

COP 3540 – Data Structures with OOP

Program #1

Due: 20 Sep 2006

Using NetBeans 5.0, you are to write a Java program using OOP principles to accommodate the following functionality

Assignment #1

Objectives:

- Provide student with experience using random numbers and the API
- Provide student with opportunity in doing file input 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:

1. Generate a sequential file of 1000 integers
 - You may use the class Random to accommodate this.
 - You may also use the routines in my web page to read a file and write to a file.
2. Build an integer array from these 1000 integers
3. Display every 25th integer – five to a line.
4. Sort the array using a bubble sort
 - Count the number of interchanges
 - Upon completion of the sort, the driver (main) is to display the number of interchanges.
5. Display every 25th integer – five to a line.
6. Given an input search file (see my web page. File contains 10 integers), you are to search the array to find / not-find each one using a sequential search. Your results are to display the search key, number of probes, and a 'found' or 'not found' literal.

7. Same as (6) above, but use a binary search. Your results, as above, are to display the search key, number of probes to find or not find the target followed by the 'found' or 'not found' literal.

UML

You are to include a UML class diagram. You may use Word or Power Point. Drag your UML design file into your P1 subfolder within your COP3540 desktop folder. It will be included in the zip file to me.

Javadoc

All programming is to be accompanied by appropriate Javadoc. Generate your Javadoc files and include the xxxxxxxx file generated.

You are to zip all files in your P1 as expected and Send them to be via Digital Dropbox using the same naming conventions as in P0.

Grading

Source Code – 30 points

- Indentation
- Internal comments
- Scope terminators
- Overall program structure

Program Design – 20 points

- Appropriateness of the objects and their services provided
- Interface to objects
- Attribute and method visibility

Javadoc – 10 points

- Appropriateness and completeness of comments

UML – 10 points

- Correctness, associations, completeness. This means that the classes you identify are correct, that associations are indicated, and that the attributes and methods are documented within the classes.

Outputs – 30 points

- Accuracy and Format
- Skip lines in between displayed numbers for readability.
- Include headers / descriptors as you may feel appropriate.

**Program must run correctly to receive a passing grade.
Start early and do this a little at a time.**