|  |
| --- |
| Rate each topic on a scale of 0-5 |

|  |
| --- |
| **5 – (Mastery) expert error-free code / fix others’ bugs** |
| **4 – (Analyze) differentiate between facts and inferences** |
| **3 – (Apply) give correct application of concept in context**  **2 – (Understand) give explanation of topic** |
| **1 – (Familiar) remember or recognize the term or concept** |
| **0 – (Unfamiliar) haven't got a clue** |

|  |  |  |
| --- | --- | --- |
| 1. Abstraction | | \_2\_\_ |
| 1. ***abstract*** keyword | | \_1\_\_ |
| 1. Access modifiers | | \_1\_ |
| 1. Algorithms | \_3\_ | |
| 1. Arithmetic operators (+, -, \*, /, %) | \_3\_ | |
| 1. Array, Arrays, & ArrayList classes | \_3\_ | |
| 1. Arrays & array-initializer | \_3\_ | |
| 1. Assignment operator | \_2\_ | |
| 1. Binary number system | \_1\_ | |
| 1. Boolean logic | \_3\_ | |
| 1. Branches / Branching | \_2\_ | |
| 1. ***break*** statement | \_4\_ | |
| 1. Casting & Promotion | 2\_ | |
| 1. Classes vs Objects | \_3\_ | |
| 1. Class association (relationships) | \_2\_ | |
| 1. Command line arguments | \_3\_ | |
| 1. Compound operators (+=, -=, \*=, /=, %=) | \_3\_ | |
| 1. Concatenation operator | \_2\_ | |
| 1. Conditional operators | \_2\_ | |
| 1. Constants | \_2\_ | |
| 1. Constructors | \_2\_ | |
| 1. ***continue*** statement | \_2\_ | |
| 1. Control structures | \_2\_ | |
| 1. Data types | \_3\_ | |
| 1. Debugging | \_2\_ | |
| 1. Declaration (class) | \_3\_ | |
| 1. Declaration (method) | \_3\_ | |
| 1. Declaration (variable) | \_2\_ | |
| 1. Declaration (interface) | \_2\_ | |
| 1. ***do-while*** loops | \_2\_ | |
| 1. dot operator | \_2\_ | |
| 1. ***else*** statement | \_2 | |
| 1. Enhanced for loop (for each loop) | \_3\_ | |
| 1. Enumeration (Enum) | \_1\_ | |
| 1. Equality (equals method vs equals operator) | \_1\_ | |
| 1. Escape sequences | \_1\_ | |
| 1. Exceptions | \_2\_ | |
| 1. Fields | \_2\_ | |
| 1. ***final*** keyword | \_2\_ | |
| 1. floating-point values (***float*** vs ***double***) | \_2\_ | |
| 1. ***for*** loop | \_3\_ | |
| 1. Format specifiers | \_2\_ | |
| 1. Garbage collection | \_2\_ | |
| 1. Getters and Setters | \_2\_ | |
| 1. Guard conditions | \_2\_ | |
| 1. ***if*** statements | \_3\_ | |
| 1. ***import*** statement | \_3\_ | |
| 1. Increment/Decrement operators | \_3\_ | |
| 1. Initialization (variables) | \_3\_ | |
| 1. Integer overflow | \_2\_ | |
| 1. Integer values (***byte, short, int*** & ***long***) | \_2\_ | |
| 1. Integrated Development Environment (IDE) | \_2\_ | |
| 1. Jar Files (Java Archive Files) | \_1\_ | |
| 1. Java API (Application Programming Interface) | \_1\_ | |
| 1. Java program 5-step (phase) process | \_1\_ | |
| 1. Java Virtual Machine (JVM) | \_1\_ | |
| 1. JavaDoc (Java Documentation) | \_1\_ | |
| 1. Logical operators (&&, ||, !) | \_1\_ | |
| 1. Memory mapping (Heap vs Stack) | \_1\_ | |
| 1. Method arguments and parameters | \_2\_ | |
| 1. Method call | \_2\_ | |
| 1. Method ***main*** | \_2\_ | |
| 1. Method return values | \_2\_ | |
| 1. Multi-dimensional arrays | \_2\_ | |
| 1. Nested loops | \_2\_ | |
| 1. ***new*** keyword | \_2\_ | |
| 1. ***null*** keyword | \_3\_ | |
| 1. Object instantiation | \_2\_ | |
| 1. Operator precedence | \_2\_ | |
| 1. Overloading | \_2\_ | |
| 1. Overriding and @Override annotation | \_1\_ | |
| 1. ***package*** keyword | \_1\_ | |
| 1. Parameters vs. arguments | \_3\_ | |
| 1. Pass by reference vs Pass by value | \_2\_ | |
| 1. Pre/Post increment/decrement | \_2\_ | |
| 1. Precision | \_2\_ | |
| 1. Primitive Data Types | \_2\_ | |
| 1. Print statement methods (print, println, printf) | \_2\_ | |
| 1. Pseudocode | \_2\_ | |
| 1. Random numbers (Math vs Random class) | \_2\_ | |
| 1. Reference types | \_2\_ | |
| 1. Relational operators (==, !=, >, >=, <, <=) | \_2\_ | |
| 1. ***return*** statement | \_3\_ | |
| 1. Return values | \_3\_ | |
| 1. Scanner class | \_3\_ | |
| 1. Scope | \_3\_ | |
| 1. Stack trace | \_2\_ | |
| 1. ***static*** keyword (static class members) | \_1\_ | |
| 1. String class (immutable strings) | \_2\_ | |
| 1. ***switch/case*** statements | \_1\_ | |
| 1. Syntax / Syntax errors | \_1\_ | |
| 1. System.exit() method | \_1\_ | |
| 1. ***this*** reference keyword | \_2\_ | |
| 1. UML Activity Diagram | \_2\_ | |
| 1. Type conversion | \_2\_ | |
| 1. UML Class Diagram | \_2\_ | |
| 1. Varargs | \_2\_ | |
| 1. Variables (local –vs– instance –vs– static) | \_2\_ | |
| 1. ***void*** keyword | \_2\_ | |
| 1. ***while*** loops | \_2\_ | |