

GIT Department of Computer Engineering
CSE 222/505 - Spring 2017
Homework 3
Due date: March 14 2017 – 23:55

SCENARIO:

Q1:

Implement a **myStringBuilder** class which works like the StringBuilder class. Your class will have:

- A single linked list
- append method to append anything
- 3 different toString methods which;
 - Uses indexes and get method
 - Uses iterator
 - Uses toString method of the linked list

Write a main class which;

- Will read 100.000 integers from the **numbers.txt** file
- Uses myStringBuilder to create a string using 3 different toString methods
- Prints their output to **result1.txt**, **result2.txt** and **result3.txt** files
- Analyzes and compares performances of 3 toString methods

Submit your code to moodle. Add detailed information about your test and performance analysis method. Write detailed analysis results in your report.

Q2:

Implement a **reverseToString** method for the SingleLinkedList class which will create a reversed String. Your method must work recursively.

Write a main class to test your method. Submit your code to moodle.

Q3:

Extend AbstractCollection class as myAbstractCollection. Implement an **appendAnything** method for myAbstractCollection class which appends any AbstractCollection object to any other AbstractCollection object by concatenating them.

You will not test your code. Give more detail about this part in your report.

Q4:

Implement a `SingleLinkedList` class which reuses deleted nodes. That is, your class will keep deleted nodes. When required, instead of creating a new node, your class will use one of the deleted nodes. This way your class will do less garbage collection.

Your class will have a `deletedToString` method which creates a `String` of deleted nodes.

Write a main class to test your `SingleLinkedList` class by inserting 100 integers, deleting 50 of them and inserting 100 more integers. (You can read from a file.) Submit your code to moodle.

RESTRICTIONS:

- Don't use any other third part library

GENERAL RULES:

- For any question firstly use course news forum in moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over twenty percent (%20).
- Register [github student pack](#) and create private project and upload your projects into github.
- Your appeals are considered over your github project process.

TECHNICAL RULES:

- Use given CSE222-VM to develop and test your homeworks (your code must be working on CSE222-VM), CSE222-VM download link will be given on Moodle.
- Implement [clean code standarts](#) in your code;
 - o Classes, methods and variables names must be meaningful and related with the functionality.
 - o Your functions and classes must be simple, general, reusable and focus on one topic.
 - o Use standart [java code name conventions](#).

REPORT RULES:

- Add all [javadoc](#) documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.
- You should submit your homework code, javadoc and report to Moodle in a `studentid_hw#.tar.gz` file.
- Use the given homework format including selected parts (for all of 4 parts of code):

Detailed system requirements	x
Usecase diagrams (extra points)	
Class diagrams	x
Other diagrams	
Problem solutions approach	x
Test cases	x
Running command and results	x

GRADING :

- No OOP design : -100
- No interface : -95
- No method overriding : -95
- No error handling : -50
- No inheritance : -95
- No polymorphism : -95
- No javadoc documentation : -50
- No report : -90
- Disobey restrictions : -100
- **Cheating : -200**
- Your solution is evaluated over 100 as your performance.

CONTACT :

Nur Banu Albayrak