
This assignment contains 4 pages (including this page) and 33 questions. Check to see if any pages are missing.

This week's notes are based on the in-class session

1. What is a method?

2. Why are all functions called methods in Java? (probably all are true, but which one is objectively accurate)
 - Because Java is weird
 - It's just a style thing
 - In Java all functions must be in a method
 - To make Java look different than C++
3. What are three advantages of using methods?
 -
 -
 -
4. How is a static method called?

5. Give an example of how `parseInt` in class `Integer` would be called on a string called `number`.
6. How is a non-static method called?
7. Give an example of how `length` in class `String` would be called on a string called `number`.
8. When a static method is called *from within its own class* how can it be called?



9. Does the method alluded to in the previous question also apply to non-static methods?
10. What is the syntax for creating a method?
11. Write a method named `printBar`. It takes no arguments and has no return value. It only prints `=====`.
12. Write a method named `greeting`. It takes a string argument and has no return value. It only prints `=====`.
13. Write a method named `square`. It takes a double as an argument and returns a double. It only prints `=====`.
14. What is the difference between a parameter and an argument?
hint: It might help if you think of "parameter" as "formal parameter."



15. What is the difference between public and private?
16. So far, all of our course we have only had a single class in our programs. Does the difference between public and private matter right now?
17. What is the return type of a method that does not return a value?
18. How is a method that returns a value called differently than a function that does not return a value?
19. Does a method that calls a function have to use the return value, or may it be used like a regular statement if we want to ignore the return value? (Try it and find out if you need to)
20. What is "Call by value?" Why is it also called "Call by copy?"
21. What is a "Call by reference?"
22. "Java only has call by value"
 - Absolutely True with no fine print
 - False
 - Technically && Absolutely True, but with a huge loophole in the fine print.
23. What is the "method signature?"
24. Are the formal parameters part of the method signature?
25. Is the return type part of the method signature
26. What is "method overloading?"
27. What is the relationship between the method signature and method overloading?



28. Can a method be overloaded if the only difference is the return type? Explain why (which should be pretty obvious after the last three questions)
29. How does scope apply to methods?
30. What are "local variables?"
31. Can two different methods have the same names as their formal parameters?
32. Can a method have a parameter with the same name as one used in the `main()` method?
33. Hey, wait a minute. Is `main(String[] args)` a method?