Task 4: Java Types & JVM Management - Java programmers must understand the security implications of built-in data types and Java-specific memory management.
01.4.1 java.lang.String - Java programmers must have a complete mastery of the String class's immutability and how to compare String objects.
01.4.2 Integer and Double Overflows - Java programmers must understand the limitations of **Java**'s numerical data types and the resulting security implications.
01.4.3 Garbage Collector - Java programmers must have an understanding of how the Java Garbage Collector works and the resulting security implications.
01.4.4 ArrayList vs Vector - Java programmers must understand the differences and the resulting security considerations between the ArrayList and the Vector.
01.4.5 Class Security - Java programmers should be familiar with accessibility modifiers, the final modifier, class comparisons, serialization, clone-ability, and inner classes.
01.4.6 Code Privileges - Java Programmers must understand how to manage the privileges of code as well as the different protection domains. This includes an understanding of the Security Manager and its policy file.

Task 5: Application Faults & Logging - All Java application programmers need to be able to properly handle application faults.
01.5.1 Exception Handling - Java application developers must understand **Java's** try/catch/finally construct to appropriately handle application and system exceptions. Developers must determine how much information should be logged when an exception is encountered depending on the nature of the exception.
01.5.2 Logging - Developers must understand the principles behind logging security-relevant events such as login, logoff, credential changes, etc. Developers should also be familiar with **Java**'s logging package, java.util.logging.
01.5.3 Configuration of Error Handling - J2EE developers should be familiar with the configuration to return a default error page for HTTP 404 and 500 errors.

Task 6: Encryption Services - Java programmers must understand when and how to use encryption to protect sensitive data.
01.6.1 Communications Encryption - Java application developers must be familiar with the **Java** Secure Sockets Extension (JSSE) packages as well as how to configure SSL communication for J2EE applications. Developers are also responsible for knowing which of their application's external links should be protected with encryption.
01.6.2 Encryption of Data at Rest - Java developers must understand how to store sensitive data in encrypted format.

Task 7: Concurrency and Threading - Java programmers must understand how to properly structure multi-threaded programs.
01.7.1 Race Conditions - All **Java** application developers must understand race conditions and how they affect system security. This includes avoiding caching security relevant information that can be accessed by multiple threads.
01.7.2 Singletons & Shared Resources - Java developers must understand how to implement the Singleton pattern in **Java** and how to protect other resources that are accessed by multiple threads.

Task 8: Connection Patterns - Java programs must be able to securely interface with other applications. Developers must be familiar with parameterized queries, output encoding, and fail-safe connection patterns.
01.8.1 Parameterized Queries / PreparedStatements - Java programmers must understand the security risks introduced by using dynamic queries and how to safely use the PreparedStatement to safely and securely interact with databases based on user-supplied input.
01.8.2 Output Encoding - Java programmers must understand when and how to use output encoding to display data to user interfaces, as this is a primary mitigation technique to UI injection attacks, e.g. Cross-site Scripting.
01.8.3 Fail-safe Connection Patterns - Java programmers must properly form connection patterns using **Java**'s try/catch/finally to prevent resource leaks. Resource leaks can occur as a result of failures while operating with connections to external systems.

Task 9: Miscellaneous
01.9.1 Class/Package/Method Access Modifiers - All Java programmers must understand how the **Java** access modifiers (public, private, protected) can be used to protect class members and methods.
01.9.2 Class File Protection - Java programmers must understand how JAR sealing is used.
01.9.3 J2EE Filters - J2EE programmers must be familiar with J2EE Filters and how they can be used to implement many of the tasks listed above.