



COURSE SYLLABUS

Java and J2EE
(SCJA Exam CX-310-019)



50 Cragwood Rd, Suite 350
South Plainfield, NJ 07080

Victoria Commons, 613 Hope Rd Building #5,
Eatontown, NJ 07724

130 Clinton Rd,
Fairfield, NJ 07004

Avtech Institute of Technology Course

Instructor:

Course Duration: 60

Date/Time:

Training Location:

Course: Java and J2EE/Sun Certified Associate for the Java Platform, Standard Edition, Exam Version 1.0 (CX-310-019)

Text / Lab Books

Recommended:



ePractice Certification Exam for the Sun(TM) Certified Associate for the Java Platform, Standard Edition, Exam Version 1.0 (WGS-PREX-J019C)

This Sun ePractice exam is comprised of 150 questions, divided into three forms of 50 questions each.

Get Sun Certified, Get Going: eLearning Certification Success Packages provide the value and flexibility you need for certification.

Some of the following resource websites are free:



DEVELOP SOFTWARE
WITH CONFIDENCE



Course Description

This course teaches the fundamentals, platform and the development of Web Services, Web 2.0 for the purpose to pass the Sun Exams as Sun Certified Associate for the Java Platform, Standard Edition, and Exam Version 1.0 (CX-310-019)

The subjects of this course are comprehensive solutions that can help ease the learning and certification process by providing a combination of Website information, on-hand labs and the training document provided in this class. It helps the programmers who interested in learning how to develop Java technology applications and who have prior experience programming in another language, and then demonstrating the newly acquired skills by taking the Sun Certified Programmer exam (Java SE 5.0 version) and the Professional for Programmers Migrating to Java and Have Experience in Another Language.

Students who can benefit from this course are candidates preparing for the Sun Certified Java Associate examination and the professionals interested in entering a career in application development or a software project management using Java technologies, and who want to demonstrate their newly acquired skill set by taking the Sun Certified Java Associate (SCJA) exam (version 1.0).

Sun Certified Java Associate (SCJA)
Sun Certified Java Programmer (SCJP)
Sun Certified Web Component Developer (SCWCD)
Sun Certified Business Component Developer (SCWCD)
Sun Certified Business Component Developer (SCBCD)

Learning Objectives

1.0 Section 1: Fundamental Object-Oriented Concepts

- 1.1. Describe, compare, and contrast primitives
 - 1.1.1. integer, floating point, Boolean, and character
 - 1.1.2. enumeration types, and objects
- 1.2. Describe, compare, and contrast concrete classes, abstract classes, and interfaces, and how inheritance applies to them
- 1.3. Describe, compare, and contrast class compositions, and associations (including multiplicity: (one-to-one, one-to-many, and many-to-many), and association navigation
- 1.4. Describe information hiding (using private attributes and methods)
 - 1.4.1. Encapsulation and exposing object functionality using public methods
 - 1.4.2. describe the JavaBeans conventions for setter and getter methods
- 1.5. Describe polymorphism as it applies to classes and interfaces
- 1.6. describe and apply the "program to an interface" principle.

2.0 Section 2: UML Representation of Object-Oriented Concepts

- 2.1. Recognize the UML representation of classes, (including attributes and operations, abstract classes, and interfaces)
- 2.2. UML representation of inheritance (both implementation and interface
- 2.3. UML representation of class member visibility modifiers (-/private and +/public)
- 2.4. Recognize the UML representation of class associations, compositions, association multiplicity indicators, and association navigation indicators

3.0 Section 3: Java Implementation of Object-Oriented Concepts

- 3.1. Develop code that uses primitives, enumeration types, and object references, and recognize literals of these types
- 3.2. Develop code that declares concrete classes, abstract classes, and interfaces,
- 3.3. Develop code that supports implementation and interface inheritance, code that declares instance attributes and methods, and code that uses the Java access modifiers: private and public
- 3.4. Develop code that implements simple class associations, code that implements multiplicity using arrays
- 3.5. Recognize code that implements compositions as opposed to simple associations, and code that correctly implements association navigation
- 3.6. Develop code that uses polymorphism for both classes and interfaces, and recognize code that uses the "program to an interface" principle

4.0 Section 4: Algorithm Design and Implementation

- 4.1. Describe, compare, and contrast these three fundamental types of statements: assignment, conditional, and iteration
- 4.2. Given a description of an algorithm, select the appropriate type of statement to design the algorithm
- 4.3. Given an algorithm as pseudo-code, determine the correct scope for a variable used in the algorithm, and develop code to declare variables in any of the following scopes:
 - 4.3.1. instance variable
 - 4.3.2. method parameter
 - 4.3.3. local variable
- 4.4. Given an algorithm as pseudo-code, develop method code that implements the algorithm using conditional statements (if and switch), iteration statements (for, for-each, while, and do-while), assignment statements, and break and continue statements to control the flow within switch and iteration statements
- 4.5. Given an algorithm with multiple inputs and an output, develop method code that implements the algorithm using method parameters, a return type, and the return statement, and recognize the effects when object references and primitives are passed into methods that modify them
- 4.6. Given an algorithm as pseudo-code, develop code that correctly applies the appropriate operators including assignment operators (limited to: =, +=, -=), arithmetic operators (limited to: +, -, *, /, %, ++, --), relational operators (limited to: <, <=, >, >=, ==, !=), logical operators (limited to: !, &&, ||) to produce a desired result
- 4.7. Write code that determines the equality of two objects or two primitives
- 4.8. Develop code that uses the concatenation operator (+), and the following methods from class String: charAt, indexOf, trim, substring, replace, length, startsWith, and endsWith

