

# Chapter 1: Java Basics

1. D. An entry point in a Java application consists of a `main()` method with a single `String[]` argument, return type of `void`, and modifiers `public` and `static`. The name of the variable in the input argument does not matter. Option A is missing the `static` modifier, Option B is missing the `String[]` argument, and Option C has the wrong access modifier and method name. Only Option D fulfills these requirements. Note that the modifier `final` is optional and may be added to an entry point method.
2. A. The diagram is an example of object-oriented design in Java, making Option B a true statement. Options C and D are also true, as they follow from the inheritance model in the diagram. Option A is the correct answer, since platform independence has nothing to do with the diagram.
3. C. The proper extension for a Java compiled bytecode file is `.class`, making Option C the correct answer.
4. B. The fact that the `Date` class exists in both packages does not impact the ability to import both packages, so lines 1 and 2 compile without issue, and Option A is

12. A. The code does not compile because of line 5, making Option A the correct answer. For this question, it helps to understand variable scope. The `main()` method is `static` and does not have access to any class instance variables. The `birds` variable is not `static` and requires a class instance variable to access. Therefore, the code does not compile when the `static` method attempts to access a non-`static` variable without an instance of the class.
13. D. The `java` command can only execute compiled `.class` files, so I is false. Java is most certainly object oriented, one of the key design principles, so II is also false. The `javac` command compiles into bytecode, which must be run in a Java virtual machine (JVM), and is not native machine code, so III is false as well. Since none of the statements are true, Option D is the correct answer.
14. D. A class can start with a comment, an optional `package` statement, or an `import` statement if there is no `package` statement. It cannot start with a variable definition, making Option D the correct answer.
15. C. Classes may be defined without a package declaration and are placed in the default package, so Option A is incorrect. Option B is a completely false statement as no such file is required in Java. Option C is correct as it is one of the primary reasons for organizing your application into packages. Option D is incorrect as package-private allows access to methods and variables to be limited to those classes within the same package.
16. B. The compilation command requires the full or relative name of the file, including the `.java` extension, making Options A and C incorrect. The execution command requires the class name without a filename extension, making Option D incorrect. Option B is the only correct set of compilation and execution commands.
17. D. Encapsulation is the technique of removing access to a class's instance variables from processes outside the class, making Option D the correct answer.
18. D. The `height` variable is declared within the if-then statement block. Therefore, it cannot be referenced outside the if-then statement and the code does not compile.
19. A. A Java bytecode file is a binary encoded set of instructions designed to be run on any computer with a compatible JVM, making Option A the only correct answer. By compatible JVM, we mean one capable of running the class file. For example, a Java 6 JVM may have trouble executing a Java 8 compiled file. Option B is incorrect, and is more a facet of machine language compiled programming classes. Option C is also incorrect as binary data is not particularly human readable. Finally, Option D is incorrect as the compiled file can be distributed without its `.java` source file and execute without issue.
20. D. Unlike with some other programming languages, the proper way to terminate a line of code is with a semicolon (`;`), making D the only correct answer.
21. C. The code compiles and runs without issue, so Options A and B are incorrect. The

question relies on your ability to understand variable scope. The variable `today` has local scope to the method in which it is executed. The variable `tomorrow` is re-declared in the method, but the reference used on line 7 is to the instance variable with a value of 10. Finally, the variable `tomorrow` is `static`. While using an instance reference to access a `static` variable is not recommended, it does not prevent the variable from being read. The result is line 7 evaluates and prints  $(20 + 10 + 1) = 31$ , making C the correct answer.

