

## Study guide for CS165 Review 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**
  - b. Combining literals and variables
- 6) Reading from the console
  - a. Declaring and use a **Scanner**
  - b. Methods: **next**, **nextLine**, **nextInt**, **nextDouble**
- 7) Conditionals
  - a. **if**, **if-else**, **else** and **switch** statements
- 8) Wrapper Classes
  - a. Integer: **parseInt()**, Double: **parseDouble()**
  - b. Character: **isUpperCase()**, **isDigit()**, **isLetter()**, **toString()**
- 9) Math
  - a. Math constants: **PI**
  - b. Math methods: **sqrt()**, **pow()**, **min()**, **max()**, **round()**, **floor()**, **ceil()**
- 10) Control Loops
  - a. **For**, **while**, **do/while**, **foreach** loops
  - b. **break** / **continue** keywords
- 11) Methods
  - a. Declaring (defining) methods
  - b. Invoking (calling) methods
  - c. Argument values and types
  - d. Return values and types, including **void**
  - e. Passing references vs. primitives
  - f. Method overloading
  - g. Local variables (not initialized)
  - h. Scope of locals

## 12) Single-Dimensional Arrays

- a. Declaring, allocating, initializing arrays
- b. Array indexing, array access
- c. Array **length**
- d. Arrays as method parameters
- e. Arrays as return values
- f. Linear search, Binary search
- g. Arrays class: **toString(), sort(), equals()**

## 13) Multi-Dimensional Arrays

- a. Declaring, allocating, initializing
- b. Array indexing, array access
- c. Array **array.length, array[i].length**

## 14) Classes

- a. Classes versus Objects
- b. Class variables (**static**) and methods
- c. Instance variables (non-static) and methods
- d. Scoping of variables
- e. Object references
- f. **public** versus **private** data and methods
- g. Arrays of objects
- h. **this** keyword

## 15) ArrayLists

- a. Type Parameters (Generics)
- b. Store any type of object
- c. Basic methods – add, clear, indexOf, get, remove, set, size, toString, contains, equals
- d. Iterating through a list
- e. Modifying while looping
- f. Storing primitives in wrapper classes

## 16) Recursion

- a. Calling a method from itself
- b. Stack model for recursion
- c. Helper methods

## 17) File Input/Output

- a. File objects
- b. Scanner basic methods – next, nextLine, nextInt, nextDouble, hasNext, hasNextLine, hasNextInt, hasNextDouble
- c. PrintWriter for file output

## 18) Interfaces

- a. Only contains *abstract* methods
- b. Cannot contain *concrete* methods
- c. **implements** keyword
- d. Arrays defined as interface type can store any class that implements
- e. Comparable

## 19) Linked Lists

- a. Java References
- b. Node class
- c. Get / Add / Remove
- d. Clear / Size / isEmpty
- e. Doubly linked lists

## 20) Sorting

- a. What are  $N^2$  Algorithms?
- b. Bubble
- c. Selection
- d. Insertion