4. How are standard methods like String.toUpperCase() incorporated into the JVM?

**.method public toUpperCase(S)S**

5. What is the difference between a field and a local variable?

**fields within classes are named, not numbered**

6. What is the difference between invokevirtual and invokestatic?

**From the programmer’s viewpoint, there is no difference between invokevirtual and invokespecial, but our standard boilerplate code**

7. Why do static methods take one fewer than nonstatic arguments?

**Because for static methods the object is in location 0.**

8. What is the corresponding type string for the following methods?

a. float toFloat(int) 🡪 **.method public toFloat(F)I**

b. void printString(String) 🡪 **.method public printString(S)S**

c. float average(int, int, int, int) 🡪 **.method public average(IIII)I**

d. float average(int []) 🡪 **.method public average([I)F**

e. double [][] convert(long [][]) 🡪 **.method public convert([[L)[[D**

f. boolean isTrue(boolean) 🡪 **.method public isTrue(Z)Z**

9. What is special about the *<*init*>* method?

**It is invoked with invokespecial**