

## JAVA PROGRAMMING COURSE



The **Java Programmer Course** covers the Latest Sun Microsystem's Certification syllabus and is designed to help students to prepare for and pass Sun Microsystem's SCJP certification for Java 6 (CX-310-065) exam.

The exam's objectives touch on many of the more commonly used of Java's APIs. This course book follows closely both the breadth and the depth of the real exam. After completing the course, you should feel confident that you have thoroughly reviewed all of the objectives that Sun has established for the exam.

The certification program in Java technology is an industry recognised, worldwide program that focuses on critical job roles in software application development and enterprise architecture. Since these certifications focus on the technology, the knowledge and skills learned while preparing for Sun certifications are transportable from one company to another.

Achieving this certification provides clear evidence that a programmer understands the basic syntax and structure of the Java programming language and can create Java technology applications that run on server and desktop systems using Java SE 6.

The course materials used are part of the official CIW Course materials which includes the Java Programming Certification as part of its Enterprise Developer Professional syllabus. CIW certification is a worldwide credential that establishes an individual as an expert in Internet technologies.

### The course is split into the following Units:

#### ➤ **Unit One – Declarations and Access Control**

The following topics are covered: Java Refresher; Identifiers & JavaBeans (Objectives 1.3 and 1.4); Legal Identifiers; Sun's Java Code Conventions; JavaBeans Standards; Declare Classes (Exam Objective 1.1); Source File Declaration Rules; Class Declarations and Modifiers; Exercise 1-1: Creating an Abstract Superclass and Concrete Subclass; Declare Interfaces (Exam Objectives 1.1 and 1.2); Declaring an Interface; Declaring Interface Constants; Declare Class Members (Objectives 1.3 and 1.4); Access Modifiers; Nonaccess Member Modifiers; Constructor Declarations; Variable Declarations; Declaring Enums; Self-Test; & Self Test Answers.

#### ➤ **Unit Two – Object Orientation**

The following topics are covered: Encapsulation (Exam Objective 5.1); Inheritance, Is-A, Has-A (Exam Objective 5.5); IS-A; HAS-A; Polymorphism (Exam Objective 5.2); Overriding / Overloading (Exam Objectives 1.5 and 5.4); Overridden Methods; Overloaded Methods; Reference Variable Casting (Objective 5.2); Implementing an Interface (Exam Objective 1.2); Legal Return Types (Exam Objective 1.5); Return Type Declarations; Returning a Value; Constructors and Instantiation; (Exam Objectives 1.6, 5.3, and 5.4); Determine Whether a Default; Constructor Will Be Created; Overloaded Constructors; Statics (Exam Objective 1.3); Static Variables and Methods; Coupling and Cohesion (Exam Objective 5.1); Self-Test; & Self Test Answers.

**Price:**  
£250.00

**Instalment Options:**  
You can spread the payments for this course over 4 monthly payments. 1 initial payment of £100.00, followed by 3 monthly payments of £50.00.

**Course Format:**  
Course Book

**Assessment:**  
Java Platform 6 (CX-310-065)  
Examination

**Approximate Study Time:**  
120 Hours of Self Study

## ➤ **Unit Three – Assignments**

The following topics are covered: Stack and Heap—Quick Review; Literals, Assignments, and Variables (Exam Objectives 1.3 and 7.6); Literal Values for All Primitive Types; Assignment Operators; Exercise 3-1: Casting Primitives; Using a Variable or Array Element; That Is Uninitialized and Unassigned; Local (Stack, Automatic) Primitives and Objects; Passing Variables into Methods (Objective 7.3); Passing Object Reference Variables; Does Java Use Pass-By-Value; Semantics?; Passing Primitive Variables; Array Declaration, Construction, and Initialization (Exam Objective 1.3); Declaring an Array; Constructing an Array; Initializing an Array; Initialization Blocks; Using Wrapper Classes and Boxing (Exam Objective 3.1); An Overview of the Wrapper Classes; Creating Wrapper Objects; Using Wrapper Conversion Utilities; Autoboxing; Overloading (Exam Objectives 1.5 and 5.4); Garbage Collection (Exam Objective 7.4); Overview of Memory Management and Garbage Collection; Overview of Java's Garbage Collector; Writing Code That Explicitly Makes Objects Eligible for Collection; Exercise 3-2: Garbage Collection Experiment; Self-Test; & Self Test Answers.

## ➤ **Unit Four – Operators**

The following topics are covered: Java Operators (Exam Objective 7.6); Assignment Operators; Relational Operators; instance of Comparison; Arithmetic Operators; Conditional Operator; Logical Operators; Self-Test; & Self Test Answers.

