CS61B Week 5: Java Esoterica

List one or more reasons you might encounter each of the following exceptions. Also, specify whether each exception is checked or unchecked.

IOException

ArrayIndexOutOfBoundsException

NoClassDefFoundError

OutOfMemoryError

NullPointerException

StackOverflowError

IllegalArgumentException

ClassCastException

What do these compiler messages mean (besides that you did something wrong)?

```
MyClass.java:23: cannot find symbol
symbol  : method foo(java.lang.String,int)
location: class MyClass
    foo("blah", 8);
    ^
```

```
MyOtherClass.java:47: bar(int) in MyOtherClass cannot be applied to (java.lang.String,int)
    bar("cat", -100);
    ^
```

```
Note: MyThirdClass.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

The United States Congress, in its infinite wisdom, passes a law that all nickels are worth 7 cents. Calculate the least number of coins you need to make change for exactly n cents. Assume you can only use combinations of pennies, dimes, quarters, and these new nickels (use brute force!).

```
public static int makeChange(int n) {
```

Sample Midterm Question of the Week:

```
/** Slice the list L into a list of N lists such that list #k contains
 *  all the items in L that are equal to k modulo N, in their original
 *  order. For example, if N is 3 and L contains [9, 2, 7, 12, 8, 1, 6],
 *  then the result is [ [9, 12, 6], [7, 1], [2, 8] ]. The operation
 *  is destructive (it may destroy the original list) and creates no new
 *  IntList objects (it will, of course, create new IntList2 objects).
 */
static IntList2 dslice (IntList L, int n) {
```

Sample Interview Question of the Week:
The makeChange function we wrote above starts to get really slow as we try larger numbers (even just $20, or 2000 coins, takes a few seconds to calculate). Write a more efficient implementation. Your program, running on an instructional machine, should easily be able to make change for numbers of coins in the $10,000 range.