CPS 501 **Assignment 6**

Assigned: 03/23/2020. Due: 03/29/2020, 11 pm (on Isidore).

Instructor: Tamisra H. Sanyal

1. Please complete the following Java class which is an implementation of a buffer of integers.

class IntBuff

{

int[] elements; // the content of the buffer

int current\_size; // the number of valid elements

IntBuff() // initialize an empty buffer of maximum size 0

{

...

}

// initialize an empty buffer of maximum size n

IntBuff(final int n)

{

...

}

// initialize a buff which is a copy of buff

IntBuff(IntBuff buff)

{

...

}

// append the value v to the buffer

// return true if the operation is successful otherwise false

boolean append(final int v)

{

...

}

// insert the value v to the buffer at the position of index

// return true if the operation is successful otherwise false

boolean insert(final int index, final int v)

{

...

}

// delete the element at the position of index in the buffer

// return true if the operation is successful otherwise false

boolean delete(final int index)

{

...

}

// display all of the (valid) elements in the buffer

void println()

{

...

}

}

**Requirements:**

a) Please complete the above implementation of the class IntBuff and keep it as a file IntBuff.java.

b) Please write a main method to perform the following steps based on the class IntBuff.

1. Create an integer buffer by appending the values 51, 7, 89, 106, 2, 0, -18 to an empty one.

2. Display the content of the buffer.

3. Delete the element of the index 3.

4. Display the content of the buffer.

5. Insert the value -50 at the position of index 5.

6. Display the content of the buffer.

2. Please add the following methods to your IntBuff class.

// Performing a linear search for the value of v in the buffer.

public boolean linearSearch(final int v)

{

...

}

// Performing a linear search for the value of v in the buffer.

public boolean binarySearch(final int v)

{

...

}

// Sorting the elements (ascending order) in the buffer using insertion sort.

public void insertionSort()

{

...

}

3. Please write a main method to perform the following tests on your new methods.

1) Create an integer buffer by appending the values 64, 5, 97, 101, 2, 0, -18, 13, 2008 to an empty one.

2) Display the content of the buffer.

3) Use linear search to find the value 0 and display the result.

4) Use linear search to find the value -1 and display the result.

5) Use binary search to find the value 107 and display the result.

6) Use binary search to find the value 13 and display the result.

7) Sort the elements and display the result.

4. Are you able to provide the static version for the above methods? If so, which versions do you think better? the static ones or non-static ones?