# Career Services Assignment 3 – Java Flash Cards

**Points possible:** 50

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Completeness | All requirements of the assignment are complete. | 100 |

**Instructions:** Research common Java interview questions online and create 20 flash cards from the information you find. Study your flash cards regularly to better prepare for interviews. Fill out the table below with the information you put on each of your flash cards.

|  |  |
| --- | --- |
| **Front of Card** | **Back of Card** |
| Data Types | Primitives/Wrapper : size : value range  byte/Byte: 1 byte : 2^8 (-128 to 127)  short/Short: 2 bytes: 2^16 (-32768 to 32767)  int/Integer: 4 bytes: 2^32  long/Long: 8 bytes: 2^64  float/Float: 4 bytes: 6-7 decimal digits  double/Double: 8 bytes: 15 decimal digits  boolean/Boolean: 1 bit: true/false  char/Character: 2 bytes: single char/letter/ASCII value |
| Access Modifier | public – accessible everywhere  private – only accessible within the class itself  protected – accessible within the class, other classes in the same package, and all subclasses  <no modifier> same as protected, except not accessible in a subclass in different packages  Encapsulation – limit access when possible, and use getters/setters for private |
| 4 Pillars – Abstraction | Keep code simple. Limit what the user is exposed to. Small simple objects are better than large complex objects. |
| 4 Pillars – Encapsulation | Keep internal details internal.  Limit access when possible.  Use Getters/Setters (Accessors/Mutators)  this. keyword |
| 4 Pillars – Inheritance | Subclass inherits from a parent class  May use abstract methods  extends keyword –  super keyword - |
| 4 Pillars – Polymorphism | Allow similar but different implementations |
| Overloading | Using the same method name for different versions, like using different types of arguments or different return types. An example is StringBuilder.append has different versions that do the same thing, for different argument types. |
| Exceptions | Try Catch Block:  try {  …  } catch (Exception e) {  … ex. System.out.println(e.toString());  }  throws keyword  Checked Exception – check in the code (anticipated possibility)  Unchecked Exception – caused by errors in code |
| Measure Time | System.currentTimeMillis()  Current system time in milliseconds elapsed after midnight, January 1, 1970 UTC.  System.nanoTime();  Java high resolution time source, in nanoseconds. Not for absolute time calculation, but good for relative time elapsed calculations |
| Interface | Think of as abstract contract – outlines the inputs, abstract methods  Good practice to code to interface |
| Printf | %[flags][width][.precision] conversion-character  Flags: - left-justify  + output +/- sign  0 zero padded  , comma grouping  ‘ ‘ (space) minus if negative  Conversion characters  d decimal (byte, short, int, long)  f floating-point (float, double)  c character (C will uppercase)  s String (S will uppercase)  h hashcode  n newline |
| Array vs List | Array – size declared; to add object need to specify index  Array List – dynamic size; don’t need to specify index |
| HashMap vs HashTable | HashMap – Methods are not synchronized; Allows one null key and multiple null values; Better performance  HashTable – Methods synchronized and thread-safe; doesn’t allow null; slower performance |
| Lists | Array List – index by insertion order  Vector – thread safety  LinkedList – good for insertion and deletion |
| Set | Hash Set – no order  Linked Hash Set – insertion order  Tree Set – ascending order |
| Map | Hash Map – unordered, allows 1 null key and multiple null values  Hash Table – thread safety, no null  Linked Hash Map – maintains insertion order  TreeMap – ascending order |
| Final | Final keyword sets a variable to not change, as in a constant |
| JDK vs JRE vs JVM | JVM – Java Virtual Machine, provides runtime  JRE – Java Runtime Environment, an implementation of JVM, contains libraries  JDK – Java Development Kit, contains JRE and development tools |
| Static | Static belongs to the thing itself, rather than an instance of the thing. A static variable in a class belongs to the class and not to any one particular instance/object of the class. |
| Input | BufferedReader – thread safety, larger buffer, remember to close  Scanner – thread safety, good for regular expressions, remember to close  Console – only character-based system console, but has some convenient methods |

Sources:

Promineo video lectures on Youtube

[**https://www.softwaretestinghelp.com/core-java-interview-questions/**](https://www.softwaretestinghelp.com/core-java-interview-questions/)

[**https://www.javatpoint.com/corejava-interview-questions**](https://www.javatpoint.com/corejava-interview-questions)

[**https://www.baeldung.com/bufferedreader-vs-console-vs-scanner-in-java**](https://www.baeldung.com/bufferedreader-vs-console-vs-scanner-in-java)