Objectives 7

Fundamentals

Static Imports (Objective 7.1)

- You must start a static import statement like this: import static
- You can use static imports to create shortcuts for static members (static variables, constants, and methods) of any class.

Using javac and java (Objective 7.2)

- Use -d to change the destination of a class file when it's first generated by the javac command.
- The -d option can build package-dependent destination classes on-the-fly if the *root* package directory already exists.
- Use the -D option in conjunction with the java command when you want to set a system property.
- System properties consist of name=value pairs that must be appended directly behind the -D, for example, java -Dmyproperty=myvalue.
- Command-line arguments are always treated as Strings.
- The java command-line argument 1 is put into array element 0, argument 2 is put into element 1, and so on.

Passing Variables into Methods (Objective 7.3)

- Methods can take primitives and/or object references as arguments.
- Method arguments are always copies.
- Method arguments are never actual objects (they can be references to objects).
- A primitive argument is an unattached copy of the original primitive.
- A reference argument is another copy of a reference to the original object.
- Shadowing occurs when two variables with different scopes share the same name. This leads to hard-to-find bugs, and hard-to-answer exam questions.

Garbage Collection (Objective 7.4)

- In Java, garbage collection (GC) provides automated memory management.
- The purpose of GC is to delete objects that can't be reached.
- Only the JVM decides when to run the GC, you can only suggest it.
- You can't know the GC algorithm for sure.
- Objects must be considered eligible before they can be garbage collected.
- An object is eligible when no live thread can reach it.
- To reach an object, you must have a live, reachable reference to that object.

Garbage Collection (Objective 7.4) [contd.]

- Java applications can run out of memory.
- Islands of objects can be GCed, even though they refer to each other.
- Request garbage collection with System.gc(); (recommended).
- Class Object has a finalize() method.
- The finalize() method is guaranteed to run once and only once before the garbage collector deletes an object.
- The garbage collector makes no guarantees, finalize() may never run.
- You can uneligibilize an object for GC from within finalize().

Searching with java and javac (Objective 7.5)

- Both java and javac use the same algorithms to search for classes.
- Searching begins in the locations that contain the classes that come standard with J2SE.
- Users can define secondary search locations using classpaths.
- Default classpaths can be defined by using OS environment variables.
- A classpath can be declared at the command line, and it overrides the default classpath.
- A single classpath can define many different search locations.
- In Unix classpaths, forward slashes (/) are used to separate the directories that make up a path. In Windows, backslashes (\) are used.
- In Unix, colons (:) are used to separate the paths within a classpath. In Windows, semicolons (;) are used.
- In a classpath, to specify the current directory as a search location, use a dot (.)
- In a classpath, once a class is found, searching stops, so the order of locations to search is important.

Packages and Searching (Objective 7.5)

- When a class is put into a package, its fully qualified name must be used.
- An import statement provides an alias to a class's fully qualified name.
- In order for a class to be located, its fully qualified name must have a tight relationship with the directory structure in which it resides.
- A classpath can contain both relative and absolute paths.
- An absolute path starts with a / or a \.
- Only the final directory in a given path will be searched.

JAR files (Objective 7.5)

- An entire directory tree structure can be archived in a single JAR file.
- JAR files can be searched by java and javac.
- When you include a JAR file in a classpath, you must include not only the directory in which the JAR file is located, but the name of the JAR file too.
- For testing purposes, you can put JAR files into .../jre/lib/ext, which is somewhere inside the Java directory tree on your machine.