Midterm Exam Review

1. Be familiar with the following terms and what they mean in relation to Java:
    a. JVM
    b. bytecode
    c. main
    d. void
    e. constructor
    f. default constructor
    g. import
    h. primitive variable
    i. reference variable
    j. local variable
    k. instance
    l. Object
    m. extends
    n. class
    o. method
    p. superclass
    q. subclass
    r. toString
    s. overloading
    t. override
    u. final
    v. static
    w. Just-In-Time compilation (JIT)

2. Know the structure of a basic java project folder/package, and file types
3. Know what it means to instantiate an object and how to do that. Know the difference between a variable and an object.
4. What are the two main categories of variable types (NOT scopes) in Java?
5. Know the conventions for naming classes and variables/methods in Java.
6. What are the primitive types? (Be able to identify them and/or give examples.)
7. Understand how automatic (implicit) casting can be done between primitive types. Know how to use explicit casting, when it is necessary, and how to ensure it is safe.
8. Know the meanings of public, protected, and private and how they compare with each other.
9. Know how variables are initialized by default (i.e. when they're local variables vs. instance variables).
10. How to generate random numbers.

11. What is the difference between passing a primitive-type array (its element, respectively) and an object-type array (its element object, respectively) to a method, and how may they affect the caller? How to initialize arrays?
12. What is method overloading and how it is accomplished?
13. Understand the Java class hierarchy including knowing what is the superclass of all classes. Be able to give a method available via that superclass?
14. Understand what is meant by an "is-a" relationship and a "has-a" relationship.
15. Understand variable scopes, including those for local and instance variables, and shadowing.
16. Understand how static variables and methods are different from non-static and how they can be accessed/called.
17. Be able to write basic Java code, including a complete simple program, basic methods and classes. This includes knowing the complete signature of the first method executed by the JVM when running a Java application.
18. Know how to print text to the console from a Java program, using both "println" and "printf".
19. Know how to compile and run a simple Java program, including the correct syntax for the necessary commands.
20. Be able to debug basic Java code which contains errors.
21. Understand inheritance and how to write classes which are sub-classes of other classes. Know which members of parent classes are available to children classes and how to access them.
22. Understand how constructors work, especially in relation to super-classes, and what default constructors are and when they exist.
23. Know how to call the methods of a super-class from a sub-class.
24. Know the access scope of public, protected, and private members of superclass.
25. Know why using private type variables in superclass could benefit software.
26. Know how and why to use the @Override annotation.
27. Know how to use exception handling, including how to write a try-catch block.
28. Know how to declare and utilize arrays and ArrayLists, including basic looping techniques, especially iterators. Know the built in variable for arrays used to access their size.
29. Know how to get user input (know at least one method)
30. Know how to use If, If-Else, For, While, Do-While, Break, Continue
31. Review lecture notes and coding examples for lectures related to the above topics.
32. Review all homework assignments. Make sure you completely understand and can solve all portions of each (at least HW3-HW5).