

CPS 189: Java Certification

Thank you for your time and expertise in reviewing this course. Any feedback and comments that you may be able to provide is appreciated.

Please enter your name and title.

Please enter the date.

Please submit your qualifications to review the content in this course.

This course was designed as a three week review course to help students prepare to take the Oracle Certified Java Programmer exam. Please look through the course content to determine if it covers each of the objectives listed below. Include any specific feedback, comments, and/or evidence for your response.

1. Analyze the interoperability of collections that use raw type and generic rules.

Explanation of response

2. Analyze the operability of collections that use raw type and generic types

Explanation of response

3. Build a resource bundle for each locale

Explanation of response

4. Use executor, Executor Service, Executors, Callable, and future to execute tasks using thread pools

Explanation of response

5. Check, delete, copy, or move a file or directory with the files class

Explanation of response

6. Choose between interface inheritance and class inheritance

Explanation of your response

7. Construct abstract java classes and subclasses

Explanation of response

8. Create and use RowSet objects using the RowSetProvider class and the RowSetFactory interface

Explanation of response

9. Create and use List, Set, and Deque Implementations

Explanation of response

10. Create and use Map implementations

Explanation of response

11. Create and use the thread class and the runnable interface

Explanation of response

12. Create custom exceptions

Explanation of response

13. Create top level of nested classes

Explanation of response

14. Describe the interfaces that make up the core of the JDBC API (including the driver, connection, statement, and resultset interfaces and their relationship to provider implementations)

Explanation of response

15. Design and create objects using a factory

Explanation of response

16. Design a class using the singleton design pattern

Explanation of response

17. Develop code that handles multiple exception types in a single catch block

Explanation of response

18. Develop code that implements "is-a" and/or "has/a" relationships

Explanation of response

19. Develop code that uses try-with-resources statements (including using classes that implement the AutoCloseable interface)

Explanation of response

20. Find a file with the PathMaker interface

Explanation of response

21. Format dates, numbers, and currency values for localization with the Number Format and Date Format classes (including number format patterns)

Explanation of response

22. Format strings using the formatting parameters: %b, %c, %d, %f, and %s in format strings

Explanation of response

23. Identify Code that may not execute correctly in a multi-threaded environment

Explanation of response

24. Identify when and how to apply abstract classes

Explanation of response

25. Manage and control thread lifecycle

Explanation of response

26. Load a resource bundle in an application

Explanation of response

27. Operate on file and directory paths with the path class

Explanation of response

28. Overload constructors and other methods appropriately

Explanation of response

29. Override methods

Explanation of response

30. Override the hashCode, equals, and toString methods from the Object class to improve the functionality of your class

Explanation of response

31. Read and change file and directory attributes, focusing on the BasicFileAttributes, DosFileAttributes, and PosixFileAttributes interfaces

Explanation of response

32. Read and set the locale by using the locale object

Explanation of response

33. Read and write data from the console

Explanation of response

34. Recursively access a directory tree using the Directory Stream and FileVisitor interfaces

Explanation of response

35. Search, parse, and build strings (including scanner, stringbuilder, string, and formatter)

Explanation of response

36. Search, parse, and replace strings by using regular expressions, using expression patterns for matching limited to: `.`(dot), `*`(star), `+`(plus), `?`, `\d`, `\D`, `\s`, `\S`, `\w`, `\W`, `\b`, `\B`, `[]`, `()`

Explanation of response

37. Sort and search arrays and lists

Explanation of response

38. Submit queries and read results from the database (including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections)

Explanation of response

39. Synchronize thread access to shared data

Explanation of response

40. Use JDBC transactions (including disabling auto-commit mode, committing and rolling back transactions, and setting and rolling back to savepoints)

Explanation of response

41. Test invariants by using assertions

Explanation of response

42. Use Lock, ReadWriteLock, and ReentrantLock classes in the java.util.concurrent.locks package to support lock-free thread-safe programming on single variables

Explanation of response

43. Use access modifiers: private, protected, and public

Explanation of response

44. Use collections from the java.util.concurrent package with a focus on the advantages over differences from the traditional java.util.collections

Explanation of response

45. Use package and import statements

Explanation of response

46. Use enumerated types

Explanation of response