never taken a computer based test before, we advise you to use our [Mock Exam Simulator](#) and simulate the real test on a computer at home. The simulator is meant for learning purpose and has a lot more features than the actual testing software, so it does not look exactly the same as the real test, but it will give you a decent idea of what to expect. Solving questions in a book without anybody keeping the time and taking a test on a computer are two different things. This test is particularly lengthy and it is easy to lose track of time. Using the simulator will help you in determining how quick or slow you are in answering the questions.

Finally, don't worry too much about the test. Practice all the questions in this book and you will be fine :)



# Sample

---

This section contains only a few questions for sampling the book.

01. QID-2.904

---

Given:

```
class Triangle{
    public int base;
    public int height;
    private static double ANGLE = 0.0;

    public static double getAngle();

    public static void Main(String[] args) {
        System.out.println(getAngle());
    }
}
```

Identify the correct statements:

**Select 1 option**

- A. It will not compile because it does not implement setAngle method.
- B. It will not compile because ANGLE cannot be private.
- C. It will not compile because getAngle() has no body.
- D. It will not compile because ANGLE field is not initialized.
- E. It will not compile because of the name of the method Main instead of main.

[Check Answer](#)

## 02. QID - [2.1184](#)

---

Given:

```
StringBuilder b1 = new StringBuilder("snorkler");  
StringBuilder b2 = new StringBuilder("yoodler");
```

Write the contents of b1 and b2 after the statements shown on the left are executed independent of each other.

Statements	Contents of b1	Contents of b2
<pre>b1.append(b2.substring(2, 5)) .toUpperCase();</pre>	<input type="text"/>	<input type="text"/>
<pre>b2.insert(3, b1.append("a"));</pre>	<input type="text"/>	<input type="text"/>
<pre>b1.replace(3, 4, b2.substring(4)) .append(b2.append(false));</pre>	<input type="text"/>	<input type="text"/>

[Check Answer](#)

**03. QID - [2.1112](#)**

---

What will be the result of compiling and running the following program ?

```
class NewException extends Exception {}

class AnotherException extends Exception {}

public class ExceptionTest{
    public static void main(String[] args) throws Exception{
        try{
            m2();
        }
        finally{
            m3();
        }
        catch (NewException e){}
    }

    public static void m2() throws NewException { throw new NewException(); }

    public static void m3() throws AnotherException{ throw new AnotherException(); }
}
```

**Select 1 option**

- A.** It will compile but will throw AnotherException when run.
- B.** It will compile but will throw NewException when run.
- C.** It will compile and run without throwing any exceptions.
- D.** It will not compile.
- E.** None of the above.

[Check Answer](#)



Which of the following are true about the "default" constructor?

**Select 2 options**

- A. It is provided by the compiler only if the class does not define any constructor.
- B. It initializes the instance members of the class.
- C. It calls the default 'no-args' constructor of the super class.
- D. It initializes instance as well as class fields of the class.
- E. It is provided by the compiler if the class does not define a 'no- args' constructor.

[Check Answer](#)

05. QID - [2.1258](#)

---

Consider the following code:

```
class Test{
    public static void main(String[] args){
        for (int i = 0; i < args.length; i++)    System.out.print(i == 0 ? args[i] : "
+ args[i]);
    }
}
```

What will be the output when it is run using the following command:

```
java Test good bye friend!
```

**Select 1 option**

**A.** good bye friend!

**B.** good good good

**C.** goodgoodgood

**D.** good bye

**E.** None of the above.

[Check Answer](#)

06. QID - [2.962](#)

---

Consider the following method...

```
public int setVar(int a, int b, float c) { ...}
```

Which of the following methods correctly overload the above method?

**Select 2 options**

**A.**

```
public int setVar(int a, float b, int c){  
    return setVar(a, c, b);  
}
```

**B.**

```
public int setVar(int a, float b, int c){  
    return this(a, c, b);  
}
```

**C.**

```
public int setVar(int x, int y, float z){  
    return x+y;  
}
```

**D.**

```
public float setVar(int a, int b, float c){  
    return c*a;  
}
```

**E.**

```
public float setVar(int a){  
    return a;  
}
```

[Check Answer](#)

**07. QID - [2.1149](#)**

---

Given the following class definitions :

```
interface MyIface{};
class A {};
class B extends A implements MyIface{};
class C implements MyIface{};
```

and the following object instantiations,

```
A a = new A();
B b = new B();
C c = new C();
```

Which of these assignments are legal at compile time?

