Study guide for CS163/CS164 final exam

You should also be familiar with the material from the first and second midterm study guides below. You should be able to interpret or write Java code that uses any of the items shown below:

1) Abstract Classes
    a. Abstract Methods
    b. Concrete Methods
    c. and difference between them
    d. Terminology
        i. *Superclass* versus *Subclass*
        ii. *Base* class versus *Derived* class
        iii. *Parent* class versus *Child* class
    e. Sharing code via abstract class
    f. Differentiating code via abstract class
    g. **extends** keyword
    h. **super** keyword
2) Interfaces
    a. Only contains *abstract* methods
    b. Cannot contain *concrete* methods
    c. **implements** keyword
    d. Methods only, no class or instance data
    e. Constants sometimes included
    f. Enumerations often included
    g. Pure functionality (no implementation)
    h. Differences between interface and abstract class
3) Recursion
    a. Basic definition of recursion
    b. Calling a method from itself
    c. Stack model for recursion
    d. Helper methods
    e. Tail recursion
4) Collections
    a. Collection Framework Hierarchy
    b. **Collection** interface
    c. **List** interface
    d. **ArrayList** class
    e. Difference with **LinkedList**
    f. **Comparable** interface
    g. **Comparator** interface
    h. **Collections** class
        i. Searching a collection
        ii. Sorting a collection
        iii. Shuffling a collection
        iv. Copying a collection
        v. etc.
5) Sorting
    a. Algorithms for sorting
        i. **Insertion** sort
        ii. **Selection** sort
        iii. **Bubble** sort
    b. Computational Complexity
        i. Big O Notation
        ii. Linear complexity
        iii. Logarithmic complexity
        iv. Quadratic complexity
        v. O(nlog(n)), for example **MergeSort**
        vi. etc.

Study guide for CS163/CS164 second midterm exam

You should also be familiar with the material from the first midterm study guide. You should be able to interpret or write Java code that uses any of the items shown below. These same topics will be covered on the programming quiz.

1) Methods
    a. Declaring (defining) methods
    b. Invoking (calling) methods
    c. Argument values and types
    d. Return values and types, including **void**
    e. Pass by reference vs. pass by value
    f. Method overloading
    g. Local variables (not initialized)
    h. Scope of locals
2) Single-Dimensional Arrays
    a. Declaring, allocating, initializing arrays
    b. Array indexing, array access
    c. Array **length**
    d. Iterating arrays
    e. Arrays as method parameters
    f. Arrays as return values
    g. Linear search, Binary search
    h. Selection sorting
    i. Arrays class: **toString(), sort(), equals()**
    j. Command line arguments: **main (String[] args)**
3) Multi-Dimensional Arrays
    a. Declaring, allocating, initializing
    b. Array indexing, array access
    c. Array **array.length, array[i].length**
    d. Iterating arrays
    e. Arrays as method parameters
    f. Arrays as return values
4) Classes
    a. Classes versus Objects
    b. Class variables (**static**)
    c. Instance variable (non-static)
    d. **.** operator for data and method access
    e. Class and Instance methods
    f. Scoping of variables
    g. Instantiating an **object** from a class
    h. Class constructors
    i. Object references
    j. **public** versus **private** data and methods
    k. Getter and Setter methods
    l. Arrays of objects
    m. **this** keyword
5) Exceptions
    a. **try/catch** blocks
    b. **throw** keyword
    c. **finally** clause
    d. Checked versus unchecked exceptions
6) File Input/Output
    a. **File** objects
    b. **Scanner** for file input
    c. **PrintWriter** for file output

Study guide for CS163/CS164 first midterm exam

By now, you should be able to interpret or write a Java program that uses any of the items shown below.

1) Java programs
    a. Writing a **class** with **main** method
    b. Importing packages
    c. Declaring and initializing variables
    d. Assignment statements
    e. Numeric, character, and string literals
2) Data types (and size in bits)
    a. **byte (8), short (16), int (32), long (64)**
    b. **float (32), double (64)**
    c. **boolean, char (16), String**
3) Expressions
    a. Primitive operators: *****, /, +, **-**. **%**, ++, **--**
    b. Integer versus floating point math
    c. Relational operators: ==,!=,<,<=,>,>=
    d. Boolean operators: &&, ||, ^, !
    e. Order of operations, parentheses
    f. Mixed types and type casting
4) String functions
    a. **length, charAt, indexOf, substring**
    b. String concatenation: **concat**, +
    c. String comparison: **equals (**== doesn't work!)
5) Writing to the console
    a. Using **System.out.print/println/printf**
        i. Differences between them
        ii. Formatters: %f,%d,%c,%s
        iii. Special Characters: '\n','\t'
    b. Combining literals and variables
6) Reading from the console
    a. Declaring and use a **Scanner**
    b. Reading strings that are single tokens: **next**
    c. Reading strings that are lines of text: **nextLine**
    d. Reading integers and doubles: **nextInt, nextDouble**
7) Conditionals
    a. **if**, **if-else**, and **else** statements
    b. **switch** statements
8) Wrapper Classes
    a. Integer: **parseInt()**
    b. Double: **parseDouble()**
    c. Character: **isUpperCase(), isDigit(), isLetter()**
9) Math
    a. Math constants: **PI**
    b. Math methods:
        - **sqrt()**
        - **sin(), cos()**
        - **pow(), exp(), log()**
        - **min(), max()**
        - **round(), floor(), ceil()**
10) Control Loops
    a. **for** loops
    b. **while** loops
    c. **do**/**while** loops
    d. **for each** loops
    e. **break** keyword
    f. **continue** keyword