|  |
| --- |
| True / False |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. System software is used for tasks such as word processing, calculations and playing games.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *POINTS:* | 1 | | *REFERENCES:* | 2 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2. If an object’s methods are well written, the user is unaware of the low-level details of how the methods are executed, and the user must simply understand the interface or interaction between the method and the object.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *POINTS:* | 1 | | *REFERENCES:* | 9 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3. A Java identifier can contain only letters, digits, ampersands, or number signs.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *POINTS:* | 1 | | *REFERENCES:* | 15 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. You must use the Java interpreter to translate the bytecode into executable statements before running a Java application.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *POINTS:* | 1 | | *REFERENCES:* | 23 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5. When you issue the command to compile a class containing errors, the Java compiler does not produce any error messages. You will first need to run the class in order to see error messages.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *POINTS:* | 1 | | *REFERENCES:* | 23 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6. Logic errors are easily identified when a program is compiled.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *POINTS:* | 1 | | *REFERENCES:* | 31 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7. Although not a requirement, it is Java standard that class identifiers begin with an uppercase letter and use other uppercase letters to improve readability.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *POINTS:* | 1 | | *REFERENCES:* | 16 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8. A compiler ignores whitespace (that is, any combination of nonprinting characters) between words and lines.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *POINTS:* | 1 | | *REFERENCES:* | 18 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9. Dots (or periods) in a Java statement are used to separate the names of the components that make up the statement.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | True | | *POINTS:* | 1 | | *REFERENCES:* | 15 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10. Using the void keyword in the main() method header indicates that a value will be returned by the main() method is called.   |  |  |  | | --- | --- | --- | |  | a. | True | |  | b. | False |  |  |  | | --- | --- | | *ANSWER:* | False | | *POINTS:* | 1 | | *REFERENCES:* | 20 | |

|  |
| --- |
| Multiple Choice |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11. A program written in \_\_\_\_ is the most basic circuitry-level language.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | machine language | b. | Java | |  | c. | C | d. | BASIC |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 2 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12. Locating and repairing all syntax errors is part of the process of \_\_\_\_ a program.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | interpreting | b. | executing | |  | c. | debugging | d. | compiling |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 3 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13. \_\_\_\_ are also called modules, methods, functions, and subroutines. Java programmers most frequently use the term “method.”   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Calls | b. | Classes | |  | c. | Procedures | d. | Applications |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 6 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14. \_\_\_\_ refers to the hiding of data and methods within an object.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Instance | b. | Inheritance | |  | c. | Encapsulation | d. | Passed |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 9 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15. \_\_\_\_ describes the feature of languages that allows the same word to be interpreted correctly in different situations based on the context.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Polymorphism | b. | Insulation | |  | c. | Source code | d. | Architecturally neutral |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 10 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16. Programs that are embedded in a Web page are called Java \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | consoles | b. | applications | |  | c. | applets | d. | windowed applications |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 12 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 17. \_\_\_\_ are pieces of information that are sent into, or passed to, a method, usually because the method requires the information to perform its task or carry out its purpose.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Applets | b. | Objects | |  | c. | Methods | d. | Arguments |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 14 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 18. A(n) \_\_\_\_ defines the circumstances under which a class can be accessed and the other classes that have the right to use a class.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Pascal casing | b. | parameter | |  | c. | identifier | d. | access specifier |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 17 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 19. In Java, the reserved keyword \_\_\_\_ means that a method is accessible and usable even though no objects of the class exist.