

COMPUTER SCIENCE A – HOMEWORK #8

This assignment builds on the abstract class `Currency`, and on the class `Peso`, both of which were presented in lecture. In this assignment, you will write three separate programs, as described later. In any case, the listing of `Currency.java` is as follows.

```
public abstract class Currency implements Comparable<Currency> {
    /**/
    public Currency () { }
    /**/
    public abstract double getUSDvalue ();
    /**/
    public int compareTo ( Currency c ) {
        /**/
        double dif=100.0*(getUSDvalue()-c.getUSDvalue());
        return (int)(dif+0.5);
    }
}
```

Also, class `Peso` is as listed below.

```
public class Peso extends Currency {
    /**/
    private static double VALUE = 0.053; // in USD as of Mar 27 2017
    private double amount;
    /**
     * pesoAmount must be positive
     */
    public Peso ( double pesoAmount ) {
        amount=pesoAmount;
    }
    /**
     * abstract in Currency
     */
    public double getUSDvalue() {
        return amount*VALUE;
    }
    /**/
    public static String toString ( double usdAmount ) {
        /**/
        double pesoAmount=usdAmount/VALUE;
        Peso peso=new Peso(pesoAmount);
        return peso.toString();
    }
    /**/
    public String toString() {
        /**/
        int numWhole=(int)Math.log10(amount)+1;
        String wholeS;
        if ( numWhole == 0 )    wholeS = "4";
        else                  wholeS = Integer.toString(numWhole+3);
        String formatS = "%" + wholeS + ".2f";
        return "Mex$" + String.format(formatS,amount);
    }
}
```

COMPUTER SCIENCE A – HOMEWORK #8

1) Write four more classes which extend `Currency`, *i.e.*,

a) class `Euro`.

Use €1.00 = \$1.09 as the exchange rate.

`toString()` should output something like

€87.42 (the character '€' is the *unicode* character '\u20AC')

b) class `CanadianDollar`.

Use Can\$1.00 = \$0.75 as the exchange rate.

`toString()` should output something like

Can\$87.42

c) class `USDollar`.

Use \$1.00 = \$1.00 as the exchange rate.

`toString()` should output something like

\$87.42

d) class `PoundsSterling`.

Use £1.00 = \$1.26 as the exchange rate.

`toString()` should output something like

£87.42 (the character '£' is the *unicode* character '\u00A3')

2) Write a program `ConvertCurrency.java` which converts in between the five currency types. Use the file `currency_input.txt` as input to your program. The file `currency_input.txt` is listed below.

```
Mex$:123.87
Euro:329.42
Can$: 25.52
USD$: 52.80
UKlb:160.02
```

The output of your program should look something like

```
Mex$123.87 =
Mex$123.87
    €6.02
    Can$8.75
    $6.57
    £5.21

    €329.42 =
Mex$6774.86
    €329.42
    Can$478.76
    $359.07
    £284.97

    Can$25.52 =
Mex$361.13
    €17.56
    Can$25.52
    $19.14
    £15.19

    $52.80 =
Mex$996.23
    €48.44
    Can$70.40
    $52.80
    £41.90
```

