ASU Computer Science and Engineering Department

CSE 110 - Assignment 7

Maximum Points: 20

What this Assignment Is About:

- Arrays
- Classes
- Searching

Your programming assignments require individual work and effort to be of any benefit. Every student must work independently on his or her assignments. This means that every student must ensure that neither a soft copy nor a hard copy of their work gets into the hands of another student. Sharing your assignments with others in any way is **NOT** permitted.

Violations of the University Academic Integrity policy will not be ignored. The university academic integrity policy is found at http://www.asu.edu/studentlife/judicial/integrity.html

Use the following Coding Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- Use upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.
- Use comments after the ending brace of classes, methods, and blocks to identify to which block it belongs.

Assignments Documentation:
At the beginning of each programming assignment you must have a comment block with the following information:

```
/*------------------------------------------------------------------------
// AUTHOR: your name
// FILENAME: title of the source file
// SPECIFICATION: description of the program
// YOUR Lab Letter and Name of the TA for your Closed lab
// FOR: CSE 110- homework #- days and time of your class
// TIME SPENT: how long it took you to complete the assignment
//----------------------------------------------------------------------*/
```

Part 1: Written Exercises: (5 points)

Consider the following array:

```
int[] a = { 3, 5, , 8, 10 , 12};
```

Write the contents of the array **a** after the following loops. Use the original data above for each question. (1 Point each)

```
a) for (int i = 1; i < 6; i++) { a[i] = a[i - 1]; }
b) for (int i = 5; i > 0; i--) { a[i] = a[i - 1]; }
c) for (int i = 0; i < 5; i++) { a[i] = a[i + 1]; }
d) for (int i = 4; i >= 0; i-=2) { a[i] = a[i + 1]; }
e) for (int i = 1; i < 6; i++) { a[i] = a[5 - i]; }
```

Part 2: Programming: (15 points)

Your assignment is to create a class called Collection in a file called **Collection.java**. (there is no main method in this class). A class Collection has an array of integers and a count (integer) as instance variables. The variable **count** keeps track of how many integers are store in the array. The variable name for the array of integers is **numArray**.
Note: You need to distinguish the array size (capacity) and "count" that keeps track of numbers added to this array so far.

The class Collection **must** include the following constructor and methods. (If your class does not contain any of the following methods, points will be deducted.)

Method	**Description of the Method**
`public Collection(int arraySize)`	It constructs an empty Collection object with an array capacity specified by the integer parameter "arraySize".
`private int search(int searchingNum)`	It returns the index of the number specified by the parameter is located. If the number is not found, it returns -1. It is a service (helper) method.
`public boolean add (int numberToAdd)`	The method checks if the integer specified by the parameter exists in the array (This can be done using the search method to see if it returns -1 or not) and also checks if the array has not reached its capacity. If both are satisfied, the number is added to the array at the smallest available index. If the array reached its capacity, double its size by calling the method doubleArrayCapacity() and add the number. If the number is added successfully, then the method returns true. If the number already exists in the array, the new number will not be added, and the method returns false.
`public boolean remove(int numberToRemove)`	The method checks if the integer specified by the parameter exists in the array (This can be done using the search method to see if it returns -1 or not) and if it does, it deletes the number by shifting numbers to the left and it returns true. Otherwise, it returns false.
`public int getCount()`	Returns the number of elements added so far to the array numArray.
`public int[] rotate (int n)`	The method takes an integer rotation amount, n. The method creates a new array with the items of numArray moved forward by n positions. Elements that are rotated off the array will appear at the end. For example, suppose numArray contains the following items in sequence: 1 2 3 4 5 6 7 After rotating by 3, the elements in the new array will appear in this sequence: 4 5 6 7 1 2 3

`private void doubleCapacity()`	It is a service (helper) method and doubles the capacity of the numArray. Please see the example in page 338, the method increaseSize() as a reference.

`public int[] swapSmallest()`	The method returns an array by swapping the smallest integer element of an array referenced by numArray into index 0. For example if numArray is
	{ 5, 7, 3, 9, 2, 11, 1, 6 }
	After calling the method, it returns a new array with smallest element stored at index 0
	{ 1, 7, 3, 9, 2, 11, 5, 6 }

`public String toString( )`	Returns a String containing a list of numbers stored in the numArray. An example of such string can be: {3, 6, -1, 3, 23, -50, 43} The string should start with a '{' and end with a '}'.

Save the Collection class in a file called Collection.java and use the following program stored in Assignment7.java, which has the main method to create new Collection objects and to test your class. **You need to modify Assignment7.java by adding your code(s) for the cases 'a', 'b', 'c', 'd', and e'.**
The program will ask a user to enter a size for the array as an integer. Then it will show the following menu to a user:

```
Command Options
-----------------------------------
a: add an integer in the array
b: remove an integer from the array
c: display the array
d: swap with the smallest
e: rotate the array
?: display the menu again
```

Then it will ask a user to enter one of the above commands. Based on the user's choice, the program needs to perform corresponding operation. This will be done by using a method you define in the Collection class. The program will terminate when a user enters 'q'.

Note that you will NOT be removing or modifying the existing codes in the Assignment7.java. All you need to do is to add more codes.

After compiling Collection.java file and Assignment7.java file, you need to execute Assignment7 class.

Sample Output: (the inputs entered by a user are shown in bold)
(Make sure that your program works at least with this scenario.)

```
Please enter a size for the array.

5

Command Options
```

```
-----------------------------------
a: add an integer in the array
b: remove an integer from the array
c: display the array
d: swap with smallest
e: rotate
?: display the menu again
q: quit this program


Please enter a command or type ?
a

Please enter an integer to add.
12

12 successfully added.

Please enter a command or type ?
a

Please enter an integer to add.
20

20 successfully added.

Please enter a command or type ?
a

Please enter an integer to add.
30

30 successfully added.

Please enter a command or type ?
c
{12, 20, 30}

Please enter a command or type ?
b
Please enter an integer to remove.
12

12 successfully removed.

Please enter a command or type ?
c
{20, 30}

Please enter a command or type ?
a

Please enter an integer to add.
```

```
10

10 successfully added.

Please enter a command or type ?
a

Please enter an integer to add.
40

40 successfully added.

Please enter a command or type ?
a

Please enter an integer to add.
1

1 successfully added.

Please enter a command or type ?
c
{20, 30, 10, 40, 1}

Please enter a command or type ?
d
1 30 10 40 20

Please enter a command or type ?
e
Enter rotation count:
3

The rotated array is:
40 1 20 30 10

Please enter a command or type ?
c
{20, 30, 10, 40, 1}

Please enter a command or type ?
e
Enter rotation count:
2

The rotated array is:
10 40 1 20 30

Please enter a command or type ?
q
```

****

Helpful hints for doing this assignment:

- work on it in steps ~V write one method, test it with a test driver and make sure it works before going on to the next method
- always make sure your code compiles before you add another method
- your methods should be able to be called in any order

****

Submit your homework by following the instructions below:

****

- Go to the course web site (my.asu.edu), and then click on the on-line Submission tab.
- Submit your **Assignment7.java and Collection.java** files on-line. Make sure to choose Hw7 from drop-down box.
- Assignment7.java should have the following, in order:
    - In comments, the assignment Header
    - In comments, the answers to questions for part 1
    - The working Java code requested in part 2.
    - The Assignment7.java and Collection.java files must compile and run as you submit it. **You can confirm this by viewing your submission results.**

Important Note: You may resubmit as many times as you like until the deadline, but we will only mark your last submission.

NO LATE ASSIGNMENTS WILL BE ACCEPTED.