



Let's talk about certifications: SCJA

J. M. Silveira Neto

**Sun Campus Ambassador
Universidade Federal do Ceará, Brazil**

Agenda



What/Why/How

Sun Certifications

Java Certification

Solaris Operating System

SCJA

Resources

Mock Exam

What is a certification?

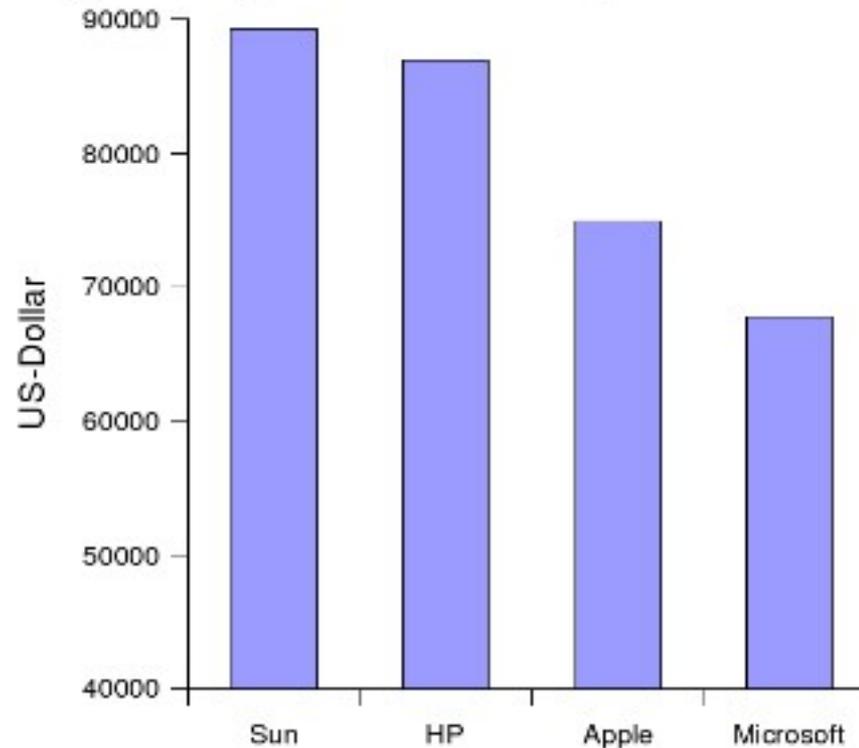
- " A designation earned by a person to demonstrates you have the competency, dedication and thrust that is valued in the highly competitive technology field.

Why Get Certified?

- " Knowledge
- " Differential CV/Resume
- " Industry Recognized
- " Academic Recognized
- " Salary

Why Get Certified?

Average salary of a certified System Administrator



CertMag's 2006 Salary Survey

more info:

http://www.certmag.com/articles/templates/CM_gen_Article_template.asp?articleid=2479&zoneid=224

How get a certification?

