**NO LATE SUBMISSIONS WILL BE ACCEPTED**

------------------------------------------------------------------------------------------------------------------------------

**NOTE**:  Be  sure  to  adhere  to  the  **University’s  Policy  on  Academic  Integrity**  as  discussed  in  class.

Programming assignments are to be written individually and submitted programs must be the result of your **own** efforts. Any suspicion of cheating will be dealt with accordingly. The minimum punishment I will impose for Academic integrity violation will be a **0%** on the programming assignment and a **2-letter grade** **reduction** in the course.  Please refer to the following site for additional details, as it is your responsibility to be aware:

http://www.legal.uncc.edu/policies/ps-105.html

**Important**: I reserve the right to revisit program grades after completion of semester and final exams. If there is a marked discrepancy in program grades and your answers to related questions on tests, this will be challenged and appropriate grade adjustments given. In addition, appropriate measures will be taken with respect to the University’s policy on Academic Integrity where indicated as a result of the outcome of such a challenge.

**Purpose**: This assignment is designed to demonstrate your understanding of the creation, compilation, and program submission process we will be using this semester.

**Assignment Description:**

In this assignment, you will create a program that allows the user to choose between the following menu choices (menu-driven program):

1. Linear Search
2. Binary Search
3. Bubble Sort
4. Selection Sort
5. Quit

Keep running the program until the user chooses to Quit. Your program should have the following parts:

**Searching Algorithms (Linear and Binary Search)**You are the owner of a book store. You have the following information available in your stock (information is stored in the form of parallel arrays):

String[] bookTitle = {“Starting out with Java”, “Java Programming”, “Software Structures”, “Design and Analysis of Algorithms”, “Computer Graphics”, “Artificial Intelligence: A Modern Approach”, “Probability and Statistics”, “Cognitive Science”, “Modern Information Retrieval”, “Speech and Language Processing”};

int[] bookID = {1101, 1211, 1333, 1456, 1567, 1642, 1699, 1755, 1800, 1999};

double[] bookPrice = {112.32, 73.25, 54.00, 67.32, 135.00, 173.22, 120.00, 42.25, 32.11, 123.75};

First **display()**the above information to the user in a tabular format, using a void method. Your program should then ask for the book ID and the number of books the user wishes to purchase. Based on the book ID provided by the user, display the following information (if the ID is found):

* Book ID
* Book Title
* Number of books bought
* Total cost of the purchase

*If the book ID is not found*, display a message saying so. The book ID needs to be searched based on **linearSearch()**or **binarySearch()**(based on user choice from the menu).

**Sorting Algorithms (Bubble and Selection Sort)**

Your program should generate 1000 random numbers in the range of 1 to 500 and store them in an array. Use **bubbleSort()**or **selectionSort()**(based on the menu choice) to sort the array of numbers. Display both the unsorted and the sorted array.

**Instructions:**Please make sure your code has following **functions:**

* **display():**To display the contents of parallel array in a tabular format. Take in the three different arrays as parameters.
* **linearSearch():**To apply the linear search algorithm to search for the book ID. Returns the index position, or -1 if not found
* **binarySearch():**To apply the binary search algorithm to search for the book ID. Returns the index position of the found book ID, or -1 if not found.
* **bubbleSort():**To apply the bubble sort algorithm to sort the elements of an unsorted array
* **selectionSort():**To apply the selection sort algorithm to sort the elements of an unsorted array

You can use additional methods (optional) for other operations. Make sure your program runs until the user decides to quit the program. Your program should **validate**(input validation) the menu choice entered by the user, and force them to reenter a menu choice if their original input was invalid. A bonus programming part has been provided below for extra points.

**Bonus Part (Optional): 20 points**

We want to test the efficiency of our searching and sorting algorithms. To test the efficiency, calculate and display the execution (elapsed) time (in seconds) required for each of the Searching and Sorting techniques. Execution time is calculated from the start of a function call to the end of the function. Look up online resources on how to calculate elapsed time and using the system time library.

Can you tell which searching technique is better/faster (linear search vs. binary search) and which sorting technique is better/faster (bubble sort vs. selection sort)?

