Objectives for the Sun Certified Programmer for the Java™ 2 Platform

To assist candidates in their preparation, the following objectives are given to point candidates toward topics to study:

Section Title: Declarations and Access Control

- Write code that declares, constructs, and initializes arrays of any base type using any of the permitted forms both for declaration and for initialization.
- Declare classes, inner classes, methods, instance variables, static variables, and automatic (method local) variables making appropriate use of all permitted modifiers (such as public, final, static, abstract, and so forth). State the significance of each of these modifiers both singly and in combination, and state the effect of package relationships on declared items qualified by these modifiers.
- For a given class determine if a default constructor will be created and if so state the prototype of that constructor.
- State the legal return types for any method given the declarations of all related methods in this or parent classes.

Section Title: Flow Control and Exception Handling

- Write code using if and switch statements and identify legal argument types for these statements.
- Write code using all forms of loops including labeled and unlabeled use of break and continue and state the values taken by loop control variables during and after loop execution.
- Write code that makes proper use of exceptions and exception handling clauses (try, catch, finally) and declares methods and overriding methods that throw exceptions.

Section Title: Garbage Collection

- State the behavior that is guaranteed by the garbage collection system and write code that explicitly makes objects eligible for collection.

Section Title: Language Fundamentals

- Identify correctly constructed source files, package declarations, import statements, class declarations (of all forms including inner classes), interface declarations and implementations (for java.lang.Runnable or other interface described in the test), method declarations (including the main method that is used to start execution of a class), variable declarations and identifiers.

- State the correspondence between index values in the argument array passed to a main method and command line arguments.
- Identify all Java programming language keywords and correctly constructed identifiers.
- State the effect of using a variable or array element of any kind when no explicit assignment has been made to it.
- State the range of all primitive data types and declare literal values for String and all primitive types using all permitted formats, bases, and representations.
- Write code to implement listener classes and methods, and in listener methods, extract information from the event to determine the affected component, mouse position, nature, and time of the event. State the event classname for any specified event listener interface in the java.awt.event package.

Section Title: Operators and Assignments

- Determine the result of applying any operator, including assignment operators and instanceof, to operands of any type, class, scope, or accessibility, or any combination of these.
- Determine the result of applying the boolean equals(Object) method to objects of any combination of the classes java.lang.String, java.lang.Boolean, and java.lang.Object.
- In an expression involving the operators &, |, &&, ||, and variables of known values state which operands are evaluated and the value of the expression.
- Determine the effect upon objects and primitive values of passing variables into methods and performing assignments or other modifying operations in that method.

Section Title: Overloading Overriding Runtime Type and Object Orientation

- State the benefits of encapsulation in object oriented design and write code that implements tightly encapsulated classes and the relationships "is a" and "has a".
- Write code to invoke overridden or overloaded methods and parental or overloaded constructors; and describe the effect of invoking these methods.
- Write code to construct instances of any concrete class including normal top level classes, inner classes, static inner classes, and anonymous inner classes.

Section Title: Threads

- Write code to define, instantiate, and start new threads using both java.lang.Thread and java.lang.Runnable

- Recognize conditions that might prevent a thread from executing.
- Write code using synchronized, wait, notify, and notifyAll to protect against concurrent access problems and to communicate between threads. Define the interaction between threads and between threads and object locks when executing synchronized wait notify or notifyAll.

Section Title: The java.awt package - Layout

- Write code using component, container, and layout manager classes of the java.awt package to present a GUI with specified appearance and resize behavior, and distinguish the responsibilities of layout managers from those of containers.

Section Title: The java.lang package

- Write code using the following methods of the java.lang.Math class: abs, ceil, floor, max, min, random, round, sin, cos, tan, sqrt.
- Describe the significance of the immutability of String objects.

Section Title: The java.util package

- Make appropriate selection of collection classes/interfaces to suit specified behavior requirements.