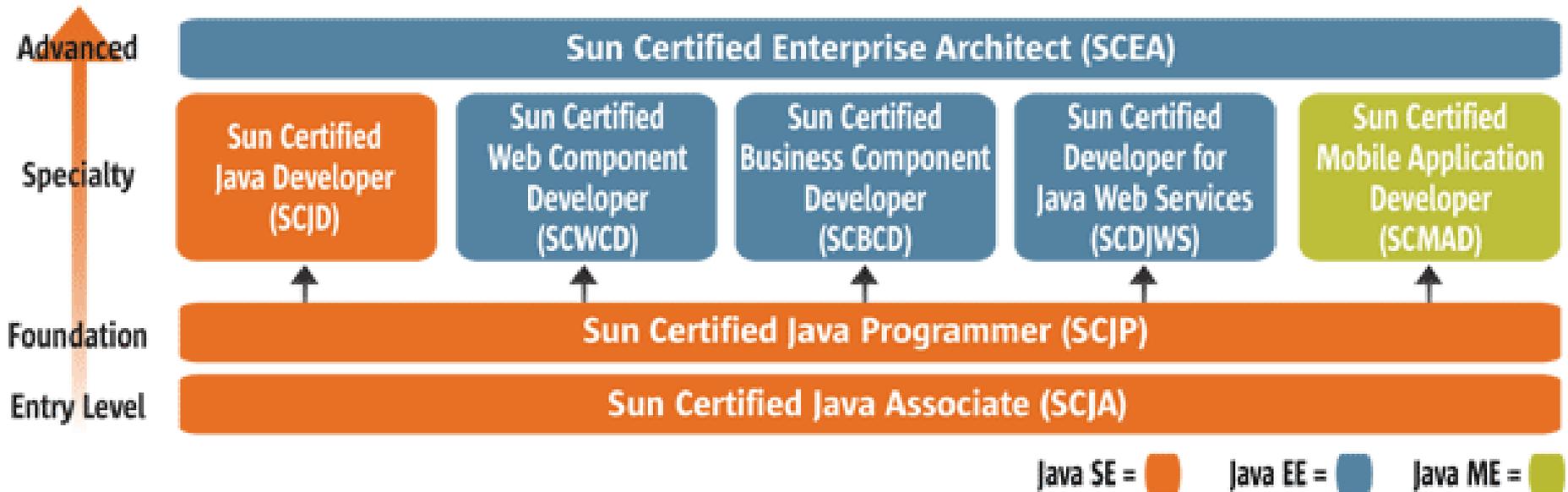
- " You need to pass a exam for the certification you want to get certified.
- " Depending on the type, the exam can be a project, a multiple choice exam or essay.
- " Once exam vouchers are purchased you have up to one year from the date of purchase to use it.
- " Authorized Prometric Testing Center

Sun Certifications

- " World-wide recognized
- " ~ 500,000 certificates
- " Java Technology
- " Solaris Operating System
- " Sun Cluster
- " Java CAPS

more info: <http://www.sun.com/training/certification/500k.xml>
<http://www.sun.com/training/certification/>

Java Certification



Solaris Operating System



Sun Certified Java Associate (SCJA)

- " An ideal entry into an application development or a software project management career using Java technologies.
- " It validates basic knowledge of OO concepts, the Java programming language and general knowledge of Java platforms and technologies.
- " Exam type: Multiple choice and Drag and Drop.
- " Number of questions: 51.
- " Pass score: 68% (35 of 51 questions).
- " Time limit: 115 minutes.
- " Languages: English, Japanese, Chinese (Traditional or Simplified), German, Korean, Brazilian Portuguese or Spanish.

SCJA Exam Objectives

- " **Section 1:** Fundamental Object-Oriented Concepts
- " **Section 2:** UML Representation of Object-Oriented Concepts
- " **Section 3:** Java Implementation of Object-Oriented Concepts
- " **Section 4:** Algorithm Design and Implementation
- " **Section 5:** Java Development Fundamentals
- " **Section 6:** Java Platforms and Integration Technologies
- " **Section 7:** Client Technologies
- " **Section 8:** Server Technologies

1. Fundamental Object-Oriented Concepts

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

2. UML Representation of Object-Oriented Concepts

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

3. Java Implementation of Object-Oriented Concepts

- " Notes: code examples may use the 'new' operator.
- " Develop code that uses primitives, enumeration types, and object references, and recognize literals of these types.
- " Develop code that declares concrete classes, abstract classes, and interfaces, 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.
- " Develop code that implements simple class associations, code that implements multiplicity using arrays, and recognize code that implements compositions as opposed to simple associations, and code that correctly implements association navigation.
- " Develop code that uses polymorphism for both classes and interfaces, and recognize code that uses the "program to an interface" principle

4. Algorithm Design and Implementation

- " Describe, compare, and contrast these three fundamental types of statements: assignment, conditional, and iteration, and given a description of an algorithm, select the appropriate type of statement to design the algorithm.
- " 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: instance variable, method parameter, and local variable.
- " 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.
- " 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. Algorithm Design and Implementation (cont.)