- 22. C. Line 1 is missing the `class` keyword. Line 2 contains two types for the same variable. Line 3 is a valid definition for a method, making C the correct answer. Finally, line 4 contains an access modifier, `private`, after the return type, which is not allowed. In addition, `void` is an invalid type for variables.
- 23. D. Platform independence is the property of Java that allows it to be run on a variety of different devices.
- 24. A. Options B, C, and D are each correct statements about JVMs. Option A is incorrect. Not only is it not a statement about JVMs, it is actually false as Java bytecode can often be easily decoded/decompiled.
- 25. B. There is no such thing as package variables, so Option A is incorrect. Option C is incorrect as the variable is only in scope within a specific instance of the class. Option D is also incorrect as the variable is only in scope for a single method that it is defined in. Option B is the only correct answer as class variables are in scope within the program.
- 26. C. Option A is incorrect as the sub-package `recurring` is not included by the `import` statements. Option B is also incorrect as it uses the plural `directors` instead of the singular `director` used in the `import` statements. Option D is incorrect as the wildcard is applied to the sub-package `movie.director`, not the package `movie`. Finally, Option C is correct as it is a valid class accessible from the wildcard import.
- 27. D. Java classes are defined in this order: `package` statement, `import` statements, class declaration, making Option D the only correct answer. Note that not all of these statements are required. For example, a class may not have a `package` statement, but if it does, it must come first in the file.
- 28. D. The `import` statements for `stars.*` and `stars.Blackhole` are redundant `import` statements, since only the class `Blackhole` is used, and therefore one of them can be safely removed. The `import` statements `java.lang.*` and `java.lang.Object` are both not required as `java.lang` is automatically imported in every Java class. Therefore, three of the four `import` statements can be safely removed, making the correct answer Option D.
- 29. C. The application prints the third argument of the input methods. Note that double quotes `""` group input arguments. Therefore, the third argument of Option A is `White-tailed deer`. The third argument of Option B is `3`. The third argument of Option C is `White-tailed`, making it the correct answer. Finally, Option D only has two input

arguments, leading to an `ArrayIndexOutOfBoundsException` trying to read the third argument at runtime.

10. B. The `javac` command compiles a `.java` file into a `.class` bytecode file, making Option B the correct answer.
11. B. Java is object oriented, not procedural, so Option A is a false statement. Java allows method overloading in subclasses, so Option B is correct. Operator overloading is permitted in languages like C++, not Java, so Option C is also untrue. Finally, Option D is not a true statement as the JVM manages the location of objects in memory that can change and is transparent to the Java application.
12. D. Option A is incorrect as the return type of the method cannot be `null`. Option B is also incorrect as the return type cannot be `void` if the method uses a `return` statement. Option C is incorrect too as the `class` keyword is replaced with `int`. Option D is correct because it's the only answer that allows the code to compile without issue. Note that other values are possible for this question. For example, either `int` or `long` can be entered in the last blank. The key here is that only one of the available answer choices allows the code to compile.
13. A. The code compiles so Option D is incorrect. The input to the constructor is ignored, making the assignment of `end` to be 4. Since `start` is 2, the subtraction of 4 by 2 results in the application printing 2, followed by 5, making Option A the correct answer.
14. D. Option A is a false statement, while Options B and C are actually arguments against using inheritance. Option D is one of the most important reasons Java supports inheritance, to allow increased code reuse among classes.
15. A. The double slash (`//`) syntax can have any number of slashes as a comment, so long as it starts with two of them, making Option A the correct answer. The (`#`) is not a comment character in Java, regardless of whether it is followed by a (`!`), so Option B is incorrect. Option C is incorrect as a single slash (`/`) is not a valid comment in Java. Finally, Option D is incorrect as Option A is a valid comment.
16. B. An entry point in a Java application consists of a `main()` method with a single `String[]` argument, return type of `void`, and modifiers `public` and `static`. Option D is the typical syntax for this method, although Options A and C are also valid forms of this method. Note that the modifier `final` is optional and may be added to the method signature. Furthermore, the `main()` method may take a vararg or array. Option B is the only invalid declaration as it does not take an array as an argument.
17. B. The line of code cannot be inserted at `a1` because no variables are allowed outside of the class declaration in this file, making Options A and D incorrect. The line of code can also not be inserted at `a3` as local variables defined within methods cannot have access modifiers such as `public`, making Option C incorrect. The code can be inserted independently at `a2` and `a4` as instance variables can be defined anywhere in the class outside a method. Therefore, Option B is the correct choice.