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | active | b. | undefined | |  | c. | void | d. | static |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 20 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20. Line comments start with \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | a forward slash and an asterisk | b. | a percent sign | |  | c. | two forward slashes | d. | a forward slash and two asterisks |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 33 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 21. In a \_\_\_\_ environment, you can change directories using the cd command. For example, to change to a directory named MyClasses, you type cd MyClasses and press Enter.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | DOS | b. | Java | |  | c. | Windows | d. | graphical |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 24 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22. After a successful compile, you can run the class file on any computer that has a \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Java language interpreter | b. | DOS | |  | c. | Web browser | d. | text editor |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 24 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 23. Java contains a class named \_\_\_\_ that allows you to produce dialog boxes.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | JBox | b. | JGUI | |  | c. | JOptionPane | d. | JDialog |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 35 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 24. Whenever a method requires multiple arguments, the arguments are always separated with \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | forward slashes | b. | periods | |  | c. | brackets | d. | commas |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 36 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 25. If a compiler detects a violation of language rules, it refuses to translate the class to \_\_\_\_.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | machine code | b. | a logic error | |  | c. | an application | d. | Java |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 25 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 26. \_\_\_\_ is the process the compiler uses to divide your source code into meaningful portions; the message means that the compiler was in the process of analyzing the code when the end of the file was encountered prematurely.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Parsing | b. | Compiling | |  | c. | Running | d. | Debugging |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 28-29 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 27. A **\_\_\_\_** is an error not detected until the program asks the computer to do something wrong, or even illegal, while executing.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | syntax error | b. | package | |  | c. | run-time error | d. | Java API |  |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | | *REFERENCES:* | 32 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 28. It is best to use the \_\_\_\_ available text editor when writing Java programs.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | simplest | b. | most complex | |  | c. | latest | d. | best |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 22 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 29. It is a tradition among programmers that the first program you write in any language produces “\_\_\_\_” as its output.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | Hello, your name! | b. | Hi, your name! | |  | c. | My first program! | d. | Hello, world! |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 22 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30. When you run a Java application using the \_\_\_\_ command, do not add the .class extension to the filename.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | static | b. | class | |  | c. | output | d. | java |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 30 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31. If you receive an error that states, “Exception in thread ‘main’ java.lang.NoClassDefFoundError,” when you try to execute the application, you probably do not have your \_\_\_\_ set correctly.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | object | b. | class length | |  | c. | java | d. | class path |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 35 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 32. {    public static void main(String[] args)    {       System.out.println(“First Java application”);    } }  Given the above code, which item identifies the access specifier?   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | static | b. | public | |  | c. | void | d. | First |  |  |  | | --- | --- | | *ANSWER:* | b | | *POINTS:* | 1 | | *REFERENCES:* | 17-18 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 33. public class First {    public static void main(String[] args)    {       System.out.println(“First Java application”);    } }  Given the above code, which item identifies the name of the class?   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | First | b. | void | |  | c. | public | d. | static |  |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | | *REFERENCES:* | 18 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 34.    public class First     {       public static void main(String[] args)       {          System.out.println(“First Java application”);       }     }  ​  Given the above code, which item identifies the method’s return type?   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | println | b. | void | |  | c. | public | d. | static |  |  |  | | --- | --- | | *ANSWER:* | b | | *POINTS:* | 1 | | *REFERENCES:* | 19 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 35.    public class First    {      public static void main(String[] args)      {         System.out.println(“First Java application”);      }    }  Given the above code, which item identifies that the method will work without instantiating an object of the class?   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | a. | println | b. | public | |  | c. | void | d. | static |  |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | | *REFERENCES:* | 19 | |