**Select 1 option**

**A.** `b = c;`

**B.** `c = b;`

**C.** `MyIface i = c;`

**D.** `c = (C) b;`

**E.** `b = a;`

[Check Answer](#)

**08. QID - [2.879](#)**

---

What will the following code print?

```
int[] scores1 = { 1, 2, 3, 4, 5, 6};  
int[] scores2 = { 0, 0, 0, 0, 0, 0};  
System.arraycopy(scores2, 2, scores1, 3, 2);  
for(int i : scores2) System.out.print(i);
```

**Select 1 option**

**A.** 123006

**B.** 000000

**C.** 000450

**D.** It throw an exception at run time.

[Check Answer](#)

## 09. QID - [2.876](#)

---

Consider the following two classes (in the same package but defined in different source files):

```
public class Square {
    double side = 0;
    double area;

    public Square(double length){        this.side = length;    }

    public double getSide() { return side;    }

    public void setSide(double side) { this.side = side;    }

    double getArea() { return area;    }
}
public class TestClass {
    public static void main(String[] args) throws Exception {
        Square sq = new Square(10.0);
        sq.area = sq.getSide()*sq.getSide();
        System.out.println(sq.getArea());
    }
}
```

You are assigned the task of refactoring the Square class to make it better in terms of encapsulation. What changes will you make to this class?

### Select 2 options

- A. Make setSide() method private.
- B. Make getArea() method private.
- C. Make side and area fields private.
- D. Make the side field private and remove the area field.
- E. Change getArea method to:  
`public double getArea(){ return side*side; }`
- F. Add a setArea() method.

[Check Answer](#)

**10. QID - [2.909](#)**

---

Which of the following declaration are valid:

1. `bool b = null;`
2. `boolean b = 1;`
3. `boolean b = true|false;`
- 4 `bool b = (10<11);`
5. `boolean b = true||false;`

**Select 1 option**

**A.** 1 and 4

**B.** 2, 3, and 5

**C.** 2 and 3

**D.** 3 and 5

**E.** 5

[Check Answer](#)

## 11. QID - [2.1083](#)

---

What will the following program print?

```
class LoopTest{
    public static void main(String args[]) {
        int counter = 0;
        outer:
        for (int i = 0; i < 3; i++) {
            middle:
            for (int j = 0; j < 3; j++) {
                inner:
                for (int k = 0; k < 3; k++) {
                    if (k - j > 0) {
                        break middle;
                    }
                    counter++;
                }
            }
        }
        System.out.println(counter);
    }
}
```

**Select 1 option**

**A. 2**

**B. 3**

**C. 6**

**D. 7**

**E. 9**

[Check Answer](#)



- [read online Hades' Gate \(Marius' Mules, Book 5\) pdf, azw \(kindle\), epub, doc, mobi](#)
- [download Answering Jihad: A Better Way Forward online](#)
- [read online Lover Eternal \(Black Dagger Brotherhood, Book 2\)](#)
- [read online Kiss of the Night \(Carnal Thirst, Book 2\)](#)
- [The Money Code: Become a Millionaire With the Ancient Jewish Code pdf, azw \(kindle\)](#)
  
- <http://cambridgebrass.com/?freebooks/Hades--Gate--Marius--Mules--Book-5-.pdf>
- <http://paulczajak.com/?library/Answering-Jihad--A-Better-Way-Forward.pdf>
- <http://growingsomeroots.com/ebooks/HarperCollins-Bible-Dictionary--Revised---Updated-.pdf>
- <http://junkrobots.com/ebooks/The-Ancient-Languages-of-Asia-Minor.pdf>
- <http://ramazotti.ru/library/The-Money-Code--Become-a-Millionaire-With-the-Ancient-Jewish-Code.pdf>