COMPUTER SCIENCE A – HOMEWORK #8

£160.02 =
 Mex\$3804.25
 €184.98
 Can\$268.83
 \$201.63
 £160.02

Use the file `ConvertCurrencyStub.java`, which is listed below, as a starting point.

```

import IO.FileInput;
/**/
public class ConvertCurrency {
    /**/
    private Currency [] cList;
    /**/
    public static void main ( String [] arg ) {
        /**/
        ConvertCurrency cc;
        /**/
        if ( arg.length != 2 ) {
            System.out.println();
            System.out.println("Usage:");
            System.out.println("java ConvertCurrency inputFile outputFile");
            return;
        }
        /**/
        cc=new ConvertCurrency(arg[0]);
        /*
        * Put code here to:
        *
        * Print out to the outputFile for each cList value:
        * (a) the Peso value
        * (b) the Euro value
        * (c) the CanadianDollar value
        * (d) the USDollar value
        * (e) the PoundsSterling value
        */
        return;
    }
    /**/
    public ConvertCurrency (String inputFile) {
        readInput(inputFile);
    }
    /**/
    public Currency [] getCurrencyList() {
        return cList;
    }
    /**/
    private void readInput ( String inputFile ) {
        /*
        * Put code here to:
        *
        * Read the input file inputFile into a String [], and then
        * convert it to the private field Currency [] cList.
        *
        * Note:
        *
        * You may want to use the private method Currency getCurrency(String,double).
        */
        return;
    }
}

```

COMPUTER SCIENCE A – HOMEWORK #8

```

/**/
private static Currency getCurrency ( String type, double val ) {
    /**/
    if      ( type.equals("Mex$") ) return new Peso(val);
    else if ( type.equals("Euro") ) return new Euro(val);
    else if ( type.equals("Can$") ) return new CanadianDollar(val);
    else if ( type.equals("USD$") ) return new USDollar(val);
    else
        return new PoundsSterling(val);
}
}

```

We want to sort a list of Currency objects. To that end:

3) Modify class BubbleSortStub as follows, to make a class BubbleSort:

- a) Change** `public static void bubbleSort (Sortable [] s)` to `public static <T extends Comparable<T>> void bubbleSort (T [] s)`.
- b) Change** `Sortable temp` to `T temp`.
- c) Change** `if (s[i+1].lessThan(s[i]))` to `if (s[i+1].compareTo(s[i]) < 0)`.

4) Write a program SortCurrency.java which sorts an input file of currency values into ascending order. Use the file currency_list.txt as the input to your program, the first few lines of which are shown here.

```

Mex$:123.87
Euro:329.42
Can$: 25.52
USD$: 52.80
UK1b:160.02
.
.
.

```

Use the file SortCurrencyStub.java as a starting point, which file is listed here.

```

import IO.FileInput;
/**/
public class SortCurrency {
    /**/
    Currency [] ca;
    /**/
    public static void main ( String [] arg ) {
        /**/
        Currency [] list;
        SortCurrency sc;
        /**/
        if ( arg.length != 2 ) {
            System.out.println();
            System.out.println("Usage:");
            System.out.println("java SortCurrency inputFile outputFile");
            return;
        }
        /**/
        sc=new SortCurrency(arg[0]);
        list=sc.getCurrencyList();
        BubbleSort.bubbleSort(list);
    }
}

```

COMPUTER SCIENCE A – HOMEWORK #8

```

/*
 * Put code here to:
 *
 * Print out to the outputFile the sorted list.
 */
return;
}
/**/
public SortCurrency ( String inputFile ) {
    readInput(inputFile);
}
/**/
public Currency [] getCurrencyList() {
    return ca;
}
/**/
private void readInput ( String inputFile ) {
/*
 * This method should be about the same
 * as occurs in class ConvertCurrency.java.
 */
return;
}
}

```

Note that `BubbleSort.bubbleSort(list)` will perform the sort. In any case, the first few lines of output of your program should look something like

```

Mex$123.87 =    $6.57
Can$25.52 =   $19.14
    $23.01 =   $23.01
Mex$478.77 =   $25.37
Mex$666.66 =   $35.33
.
.
.

```

- 5) Modify your program `SortCurrency.java` into `CannedSortCurrency.java` by changing `BubbleSort.bubbleSort(list)` to the built-in java method `Arrays.sort(list)`. You should additionally import

```
import java.util.Arrays;
```

Once again, test your program with the input file `currency_list.txt`.

Once your programs are working correctly, submit your files `Euro.java`, `CanadianDollar.java`, `USDollar.java`, `PoundsSterling.java`, `ConvertCurrency.java`, `BubbleSort.java`, `SortCurrency.java`, and `CannedSortCurrency.java` to me at sharren@d131.org.