|  |
| --- |
| Completion |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 36. In programming, named computer memory locations are called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ because they hold values that might vary.   |  |  | | --- | --- | | *ANSWER:* | variables | | *POINTS:* | 1 | | *REFERENCES:* | 6 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 37. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** comments are a special case of block comments that are used to generate documentation.   |  |  | | --- | --- | | *ANSWER:* | Javadoc | | *POINTS:* | 1 | | *REFERENCES:* | 33 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 38. You use an import statement when you want to access a built-in Java class that is contained in a group of classes called a(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.   |  |  | | --- | --- | | *ANSWER:* | package | | *POINTS:* | 1 | | *REFERENCES:* | 36 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39. A(n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ error occurs when the syntax of the program is correct and the program compiles but produces incorrect results when you execute it.   |  |  | | --- | --- | | *ANSWER:* | logic | | *POINTS:* | 1 | | *REFERENCES:* | 31 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 40. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is also called the Java class library; it contains information about how to use every prewritten Java class, including lists of all the methods you can use with the classes.   |  |  | | --- | --- | | *ANSWER:* | Java API | | *POINTS:* | 1 | | *REFERENCES:* | 38 | |

|  |
| --- |
| Matching |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Match each term with the correct statement below.*   |  |  | | --- | --- | | a. | literal string | | b. | high-level programming languages | | c. | dialog box | | d. | syntax | | e. | whitespace | | f. | compiler | | g. | attributes | | h. | Machine language | | i. | Java virtual machine (JVM) |  |  |  | | --- | --- | | *REFERENCES:* | 14 2 35 2 18 3 7 2 11 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 41. Allow you to assign intuitive names to areas of computer memory   |  |  | | --- | --- | | *ANSWER:* | b | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 42. Any combination of nonprinting characters   |  |  | | --- | --- | | *ANSWER:* | e | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 43. The characteristics that define an object   |  |  | | --- | --- | | *ANSWER:* | g | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 44. A low-level programming language   |  |  | | --- | --- | | *ANSWER:* | h | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 45. Translates an entire program before carrying out the statement   |  |  | | --- | --- | | *ANSWER:* | f | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 46. The rules for the programming language   |  |  | | --- | --- | | *ANSWER:* | d | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 47. Hypothetical computer used to run a Java program   |  |  | | --- | --- | | *ANSWER:* | i | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 48. Series of characters that will appear in output exactly as entered   |  |  | | --- | --- | | *ANSWER:* | a | | *POINTS:* | 1 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 49. A GUI object resembling a window in which you can place messages you want to display   |  |  | | --- | --- | | *ANSWER:* | c | | *POINTS:* | 1 | |

|  |
| --- |
| Subjective Short Answer |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 50. Why is creating a GUI environment for users a natural use for object orientation?   |  |  | | --- | --- | | *ANSWER:* | Creating a GUI environment for users also is a natural use for object orientation. It is easy to think of the components a user manipulates on a computer screen, such as buttons and scroll bars, as similar to real-world objects. Each GUI object contains data—for example, a button on a screen has a specific size and color. Each object also contains behaviors—for example, each button can be clicked and reacts in a specific way when clicked. Some people consider the term object-oriented programming to be synonymous with GUI programming, but object-oriented programming means more. | | *POINTS:* | 1 | | *REFERENCES:* | 7 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 51. What are the two main types of Java programs?   |  |  | | --- | --- | | *ANSWER:* | You can write two kinds of programs using Java:   * Applets are programs that are embedded in a Web page. * Java applications are stand-alone programs. Java applications can be further subdivided into console applications, which support character or text output to a computer screen, and windowed applications, which create a GUI with elements such as menus, toolbars, and dialog boxes. Console applications are the easier applications to create. | | *POINTS:* | 1 | | *REFERENCES:* | 12 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 52. Describe the components (objects and classes) that are used in the statement System.out.println().   |  |  | | --- | --- | | *ANSWER:* | Within the statement System.out.println("First Java application");, the method to which you are passing "First Java application" is named println(). The Java methods println() and print() both produce output. With println(), after the output is displayed, the insertion point moves to the following line so that subsequent output appears on a new line. With print(), however, the insertion point does not advance to a new line, so subsequent output appears at the end of the current line.  Within the statement System.out.println("First Java application");, out is an object that is a property of the System class that refers to the standard output device for a system, normally the monitor. The out object itself is an instance of the PrintStream class, which contains several methods, including println().  Within the statement System.out.println("First Java application");, System is a class. Therefore, System defines attributes for System objects, just as the Dog class defines the attributes for Dog objects. One of the System attributes is out. The dots (periods) in System.out.println() are used to separate the names of the components in the statement. | | *POINTS:* | 1 | | *REFERENCES:* | 14-15 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 53. When you define a Java class using an identifier, what are the requirements you need to know?   |  |  | | --- | --- | | *ANSWER:* | You can define a Java class using any name or identifier you need, as long as it meets the following requirements:   * A Java identifier must begin with a letter of the English alphabet, a non-English letter (such as α or π), an underscore, or a dollar sign. A class name cannot begin with a digit. * A Java identifier can contain only letters, digits, underscores, or dollar signs. * A Java identifier cannot be a reserved keyword, such as public or class. * A Java identifier cannot be one of the following values: true, false, or null. These are not keywords (they are primitive values), but they are reserved and cannot be used. | | *POINTS:* | 1 | | *REFERENCES:* | 15-16 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 54. Regarding code layout, write the following Java code using a common alternate placement of the first curly brace:  ​  public static void main(String[] args)  {  System.out.println("First Java application");  }   |  |  | | --- | --- | | *ANSWER:* | public static void main(String[] args) {  System.out.println("First Java application");  } | | *POINTS:* | 1 | | *REFERENCES:* | 18 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 55. Why is it important for programmers to use program comments?   |  |  | | --- | --- | | *ANSWER:* | Programmers use comments to leave notes for themselves and for others who might read their programs in the future. At the very least, your Java class files should include comments indicating the author, the date, and the class name or function. The best practice dictates that you also include a brief comment to describe the purpose of each method you create within a class. | | *POINTS:* | 1 | | *REFERENCES:* | 33 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 56. What are some of the reasons the javac command might not be recognized?   |  |  | | --- | --- | | *ANSWER:* | You made a typo error when typing the command javac.  You misspelled the filename.  You are not within the correct subfolder or subdirectory on your command line.  Java was not installed properly. | | *POINTS:* | 1 | | *REFERENCES:* | 24 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 57. What happens after a programmer successfully compiles a Java program named “First.java”?   |  |  | | --- | --- | | *ANSWER:* | If you receive no error messages after compiling the code in a file named First.java, the application compiled successfully. In that case, a file named First.class is created and saved in the same folder as the text file that holds the source code. After a successful compile, you can execute the program (run the class file) on any computer that has a Java language interpreter. | | *POINTS:* | 1 | | *REFERENCES:* | 24 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 58. If you modify a class, what are the steps you must take to see your changes in the executing program?   |  |  | | --- | --- | | *ANSWER:* | 1. Save the file with the changes (using the same filename).  2. Recompile the class with the javac command. 3. Interpret the class bytecode and execute the class using the java command. | | *POINTS:* | 1 | | *REFERENCES:* | 31 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 59. Provide examples of logic errors. How do programmers minimize logic errors in their code?   |  |  | | --- | --- | | *ANSWER:* | Examples of logic errors include multiplying two values when you meant to add, printing one copy of a report when you meant to print five, or forgetting to produce a total at the end of a business report when a user has requested one. Errors of this type must be detected by carefully examining the program output. It is the responsibility of the program author to test programs and find any logic errors. Good programming practices can help to minimize errors. | | *POINTS:* | 1 | | *REFERENCES:* | 31 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 60. Identify why a class named first would be different from a class named FIRST. Explain why it is important to be aware of uppercase and lowercase values when writing Java programs?   |  |  | | --- | --- | | *ANSWER:* | Java is case sensitive. Programmers must pay close attention to correct uppercase and lowercase values in Java programs. Class names with different uppercase or lowercase values constitute completely different classes. | | *POINTS:* | 1 | | *REFERENCES:* | 15 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 61. Explain the purpose of arguments and why it is necessary to pass arguments to a method.   |  |  | | --- | --- | | *ANSWER:* | Arguments are pieces of information that are sent to a method. The act of sending arguments to a method is called passing arguments to the method. You pass methods to arguments so they know what information to work with. | | *POINTS:* | 1 | | *REFERENCES:* | 14 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 62. In Java, whitespace is optional but useful. Describe why and how whitespace can be used in coding?   |  |  | | --- | --- | | *ANSWER:* | Whitespace is any combination of nonprinting characters. You use whitespace to organize your program code and make it easier to read. You can insert whitespace between words or lines in your program code by typing spaces, tabs, or blank lines because the compiler ignores these extra spaces. However, you cannot use whitespace within an identifier or keyword, or surrounding the dots in any class-object-method combination. | | *POINTS:* | 1 | | *REFERENCES:* | 18 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 63. Briefly describe the elements of the main() method:  public static void main(String[] args)  {  Java code...  }   |  |  | | --- | --- | | *ANSWER:* | ​  In the method header public static void main(String[] args), the word public is an access specifier, just as it is when you use it to define the First class.  In Java, the reserved keyword static means that a method is accessible and usable even though no objects of the class exist.  The keyword void used in the main() method header indicates that the main() method does not return any value when it is called.  The name of the method is main().  In the method header public static void main(String[] args), the contents between the parentheses, String[] args, represent the type of argument that can be passed to the main() method. | | *POINTS:* | 1 | | *REFERENCES:* | 19-20 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 64. Once an application is written and saved, the Java class must be compiled. Describe what two procedures must occur in order to view the output of the application.   |  |  | | --- | --- | | *ANSWER:* | ​  1. You must compile the class you wrote (called the source code) into bytecode. 2. You must use the Java interpreter to translate the bytecode into executable statements.  ​ | | *POINTS:* | 1 | | *REFERENCES:* | 23 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 65. public class Hello {    public static void main(String[] args)    {        \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    } {  Using the given code, write the statement in the main() method body that will produce the output “Hello to all!”.   |  |  | | --- | --- | | *ANSWER:* | System.out.println(“Hello to all!”);  Everything between the curly braces makes up the method body. | | *POINTS:* | 1 | | *REFERENCES:* | 23 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 66. public class FindMyErrors {    public static void main(String[] args)    {        System.out.println(“My application with errors)     }   Given the above code, identify three separate syntax errors.   |  |  | | --- | --- | | *ANSWER:* | There is a semicolon missing at the end of the println statement that will produce the output. There is a missing curly brace. Curly braces must be open and closing pairs. There are missing quotation marks in the println statement that will produce the output. | | *POINTS:* | 1 | | *REFERENCES:* | 24-27 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 67. Write a line comment and block comment for the following line of code identifying the author, date, and purpose of the method. In addition, explain the difference between comments that consist of a double-slash (//) and those that have a starting and ending slash asterisk (/\* .. \*/).  ​  System.out.println(“Hello Students”);   |  |  | | --- | --- | | *ANSWER:* | Possible comments include:  ​  // Written by <your name>  // Written on <date>  ​  /\* The println method will output “Hello Students” \*/  ​  // - used for line comments  /\* - used for block comments \*/ | | *POINTS:* | 1 | | *REFERENCES:* | 32-34 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 68. System.out.println(“First Java application”);  ​  Given the above code, identify and describe the use of a literal string and the use of parentheses.   |  |  | | --- | --- | | *ANSWER:* | A literal string is a series of characters that will appear exactly as entered. Any literal string in Java is written between double quotation marks. The string “First Java application” appears within parentheses because the string is an argument to a method, and arguments to methods always appear within parentheses. | | *POINTS:* | 1 | | *REFERENCES:* | 14 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 69. JOptionPane.showMessageDialog(null, “Show my message”);  ​  The showMessageDialog() method above requires two arguments. What are the two arguments and what is the purpose of each argument?   |  |  | | --- | --- | | *ANSWER:* | 1. null - message box will be placed in the center of the screen  2. string to be displayed - in this case it is “Show my message”. This string will be displayed in the message box. | | *POINTS:* | 1 | | *REFERENCES:* | 36 | |