CompuScholar, Inc.

Alignment to
"*MTA Exam 98-388: Introduction to Programming Using Java* "
Certification Exam Requirements

Oracle Exam Details:

Exam Title:	Introduction to Programming Using Java
Exam Code(s):	98-388
Exam Link:	Exam 98-388: Introduction to Programming Using Java

CompuScholar Course Details:

Course Title:	Java Programming (Abridged)
Course ISBN:	978-0-9887070-4-7
Course Year:	2019

OR

Course Title:	Java Programming (AP)
Course ISBN:	978-0-9887070-2-3
Course Year:	2019

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the course.

Note 2: Citation(s) for a "Lesson" refer to the "Lesson Text" elements and associated "Activities" within the course, unless otherwise noted. The "Instructional Video" components are supplements designed to introduce or re-enforce the main lesson concepts, and the Lesson Text contains full details.

Course Description

Candidates for this exam are application developers working with Java 6 SE or later, secondary and immediate post-secondary level students of software development, or entry-level software developers.

Candidates should have at least 150 hours or instruction or hands-on experience with Java, are familiar with its features and capabilities, and understand how to write, debug and maintain well-formed, well documented Java code.

EITHER the CompuScholar "Java Programming (Abridged)" OR the "Java Programming" courses can be used to prepare for this exam. All exam topics are covered by each course.

Exam Requirements

Understand Java fundamentals (15-20%)	CITATION(S)
Describe the use of main in a Java application	
signature of main, why it is static	Chapter 2, Lesson 2 Chapter 11, Lesson 3
how to consume an instance of your own class	N/A
command-line arguments	Chapter 6, Lesson 1
Perform basic input and output using standard packages	
print statements	Chapter 4, Lesson 3
import and use the Scanner class	Chapter 6, Lesson 2
Evaluate the scope of a variable	
declare a variable within a block, class, or method	Chapter 4, Lesson 2 Chapter 10, Lesson 2

Work with data types, variables, and expressions (40-45%)	CITATION(S)
Declare and use primitive data type variables	
data types, including byte, char, int, double, short, long, float, boolean	Chapter 4, Lessons 1-2
identify when precision is lost	Chapter 4, Lesson 2
initialization	Chapter 4, Lesson 2
how primitives differ from wrapper object types such as Integer and Boolean	Chapter 4, Lesson 2
Construct and evaluate code that manipulates strings	
string class and string literals	Chapter 5, Lesson 1
comparisons	Chapter 5, Lesson 2
concatenation	Chapter 5, Lesson 4
case and length	Chapter 5, Lesson 3

String.format methods	Chapter 5, Lesson 4
string operators	Chapter 5, Lesson 4
converting a primitive data type to a string	Chapter 5, Lessons 4-5
the immutable nature of strings	Chapter 5, Lesson 3
initialization	Chapter 5, Lesson 1
null	Chapter 5, Lesson 1
Construct and evaluate code that creates, iterates, and manipulates arrays and array lists	
one- and two-dimensional arrays, including initialization, null, size, iterating elements, accessing elements	Chapter 14, Lessons 1, 2, 5
array lists, including adding and removing elements, traversing the list	Chapter 14, Lessons 4, 5
Construct and evaluate code that performs parsing, casting and conversion	
implementing code that casts between primitive data types	Chapter 4, Lesson 2
converts primitive types to equivalent object types	Chapter 4, Lesson 2
parses strings to numbers	Chapter 5, Lesson 5
Construct and evaluate arithmetic expressions	
arithmetic operators	Chapter 4, Lesson 2
assignment	Chapter 4, Lesson 2
compound assignment operators	Chapter 4, Lesson 2
operator precedence	Chapter 7, Lesson 1

Implement flow control (15-20%)	CITATION(S)
Construct and evaluate code that uses branching statements	
if, else, else if	Chapter 7, Lesson 2

switch	Chapter 7, Lesson 3
single-line versus block	Chapter 7, Lesson 2
nesting	Chapter 7, Lesson 2
logical and relational operators	Chapter 7, Lessons 1-2
Construct and evaluate code that uses loops	
while	Chpater 7, Lesson 5
for	Chpater 7, Lesson 4
for each	Chapter 14, Lesson 5
do while	Chpater 7, Lesson 5
break and continue	Chpater 7, Lessons 3, 4, 5
nesting	Chpater 7, Lessons 2, 4, 5
logical, relational, and unary operators	Chapter 7, Lesson 1

Perform object-oriented programming (10-15%)	CITATION(S)
Construct and evaluate a class definition	
constructors	Chapter 11, Lesson 1
constructor overloading	N/A
one class per .java file	Chapter 2, Lesson 1
this keyword	Chapter 10, Lesson 2
inheritance and overriding at a basic level	Chapter 15
Declare, implement, and access data members in a class	
private, public, protected	Chapter 10, Lesson 3

instance data members	Chapter 10, Lesson 2
static data members	Chapter 11, Lesson 3
using static final to create constants	N/A
describe encapsulation	Chapter 10, Lesson 1
Declare, implement, and access methods	
private, public, protected	Chapter 10, Lesson 3
method parameters	Chapter 8, Lesson 2
return type; void; return value	Chapter 8, Lessons 2-3
instance methods	Chapter 8 Chapter 10, Lesson 2
static methods	Chapter 11, Lesson 3
overloading	Chapter 8, Lesson 2
Instantiate and use a class object in a program	
instantiation; initialization; null	Chapter 10, Lesson 2
accessing and modifying data members	Chapter 10, Lesson 2
accessing methods	Chapter 10, Lesson 2
accessing and modifying static members	Chapter 11, Lesson 3
importing packages and classes	Chapter 2, Lesson 4

Compile and debug code (5-10%)	CITATION(S)
Troubleshoot syntax errors, logic errors, and runtime errors	
print statement debugging	Chapter 9, Lesson 3
output from the javac command	Chapter 2, Lesson 3

analyzing code for logic errors	Chapter 9, Lesson 3
console exceptions after running the program	Chapter 9, Lessons 1-2
evaluating a stack trace	Chapter 9, Lesson 1
Implement exception handling	
try catch finally	Chapter 9, Lesson 2
exception class; exception class types	Chapter 9, Lesson 2
display exception information	Chapter 9, Lesson 2