## ➤ **Unit Five – Flow Control, Exceptions, and Assertions**

The following topics are covered: if and switch Statements (Exam Objective 2.1); if-else Branching; switch Statements; Exercise 5-1: Creating a switch-case Statement; Loops and Iterators (Exam Objective 2.2); Using while Loops; Using do Loops; Using for Loops; Using break and continue; Unlabeled Statements; Labeled Statements; Exercise 5-2: Creating a Labeled while Loop; Handling Exceptions (Exam Objectives 2.4 and 2.5); Catching an Exception Using try and catch; Using finally; Propagating Uncaught Exceptions; Exercise 5-3: Propagating and Catching an Exception; Defining Exceptions; Exception Hierarchy; Handling an Entire Class Hierarchy of Exceptions; Exception Matching; Exception Declaration and the Public Interface; Rethrowing the Same Exception; Exercise 5-4: Creating an Exception; Common Exceptions and Errors (Exam Objective 2.6); Working with the Assertion Mechanism (Exam Objective 2.3); Assertions Overview; Enabling Assertions; Using Assertions Appropriately; Self-Test; & Self Test Answers.

## ➤ **Unit Six – Strings, I/O, Formatting, and Parsing**

The following topics are covered: String, StringBuilder, and StringBuffer (Exam Objective 3.1); The String Class; Important Facts About Strings and Memory; Important Methods in the String Class; The StringBuffer and StringBuilder Classes; Important Methods in the StringBuffer and StringBuilder Classes; File Navigation and I/O (Exam Objective 3.2); The java.io.Console Class; Serialization (Exam Objective 3.3); Dates, Numbers, and Currency (Exam Objective 3.4); Working with Dates, Numbers, and Currencies; Parsing, Tokenizing, and Formatting; (Exam Objective 3.5); A Search Tutorial; Locating Data via Pattern Matching Tokenizing; Formatting with printf() and format(); Self-Test; & Self Test Answers.

## ➤ **Unit Seven – Generics & Collections**

The following topics are covered: Overriding hashCode() and equals() (Objective 6.2); Overriding equals(); Overriding hashCode(); Collections (Exam Objective 6.1); So What Do You Do with a Collection?; List Interface; Set Interface; Map Interface; Queue Interface; Using the Collections Framework (Objectives 6.3 and 6.5); ArrayList Basics; and Autoboxing with Collections; Sorting Collections and Arrays; Navigating (Searching) TreeSets and TreeMaps; Other Navigation Methods; Backed Collections; Generic Types (Objectives 6.3 and 6.4); Generics and Legacy Code; Mixing Generic and Non-generic Collections; Polymorphism and Generics; Generic Methods; Generic Declarations; Self-Test; & Self Test Answers.

## ➤ **Unit Eight – Inner Classes**

The following topics are covered: Inner Classes; Coding a "Regular" Inner Class; Referencing the Inner or Outer Instance from Within the Inner Class; Method-Local Inner Classes; What a Method-Local Inner Object Can and Can't Do; Anonymous Inner Classes; Plain-Old Anonymous Inner Classes, Flavor One; Plain-Old Anonymous Inner Classes, Flavor Two; Argument-Defined Anonymous Inner Classes; Static Nested Classes; Instantiating Using Static Nested Classes; Self-Test; & Self Test Answers.

## ➤ **Unit Nine – Threads**

The following topics are covered: Defining, Instantiating, and Starting Threads (Objective 4.1); Defining a Thread; Instantiating a Thread; Starting a Thread; Thread States and Transitions (Objective 4.2); Thread States; Preventing Thread Execution; Sleeping; Exercise 9-1: Creating a Thread and Putting It to Sleep; Thread Priorities and `yield()` Synchronizing Code (Objective 4.3); Synchronization and Locks; Exercise 9-2: Synchronizing a Block of Code; Thread Deadlock; Thread Interaction (Objective 4.4); Using `notifyAll()`; When Many Threads May Be Waiting; Self-Test; & Self Test Answers.

## ➤ **Unit Ten – Development**

The following topics are covered: Using the `javac` and `java` Commands (Exam Objectives 7.1, 7.2, and 7.5); Compiling with `javac`; Launching Applications with `java`; Searching for Other Classes; JAR Files (Objective 7.5); JAR Files and Searching; Using Static Imports (Exam Objective 7.1); Static Imports; Self-Test; & Self Test Answers.

### **Pre-Requirements:**

It is recommended that you have knowledge of or have studied Java Fundamentals to prepare you specifically for the Sun Certified Programmer for Java Platform 6 (CX-310-065) exam.

### **Course Duration & Support:**

Students may register at any time. The courses are designed as self-study courses but if you have any problems you can email our email support. As the course is self study you can complete in as little or as long a time as you prefer, and we do not impose a cut-off date for study.

### **Assessment:**

Assessment is from the Java Platform 6 (CX-310-065) Examination.

### **Qualification:**

On Completion of the Java Programming Course, you can apply for your **internationally recognised Java Platform 6 (CX-310-065) Examination** which is currently **£159.00 + VAT**.

Exams must be sat at a registered Prometric testing centre. You can locate testing centres and schedule appointments on their Website:

**Prometric website** – <http://www.prometric.com/>

### **Examination Information:**

**Number of questions:** 60

**Length of test:** 180 minutes

**Passing score:** 58.33%