5.0 Section 5: Java Development Fundamentals

- 5.1. Describe the purpose of packages in the Java language
- 5.2. Recognize the proper use of import and package statements
- 5.3. Demonstrate the proper use of the "javac" command (including the command-line options: -d and -classpath)

- 5.4. Demonstrate the proper use of the "java" command (including the command-line options: -classpath, -D and -version)
- 5.5. Describe the purpose and types of classes for the following Java packages: java.awt, javax.swing, java.io, java.net, java.util

6.0 Section 6: Java Platforms and Integration Technologies

- 6.1. Distinguish the basic characteristics of the three Java platforms: J2SE, J2ME, and J2EE
- 6.2. Given a high-level architectural goal, select the appropriate Java platform or platforms
- 6.3. Describe at a high level the benefits and basic characteristics of RMI
- 6.4. Describe at a high level the benefits and basic characteristics of JDBC, SQL, and RDBMS technologies
- 6.5. Describe at a high level the benefits and basic characteristics of JNDI, messaging, and JMS technologies

7.0 Section 7: Client Technologies

- 7.1. Describe at a high level the basic characteristics, benefits and drawbacks of creating thin-clients using HTML and JavaScript and the related deployment issues and solutions
- 7.2. Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating clients using J2ME midlets
- 7.3. Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Applets
- 7.4. Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Swing

8.0 Section 8: Server Technologies

- 8.1. Describe at a high level the basic characteristics of: EJB, servlets, JSP, JMS, JNDI, SMTP, JAX-RPC, Web Services (including SOAP, UDDI, WSDL, and XML), and JavaMail
- 8.2. Describe at a high level the basic characteristics of servlet and JSP support for HTML thin-clients
- 8.3. Describe at a high level the use and basic characteristics of EJB session, entity and message-driven beans
- 8.4. Describe at a high level the fundamental benefits and drawbacks of using J2EE server-side technologies, and describe and compare the basic characteristics of the web-tier, business-tier, and EIS tier

Semester Credit Hours

_____ semester credit hours

Teaching Strategies

A variety of teaching strategies may be utilized in this course, including but not limited to, lecture, discussion, written classroom exercises, written lab exercises, performance based lab exercises, demonstrations, quizzes and examinations. Some quizzes may be entirely or contain lab based components. A mid-course and end course examination will be given.

Method of Evaluating Students

Grade Distribution

Class Attendance	10
Mid Term	30
Finals	50
Special Projects Makeup projects	10
Total	100%

Grading Policy

At the end of each course, each student is assigned a final grade as follows:

Point Range	Interpretation	Grade	Quality Points
90 – 100	Excellent	A	4.0
80 – 89	Very Good	B	3.0 – 3.9
70 – 79	Average	C	2.0 – 2.9
60 – 69	Poor	D	1.0 – 1.9
Below 60	Failure	F	0
N/A	Withdrawal	W	0
N/A	Pass	P	0
N/A	Incomplete	I	0

A student earning a grade of D or above is considered to have passed the course and is eligible to pursue further studies. A student receiving a grade of F has failed the course. A failed course must be repeated and passed to meet Avtech Institute's graduation requirements, in addition to an overall program GPA of 2.0.

Requirements for Successful Completion of the Course

At a minimum, students must achieve the following:

- A passing grade of **D** or above
- Completion of all required examinations
- Submission of all required lab exercises and projects and;
- Adherence to the school attendance policy.

Equipment Needed

Industry standard desktop computer for lab exercises.

Equipment Breakdown Lab room

Videos and Projector

Library Assignments

To be determined by the instructor.

Portfolio Assignment

Student program outcome portfolios are required to demonstrate student competencies. In conjunction with your course structure, please select a project/paper that best demonstrates what you have learned in this course and add it to your program portfolio.

Course Policies

Disruptive Behavior

Disruptive behavior is an activity that interferes with learning and teaching. Inappropriate talking during class, surfing inappropriate website, tardiness, cheating, alcohol or drug use, use of cell phone, playing loud music during class, etc. all disrupt the learning process.

Copyright Infringement

Specific exemptions to copyright infringement are made for student use in the context of learning activities. Graphic design students often download images from the Internet, or scan images from publications. As long as this work is for educational purpose, and subject to faculty permission, this is not a problem.

Plagiarism

Faculty cannot tolerate the *misrepresentation of work as the student's own*. This often involves the use by one student or another student's design, whether voluntarily or involuntarily. In the event that plagiarism is evident and documented, all students involved in the conscious decision to misrepresent work must receive an F as the grade for the project. A second occurrence may result in suspension for the rest of the quarter, and return to the school only after a review by the Academic Standards Committee.

Attendance

Attendance and Lateness

In education and the workplace, regular attendance is necessary if individuals are to excel. There is a direct correlation between attendance and academic success. Attendance is mandatory. All students must arrive on time and prepared to learn at each class session. At the faculty member's

discretion, students may be marked absent if they arrive more than 15 minutes late to any class. More than five absences in a class that meets twice per week or more than two absences in a class that meets once per week may result in a failure.

Make-Up Work

Late Projects and Homework

All projects and homework must be handed in on time. Homework should be emailed to your instructor if you are going to miss a class. Work that is submitted one week late will result in the loss of one full grade; and work that is submitted two weeks late will result in the loss of two full grades; more than two weeks late you will receive a failing grade on the project.