**Understanding Java Technology and environment**

* Describe Java Technology and the Java development
* Identify key features of the Java language

**Working With Java Primitive Data Types and String APIs**

* Declare and initialize variables (including casting and promoting primitive data types)
* Identify the scope of variables
* Use local variable type inference
* Create and manipulate Strings
* Manipulate data using the StringBuilder class and its methods

**Working with Java Arrays**

* Declare, instantiate, initialize and use a one-dimensional array
* Declare, instantiate, initialize and use a two-dimensional array

**Creating and Using Methods**

* Create methods and constructors with arguments and return values
* Create and invoke overloaded methods
* Apply the static keyword to methods and fields

**Reusing Implementations Through Inheritance**

* Create and use subclasses and superclasses
* Create and extend abstract classes
* Enable polymorphism by overriding methods
* Utilize polymorphism to cast and call methods, differentiating object type versus reference type
* Distinguish overloading, overriding, and hiding

**Handling Exceptions**

* Describe the advantages of Exception handling and differentiate among checked, unchecked exceptions, and Errors
* Create try-catch blocks and determine how exceptions alter program flow
* Create and invoke a method that throws an exception

**Creating a Simple Java Program**

* Create an executable Java program with a main class
* Compile and run a Java program from the command line
* Create and import packages

**Using Operators and Decision Constructs**

* Use Java operators including the use of parentheses to override operator precedence
* Use Java control statements including if, if/else, switch
* Create and use do/while, while, for and for each loops, including nested loops, use break and continue statements

**Describing and Using Objects and Classes**

* Declare and instantiate Java objects, and explain objects' lifecycles (including creation, dereferencing by reassignment, and garbage collection)
* Define the structure of a Java class
* Read or write to object fields

**Applying Encapsulation**

* Apply access modifiers
* Apply encapsulation principles to a class

**Programming Abstractly Through Interfaces**

* Create and implement interfaces
* Distinguish class inheritance from interface inheritance including abstract classes
* Declare and use List and ArrayList instances
* Understanding Lambda Expressions

**Understanding Modules**

* Describe the Modular JDK
* Declare modules and enable access between modules
* Describe how a modular project is compiled and run