COP 3540 – Data Structures with OOP
Program #1
Due: 15 Sept 2010 – Start of Class
Drop Dead date: 20 Sept 2010 – Start of Class

Using NetBeans 6.7 or 6.8, you are to write a Java program using OOP principles to accommodate the following functionality

Assignment #1

Objectives:

Provide student with experience building arrays of objects
Provide student with opportunity in doing file input and output.
Provide student with exercises in learning UML
Provide student with exercises in Javadoc and its various formats
Provide student with exercises in searching, sorting, and comparisons of key searches and sort routines.

Functionality:

Using the external file, AsiaCountries.Fall2010, (Open with WordPad) you are to do the following:

1. **Build Array of Country Objects.** Using input file, you are to build an array of objects. You are to design and develop a class named **Country** (in addition to your class named **Main)** and create objects of **type (class) Country** - one object for each record (line) from the input file. (You should use Buffered Reader class). **Each Country object will have five properties defined individually as integers or Strings as appropriate for each Country attribute. You should download this file and use this as input to your program to create the array of Country objects. Be certain to drag this file into your project folder.** (Be aware that this file may be modified for follow on programming assignments.)

2. **Print Array of Country Objects.** From main() you are to access this array of Country objects and write code to print the array. Use toString() method in the Country objects you will be creating. You are to print a column header, followed by a blank line, followed by single spaced (one line of attributes per object (country) as output. All attributes are to be printed per country object. These are to be single spaced and all aligned nicely. (Attributes are Country Name, Capital, Region, and Region_Nbr and CountryCode.)

3. **Sort and Display** array using an ascending bubble sort using the country name attribute of each object as your sort key.

4. **Sequential Search and Print Results.** Given the input search file, **Search.Fall2010** (a text file to be supplied) you are to search the **sorted array** to 'find' or 'not-find' each **search argument** in the search file using a **sequential search**. **Your search key (from the transaction file) is the two-character country code. STEP 4 AND 5 WILL REQUIRE YOU TO RESORT THE ARRAY OF COUNTRIES BASED ON COUNTRY CODE.**

After skipping two lines and displaying an appropriate column header one time, for each search key you are to display the search key itself, number of probes, and text: 'found' or 'not found' to indicate the success of your search. **(Some of the input search arguments may not match. This is intended. Clearly USA is not a country name, nor is Japan. But JA is; ZZ is not. You will thus get a 'no-hit' on some of these in the search file.** (Others are misspelled intentionally.)

5. **Binary Search.** Similar to (4) above, but use a **binary** search. Your results, as above, are to produce a column header and to display the search key, number of probes to 'find' or 'not find' the target followed by text 'found' or 'not found' text. Again, line these up and include appropriate spacing between each of these attributes on a line. Be certain to skip two lines between the search results generated from (4) and (5). For each search argument, you need only display the same format as above: search key, number of probes, and found or not found indicators. The same input file is used…

6. **Ragged Array.** Given the four Asian geographical areas, you are to build four ragged arrays – one for each geographical region. Each array is to have **only** the objects belonging to that region of Asia.

7. **Printing Ragged Arrays.** By sending each of these ragged arrays to a special print routine, you are to then display in a professional manner each of the four one-dimensional array of region objects in turn and their respective countries in the following format:

East Asia Countries <note countries below are listed in alphabetical order>

<one blank line>

Country name Capital

<one blank line>

China Beijing (entries are to be left justified under the header)
Japan Tokyo
Etc.

<two blank lines>

North Asia Countries.

Country name Capital

Kazakhstan Astana
Kyrgyzstan Kishkek <note these are single spaced>

Etc.

At the end of printing, you are to display:

Total Countries in East Asia: xxxxx <line up these columns>
Total Countries in North Asia: xxxxx
…

Total Countries in Asia: xxxxx

All done. Be proud of your work.

UML

You are to include a UML class diagram. You may use Word or Power Point only. You may also use SmartDraw, if you wish. Do NOT use Visio or other applications. **No other technology may be used!** Once you have developed your UML (architectural design), be certain to drag this file into your project folder and be certain it is included with the zip file you will turn in to me.

Use the examples in your 2551 text. Each class listed in your UML diagram must have attributes listed (name, type). Methods must be shown with visibility indicator (+, -) , and number /type of arguments plus the return type.