- " 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. Also, write code that determines the equality of two objects or two primitives.
- " Develop code that uses the concatenation operator (+), and the following methods from class String: charAt, indexOf, trim, substring, replace, length, startsWith, and endsWith.

5. Java Development Fundamentals

- " Describe the purpose of packages in the Java language, and recognize the proper use of import and package statements.
- " Demonstrate the proper use of the "javac" command (including the command-line options: -d and -classpath), and demonstrate the proper use of the "java" command (including the command-line options: -classpath, -D and -version).
- " Describe the purpose and types of classes for the following Java packages: java.awt, javax.swing, java.io, java.net, java.util.

6. Java Platforms and Integration Technologies

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

7. Client Technologies

- " 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.
- " Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating clients using J2ME midlets.
- " Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Applets.
- " Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Swing.

8. Server Technologies

- " 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.
- " Describe at a high level the basic characteristics of servlet and JSP support for HTML thin-clients.
- " Describe at a high level the use and basic characteristics of EJB session, entity and message-driven beans.
- " 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.

Certifications Resources

- " Trail: Learning the Java Language
 - > java.sun.com/docs/books/tutorial/java/index.html
- " UML for the Java Associate
 - > hfoobook.com/pdf/hfoo-associateuml.pdf
- " J2EE Specification, Section 2.6
 - > java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf
- " JavaRanch SCJA FAQ
 - > faq.javaranch.com/view?ScjaFaq
- " Student Developers, Certification Session
 - > developers.sun.com/students/certification.jsp
- " Certification Learning Paths
 - > sun.com//training/certification/resources/paths.html

Free Mock Exams

- " WGS-PREX-J019C, 150 questions.
 - > Free for Sun Academic Initiative (SAI)
 - > Sun Learning Connection
 - > Also free at uk.sun.com/training/catalog/courses/WGS-PREX-10-QUEST.xml
- " 25 free questions
 - > scja.de/ebook/order.html
- " 10 free questions.
 - > ejavaguru.com/scjafreemockexam.php
- " Caelum mock
 - > caelum.com.br/caelummock/