**Important:**

A grading rubric has been provided to you as a reference. A few important things to remember:

1. Start working early. Do not wait till the due date to start working on your assignment.

1. Always store a backup copy of your code online.

1. Make sure you submit a **.java** file. A **.class** file will result in grade of 0.

1. Ask for  timely  help.  You  have  various  resources  available  (Professor,  TA’s,  CCI  Tutors),

make sure you ask for help if you are struggling with something.

1. **Submit your own work!**

**Sample Run:**

1. Linear Search
2. Binary Search
3. Bubble Sort
4. Selection Sort
5. Quit

Please enter a number between 1-5: 1

\*\*\*\* LINEAR SEARCH \*\*\*\*

Book ID                                                   Title                                                                         Cost

1101                     Starting out with C++                        112.32

1211                     Java Programming                             73.25

1333                     Data Structures                                   54

1456                     Design and Analysis of Algorithms:           67.32

1567                     Computer Graphics                            135

1642                     Artificial Intelligence: A Modern Approach   173.22

1699                     Probability and Statistics                   120

1755                     Cognitive Science                            42.25

1800                     Modern Information Retrieval                 32.11

1999                     Speech and Language Processing               123.75

Please enter the book ID you wish to purchase from the list above: 1211

How many copies you wish to purchase: 2

Book ID: 1211

Book Title: Java Programming

Number of books bought: 2

Total Cost: $146.5

                Execution of this function took: 0.143 seconds.

1. Linear Search
2. Binary Search
3. Bubble Sort
4. Selection Sort
5. Quit

Please enter a number between 1-5: 2

\*\*\*\* BINARY  SEARCH \*\*\*\*

Book ID                                                   Title                                                                         Cost

1101                     Starting out with C++                        112.32

1211                     Java Programming                             73.25

1333                     Software Structures                          54

1456                     Design and Analysis of Algorithms:           67.32

1567                     Computer Graphics                            135

1642                     Artificial Intelligence: A Modern Approach   173.22

1699                     Probability and Statistics                   120

1755                     Cognitive Science                            42.25

1800                     Modern Information Retrieval                 32.11

1999                     Speech and Language Processing               123.75

Please enter the book ID you wish to purchase from the list above: 1211

How many copies you wish to purchase: 2

Book ID: 1211

Book Title: Java Programming

Number of books bought: 2

Total Cost: $146.5

                Execution of this function took: 0.215 seconds.

1. Linear Search
2. Binary Search
3. Bubble Sort
4. Selection Sort
5. Quit

Please enter a number between 1-5: 3

\*\*\*\* BUBBLE SORT \*\*\*\*

The unsorted array is:

384387  278  416  294  336  387  493  150  422

The sorted array is:

150278  294  336  384  387  387  416  422  493

Execution of this function took: 0.018 seconds.

1. Linear Search
2. Binary Search
3. Bubble Sort
4. Selection Sort
5. Quit

Please enter a number between 1-5: 4

\*\*\*\* SELECTION SORT \*\*\*\*

The unsorted array is:

363   28   191  60   264  427  41   427  173  237

The sorted array is:

28   41   60   173   191  237  264  363  427  427

Execution of this function took: 0.016 seconds.

1. Linear Search
2. Binary Search
3. Bubble Sort
4. Selection Sort
5. Quit

Please enter a number between 1-5: 5

**Grading Scheme:**

The assignment is worth 100 points but you can achieve up to 120 points if you do the extra credit. The following is the grading Scheme:

**Program completeness with pseudocode:**                                       20 Points

**Documentation:**

Header documentation (Author, Date, Description)                                 15 Points

Internal documentation (what your program is doing)                              10 Points

**Process:**

Correct use of arrays in searching/sorting                                              25 Points

Correct calculations/process                                                                 20 Points

**Output:**

Well-formatted correct output                                                                 10 Points

**Extra Credit                                                                                        20 Points**

**Total:                                                                                                  120 Points**

Make sure to submit your assignment based on the following guidelines:

* Rename your file with a meaningful name.
* Upload your file on your lab canvas page before the due date. There will be a submission link on your canvas page.
* Make sure you upload the correct filename.java file. Don not upload the class file (.class) as this will result in no grade for your assignment.