CSPP51036
Fall 2009
Homework 2
Due: Mon. Oct 25, 5:30pm

1. Use the java.util.Currency class to write a program that takes as
input a valid ISO 4217 currency code (Google it -- e.g. "USD", etc.)
and prints out the currency "symbol" (e.g. "$").

2.. Write a class MutableString.java that supports the basic
java.lang.String API but that also contains a set method (to
change the value of the String). You should do this by leveraging the
String class. I do not expect you to recode all of the basic String
functionality. Include a main program to demonstrate this functionality.

Also, I say "basic" String functionality because it is not necessary
to support every method in java.lang.String . Picking your favorite
five or so as fine as long as it is clear how the others could be
included via the same strategy. hint: Your class should be very small. If
approached correctly this is a ten minute exercise.

Also, yes, I realize that java.lang.String is final.

3.

a. Create a very simple Student class with some appropriate constructors, a few methods, etc. Include a gpa field.

b. Create and intialize an array of students and use the built-in Arrays.sort method to sort by gpa.

c. Use Arrays.sort to then randomly scramble the list (be careful).

---

4. Write a simple sort method in java (bubble sort is fine) that can sort 1d arrays of arbitrary objects. You decide how this is all set up, what requirements are placed on an object such that it can be used with your sort method, etc. This should be more of a design than a coding exercise (since bubble sort is trivial). Be sure to document your method so that it is clear how it is used.

*Note that the java library contains such a capability. I would like you to create your own to confront the design issues that arise when doing this in java.

---

5. It is often not possible to analytically solve a simple algebraic equation. For a simple case such as

    f(x) = x^2 + 5x + 6 = 0

we can "factor" the equation

    f(x) = (x+2)(x+3) = 0

so that the solution ("roots") are obvious (x=-2 and x=-3). In most cases though the equation cannot be factored and we rely on numerical techniques to find the roots. The most popular such technique is called Newton Raphson. It is very simple and pretty reliable and efficient for many well behaved cases. The algorithm is as follows:

 (1) start with a guess x0
 (2) correct the guess with the x-intercept of the line
   that passes through x0 and whose slope is the local
   tangent to f
 (3) repeat until convergence

step (2) requires computing the derivative of f at point x0 (since the derivative is the local slope). This can be done many ways and is obviously unique to each f(x). The point of this
excerise is to write the Newton Raphson code generally, allowing
the user to "pass in" the function that computes both the derivative
and the value of the function at x0. In Java this is done using
interfaces, as discussed in class.

a) Write a Newton-Raphson root finder for an arbitary function f.
b) Test this for a specific f(x). Something very simple like (x-2)^2
is fine. (The derivative is just 2x-4).