Core Java
Contents

Exam Objectives
Section 1: 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, nested classes, methods, instance variables, static variables and automatic (method local) variables making appropriate use of all permitted modifiers (such as public, final, static, abstract, etc.). 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.

•Identify legal return types for any method given the declarations of all related methods in this or parent classes.

Section 2: Flow control, Assertions, 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 counter 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.

•Recognize the effect of an exception arising at a specified point in a code fragment. Note: The exception may be a runtime exception, a checked exception, or an error (the code may include try, catch, or finally clauses in any legitimate combination).

•Write code that makes proper use of assertions, and distinguish appropriate from inappropriate uses of assertions.

•Identify correct statements about the assertion mechanism.

Section 3: Garbage Collection
•State the behavior that is guaranteed by the garbage collection system.

•Write code that explicitly makes objects eligible for garbage collection.

•Recognize the point in a piece of source code at which an object becomes eligible for garbage collection.

Section 4: Language Fundamentals
•Identify correctly constructed package declarations, import statements, class declarations (of all forms including inner classes) interface declarations, method declarations (including the main method that is used to start execution of a class), variable declarations, and identifiers.

•Identify classes that correctly implement an interface where that interface is either java.lang.Runnable or a fully specified interface in the question.

•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. Note: There will not be any questions regarding esoteric distinctions between keywords and manifest constants.

•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 formats, data types and declare literal values for String and all primitive types using all permitted formats bases and representations.

Section 5: Operators and Assignments
•Determine the result of applying any operator (including assignment operators and instance of) 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 6: 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 and nested classes.

Section 7: 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 among threads and object locks when executing synchronized wait, notify or notifyAll.

Section 8: Fundamental Classes in 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.

•Describe the significance of wrapper classes, including making appropriate selections in the wrapper classes to suit specified behavior requirements, stating the result of executing a fragment of code that includes an instance of one of the wrapper classes, and writing code using the following methods of the wrapper classes (e.g., Integer, Double, etc.):
○o doubleValue
○o floatValue
○o intValue
○o longValue
○o parseXxx
○o getXxx
○o toString
○o toHexString

Section 9: The Collections Framework

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

•Distinguish between correct and incorrect implementations of hashcode methods.

<--Back **Home-->**