

AP COMPUTER SCIENCE B – INTRODUCTION TO INTERFACES

Consider the interface Sortable (file Sortable.java) as listed here.

```
public interface Sortable {
    /**/
    boolean lessThan ( Sortable s );
}
```

An interface is not a class: it cannot be instantiated with a Sortable constructor (there is none). Rather a class implements an interface. For example, the class Date (in file DateStub.java), listed below, implements the interface by having the method `public boolean lessThan (Sortable other)`. We will return to this class later in problem 1.

```
public class Date implements Sortable {
    /**/
    private static String [] MONTHS = new String [] {
        "Jan" , "Feb" , "Mar" , "Apr" , "May" , "Jun" ,
        "Jul" , "Aug" , "Sep" , "Oct" , "Nov" , "Dec"
    };
    /**/
    private int comparable;
    private String s;
    /*
    * s = 03Mar1960
    */
    public Date ( String s ) {
        /**/
        this.s=s;
        initComparableInt();
    }
    /**/
    public int getComparableInt () {
        return comparable;
    }
    /**/
    public String toString () {
        return s;
    }
    /**/
    public boolean lessThan ( Sortable other ) {
        /**/
        Date otherDate=(Date)other;
        /*
        * As per interface Sortable.
        *
        * Put code here to determine whether or not
        * this Date is less than otherDate.
        */
    }
    /**/
    private void initComparableInt () {
        /*
        * Put the needed code here to
        * initialize the int comparable
        * as explained in the class work
        * handout.
        */
    }
}
```

AP COMPUTER SCIENCE B – INTRODUCTION TO INTERFACES

The goal of this assignment is to write a program `SortDates` which will sort a list of dates into chronological order. Consider the (input) file `dates.txt`, the first few lines of which are listed here.

```
11Aug2003
27Oct1973
09Dec1983
12Jun2005
02Feb1992
02Jun1996
19Aug1978
.
.
.
```

Note that a `Date` object can be instantiated via, *e.g.*, `Date d=new Date("11Aug2003")`. In any case, a `Date [] da` can be sorted via `BubbleSort.bubbleSort(da)` with class `BubbleSort`, listed here.

```
public class BubbleSort {
    /**/
    public static void bubbleSort ( Sortable [] s ) {
        /**/
        int len;
        Sortable temp;
        /**/
        len=s.length;
        for ( int p=0; p<len-1; ++p ) {
            for ( int i=0; i<len-p-1; ++i ) {
                if ( s[i+1].lessThan(s[i]) ) {
                    temp=s[i+1];
                    s[i+1]=s[i];
                    s[i]=temp;
                }
            }
        }
        /**/
        return;
    }
}
```

Note that in class `BubbleSort`, the `Date [] da` is declared as being a `Sortable []`. Again, this is called *polymorphism*. Thus, any list of objects whose class implements `Sortable` can be sorted with class `BubbleSort`.

1) Complete class `Date` by:

- a) Adding the needed code to method `private void initComparableInt ()`. A good way to proceed, *e.g.*, with the date string `03Mar1960`, is to create three ints `d`, `m`, and `y`. The `03` should be converted to `3` and put into `d`. The `Mar` should be converted to `2`, *i.e.*, the index of `Mar` in array `MONTHS`, and put into `m`. Finally, the `1960` should be converted into the int `y`. The integer `comparable=10000*y+100*m+d` then can be used to compare the dates.
- b) Adding the needed code to method `public boolean lessThan (Sortable other)`. Note that you will have to compare, via `<`, the two integers `comparable` and `otherDate.getComparableInt()`.

AP COMPUTER SCIENCE B – INTRODUCTION TO INTERFACES

- 2) Write a class `SortDates` containing a main method which:
- a) Takes the input and output file names from the argument list. If the number of arguments is not as expected, the program should print a `Usage` message and `return`.
 - b) Reads in the input file into an `ArrayList<String>`, and then creates a `Date [] da`.
 - c) Sorts the `Date [] da`.
 - d) Prints the sorted `Date [] da` to an output file.

The first few lines of your output file should be:

```
18Feb1970
24Mar1970
05Apr1970
26Jul1970
30Jul1970
24Sep1970
17Oct1970
04Nov1970
.
.
.
```

Once your program is working correctly, email the files `SortDates.java` and `Date.java` to me at sharren@d131.org.