MTA: Introduction to Programming Using Java – Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Exam 98-388: Introduction to Programming Using Java

Understand Java fundamentals (15-20%)

Describe the use of main in a Java application

- signature of main, why it is static; how to consume an instance of your own class; command-line arguments

Perform basic input and output using standard packages

- print statements; import and use the Scanner class

Evaluate the scope of a variable

- declare a variable within a block, class, or method

Work with data types, variables, and expressions (40-45%)

Declare and use primitive data type variables

- data types, including byte, char, int, double, short, long, float, boolean; identify when precision is lost; initialization; how primitives differ from wrapper object types such as Integer and Boolean

Construct and evaluate code that manipulates strings

- string class and string literals, comparisons, concatenation, case and length; String.format methods; string operators; converting a primitive data type to a string; the immutable nature of strings; initialization; null

Construct and evaluate code that creates, iterates, and manipulates arrays and array lists

- one- and two-dimensional arrays, including initialization, null, size, iterating elements, accessing elements; array lists, including adding and removing elements, traversing the list

Construct and evaluate code that performs parsing, casting and conversion

- implementing code that casts between primitive data types, converts primitive types to equivalent object types, or parses strings to numbers

Construct and evaluate arithmetic expressions

- arithmetic operators, assignment, compound assignment operators, operator precedence

Implement flow control (15-20%)

Construct and evaluate code that uses branching statements

- if, else, else if, switch; single-line versus block; nesting; logical and relational operators

Construct and evaluate code that uses loops

- while, for, for each, do while; break and continue; nesting; logical, relational, and unary operators

Perform object-oriented programming (10-15%)

Construct and evaluate a class definition

- constructors; constructor overloading; one class per .java file; this keyword; inheritance and overriding at a basic level

Declare, implement, and access data members in a class

- private, public, protected; instance data members; static data members; using static final to create constants; describe encapsulation

Declare, implement, and access methods

- private, public, protected; method parameters; return type; void; return value; instance methods; static methods; overloading

Instantiate and use a class object in a program

- instantiation; initialization; null; accessing and modifying data members; accessing methods; accessing and modifying static members; importing packages and classes

Compile and debug code (5-10%)

Troubleshoot syntax errors, logic errors, and runtime errors

- print statement debugging; output from the javac command; analyzing code for logic errors; console exceptions after running the program; evaluating a stack trace

Implement exception handling

- try catch finally; exception class; exception class types; display exception information