8. A. Option A is the only correct answer as a `class` definition is the only required component in a Java class file. Note that we said a Java class file here; Java also allows interfaces and enums to be defined in a file. A `package` statement and `import` statements are optional for declaring a class, making Options B and C incorrect. A class may also be defined with package-level access in a file, making Option D an incorrect answer.
9. D. The proper extension for a Java compiled bytecode file is `.java`, making Option D the correct answer.
10. C. Remember that `java.lang` is automatically imported in all Java classes, therefore both `java.lang.Math` and `pocket.complex.Math` are both imported into this class. Importing both sets of packages does not cause any compilation issues, making Option A incorrect. Line 3 is unnecessary import but including it does not prevent the class from compiling, making Option B incorrect. While both versions of `Math` may be imported into the class, the usage of the `Math` class requires a package name. Because of this, line 6 does not compile as the class reference is ambiguous, making Option C the correct answer and Option D incorrect.
11. A. Options B and C are accessible within the class as they are covered by the `import` statements. Option D is also fine as `java.lang.Object` is available without an explicit import. The only class not automatically accessible within the class without the full package name is `dog.puppy.female.KC` as the `import` statements do not include sub-packages; therefore, Option A is the correct answer.
12. B. Object-oriented programming is the technique of structuring data into objects, which may contain data and a set of actions that operate on the data, making Option B the correct answer.
13. A. All of the `import` statements in this class are required. Removing any of them would cause the class to not compile, making Option A the correct answer.
14. C. The `numLock` variable is not accessible in the `static main()` method without an instance of the `Keyboard` class; therefore, the code does not compile, and Option C is the correct answer.
15. D. The code compiles and runs without issue, so Option A is incorrect. The question involves understanding the value and scope of each variable at the `print()` statement. The variables `feet` and `tracks` are locally scoped and set to 4 and 15, respectively, ignoring the value of `tracks` of 5 in the instance of the class. Finally, the `static` variable `s.wheels` has a value of 1. The result is the combined value is 20, making Option D the correct answer.
16. B. First off, the `color` variable defined in the instance and set to `red` is ignored in the method `printColor()` as local scope overrides instance scope, so Option A is incorrect. The value of `color` passed to the `printColor()` method is `blue`, but that is lost by the assignment to `purple`, making Option B the correct answer and Option C incorrect. Option D is incorrect as the code compiles and runs without issue.



47. C. The `javac` command takes a text-based `.java` file and returns a binary bytecode `.class` file, making II a true statement. The `java` command uses a period (`.`) to separate packages, not a slash (`/`), making I a true statement and III a false statement. For these reasons, Option C is the correct answer.
48. D. The application compiles without issue, so Option C is incorrect. The application does not execute though, as the `main()` method does not have the correct method signature. It is missing the required input argument, an array of `String`. Trying to execute the application without a proper entry point produces an error, making Option D the correct answer.
49. C. Option A does not compile because it is missing the closing bracket for the class. Option D does also not compile as `void` is not a valid type for a variable. Regardless, Options A and D are incorrect as they are missing the `getRating()` method. Note that Option A also uses an abbreviation for `numberOfPages`. Option B is incorrect as it is missing the `numberOfPages` attribute. Option C is the correct answer as it properly defines the attribute `numberOfPages` and method `getRating()`.
50. C. Garbage collection can happen at any time while an application is running, especially if the available memory suddenly becomes low, making Option A incorrect. Option B is also incorrect, since it is trivial to create a Java application with an infinite loop that never terminates. Option D is incorrect because the computer must be able to run the JVM in order to execute a Java class. Option C is the only correct answer, as the JVM does require an entry point method to begin executing the application.