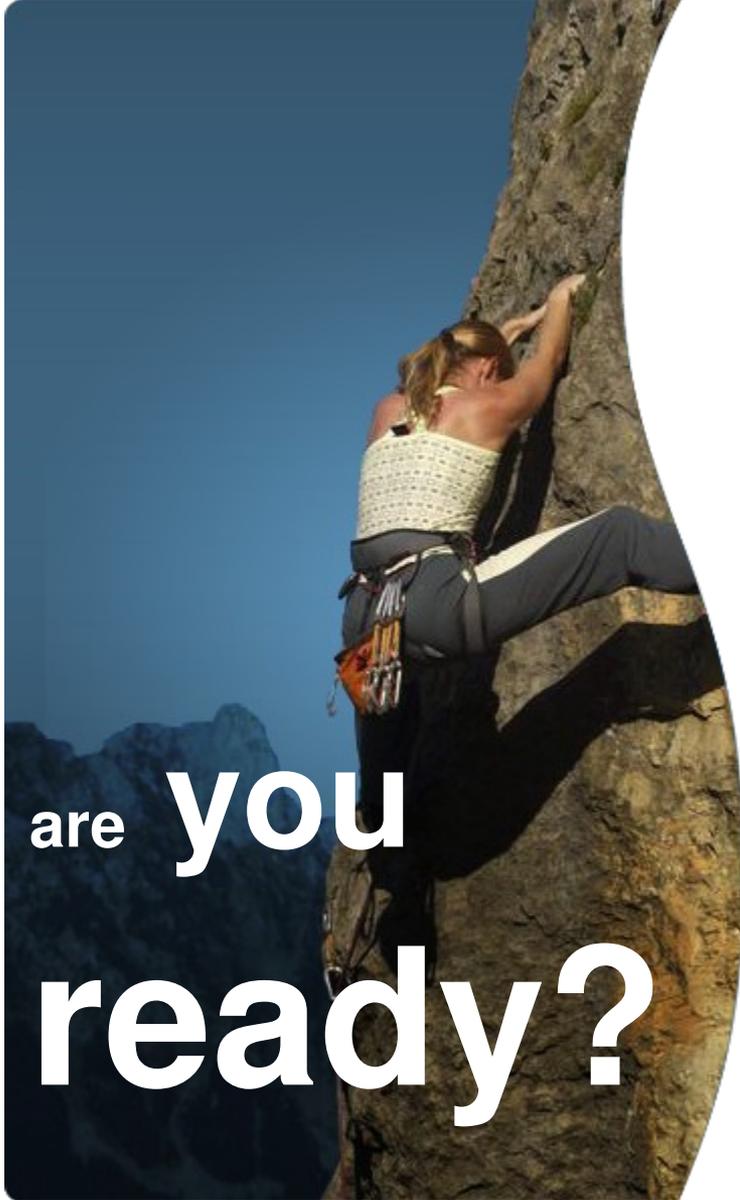
More resources

- " Local JUG (CEJUG)
- " Java Students Groups
- " Books
- " Mock Exams

How get discounts?

- " Sun Academic Initiative (SAI)
- " Local JUG Initiatives

Let's try some typical
SCJA questions now.

A person is climbing a rock face, wearing a white patterned tank top, dark pants, and a climbing harness. The background shows a clear blue sky and distant mountains.

are **you**
ready?

Mock Exam

Question 1

Select the correct setter and getter method for the property *speed* according to the JavaBeans convention.

```
public class Question {  
    private int count;  
}
```

Select all correct answers:

- A)** `public void setCount(int count) { this.count=count; }`
- B)** `public int setCount(int count) { return this.count; }`
- C)** `public void getCount(int count) { this.count=count; }`
- D)** `public int getCount() { return this.count; }`
- E)** `public int setCount() { return this.count; }`

Question 2

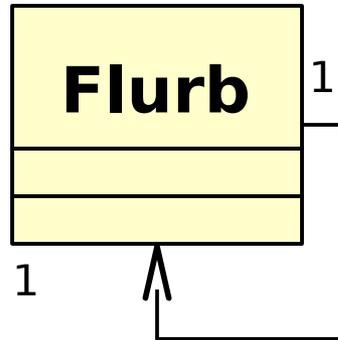
Object created using *new* is placed on

Select the correct answer

- A)** Stack
- B)** Queue
- C)** Heap
- D)** None of the above

Question 3

Which two are true?



Select two correct answers

- A)** Every Flurb contains a reference to itself.
- B)** Every Flurb is associated with exactly one Flurb.
- C)** There can only be one instance of the Flurb class.
- D)** There must always be an even number of Flurb instances.
- E)** A Flurb can be associated with itself, but it also could be associated with a different Flurb.

Question 4

Which two compile without error?

Select two correct answers

- A)** `boolean b = 0;`
- B)** `float f = 3.14;`
- C)** `double d = 1000;`
- D)** `char c = '\u0078';`

Question 5

Which two are true?

Select two correct answers

- A)** J2EE includes the MIDP API.
- B)** J2EE application developers need J2SE.
- C)** J2EE includes servlet APIs and EJB APIs.
- D)** J2EE applications depend on web servers.
- F)** J2EE runs on consumer and embedded devices.

Answers

- " **Question 1:** A, D.
- " **Question 2:** C.
- " **Question 3:** B, E.
- " **Question 4:** C, D.
- " **Question 5:** B, C.

?

Questions?

?



Thank you!

J. M. Silveira Neto

jose.neto@sun.com
MSN/Gtalk: silveiraneto@gmail.com
<http://silveiraneto.net>

Thanks



Flickr images:

flickr.com/photos/philgimp/370739618/



Creative Commons
Attribution Share Alike