

SAUDI GREEN INITIATIVES



CS210 – Data Structures (232) Project 1

Weight (10%)

Due: February, 12 February 2024 (00:00 hrs)

Following the launch of Vision 2030 in 2016, the Kingdom of Saudi Arabia has taken decisive steps towards a more sustainable future. Inaugurated in 2021, SGI unites environmental protection, energy transition and sustainability programs with the overarching aims of offsetting and reducing emissions, increasing the Kingdom's use of clean energy and addressing climate change. SGI is an ambitious national initiative for the Kingdom of Saudi Arabia that aims to combat climate change, improve quality of life and protect the planet for future generations. The first wave of more than 60 initiatives announced in 2021 represented over SAR 700 billion of investments, contributing to the growth of the green economy. In 2022, 17 new environment-focused initiatives were launched. More initiatives will launch in the coming years, scaling up Saudi Arabia's ambition to create a green future.

The Saudi Green Initiative encourages companies, individuals, entrepreneurs and NGOs who are working on sustainability initiatives in Saudi Arabia to invest and help SGI meet its goals. About the Initiative: <https://www.greeninitiatives.gov.sa>

The SGI includes three main initiatives:

1. Reducing Emissions
2. Greening Saudi
3. Protecting Land and Sea

While entities and individuals can help these initiatives in many ways; for the sake of this project, we assume that the investors would be able to invest in these initiatives online. As a member of a 2-student team your job would be to build a SGI simulation (sim) tool which would help investors to plan and invest money in the SGI initiatives. Each team would be responsible to develop back-end of this application (sim tool). You will develop a java software to store investor data and process various requests. Your sim tool reads data as sets of commands from console, in batches, processes this data and computes the total investments for the initiatives and provide necessary output on the console. This batch-processing tool would use an ordered single linked list to implement the various attributes, methods and classes written in the java programming language.

The following is an arbitrary listing of various memberships that are offered by the SGI to investors based on investment amount. Investors would be able to invest in the initiatives separately and the information for investments would be stored in the data structure.

SGI_ID	SGI_name	Gold Member	Silver Member	Bronze Member	Blue Member
1	Reducing Emissions	80000	50000	20000	<20000
2	Greening Saudi	30000	15000	5000	<5000
3	Protecting Land and Sea	50000	20000	10000	<10000

Based on the memberships entities and individuals would be eligible to get benefits by the SGI. Your project implements three classes namely Node, List and Solution. The following provides an API for your project.

Node	Description
<pre>int memberID; int iID; int SGI_ID; int investment; String iName; Node next;</pre>	<pre>// A unique id generated for each investor; // Investor ID number; // SGI ID, takes values between 1 and 3 (inclusive) // Investment amount // Name of the investor // Node next</pre>
<pre>Node() Node(, , ,)</pre>	<pre>// default constructor // override constructor as necessary. You can edit the params as needed.</pre>
<pre>String toString()</pre>	<pre>// returns a String with the contents of the Node as follows memberID iID SGI_ID investment iName</pre>

The following is the API for the List class. This implements a Singly Linked List consisting of Nodes from the Node class.

List	Description
<pre>Node Head; int size;</pre>	<pre>// Anchors the Head of the List // Maintains the size of the List</pre>
<pre>List(); Insert(Node N) void remove(iID) int investment(iID) void addinv(memberID,investment) void print1(iID) void print2(SGI_ID) void print3(memberID)</pre>	<pre>// default constructor, you may have overloaded constructors // inserts the node N in the sorted position according to the investment made by the investor. The largest investor appears at the head of the List. // removes the first occurrence of a node with matching iID. // This method searches for all Nodes with iID and computes the sum of ALL investments for an investor with iID. The return value is rounded off (int rounding). //searches for memberID in the list. Adds investment for the investor and updates their membership status. //searches for iID in the list. Prints ALL nodes with iID. //searches for SGI_ID in the list. Prints ALL nodes with SGI_ID. //searches for memberID in the list. Prints ONE node with memberID.</pre> <p>NOTE: print methods call Node.toString() as necessary.</p>

The following is the API for the Solution class. This class implements the main method, makes appropriate data Input / Output calls and implements data structures and all necessary method calls.

Solution	Description
<pre>//Attributes NA</pre>	<pre>Not applicable</pre>
<pre>main()</pre>	<pre>Implements the main method; reads data from console, processes information and make appropriate calls as necessary.</pre>

Sample Input

Here is a sample input

```
1 1225 101 3 30000 Ahmed Ali
1 1226 103 2 20000 Mohammad Othman
1 1227 101 1 50000 Ahmed Ali
3 101
2 103
4 1227 30000
5 101
```

Sample Output

For the above sample input, the following would be printed on the console.

```
80000
1227 101 1 80000 Ahmed Ali Gold
1225 101 3 30000 Ahmed Ali Silver
```

Explanation

<pre>1 1225 101 3 30000 Ahmed Ali 1 1226 103 2 20000 Mohammad Othman 1 1227 101 1 50000 Ahmed Ali</pre>
<p>Your program reads a set of commands from the console using the following format:</p> <pre><command> <memberID> <iID> <SGI_ID> <investment> <iName></pre> <p>The line starting with integer 1 indicates that insert call would be made. No Output is generated for the insert call.</p>
<pre>3 101</pre>
<p>The line starting with integer 3 indicates that cost method would be called. The program computes the cost for all visits by Visitor with iID = 101. In the list, there are 2 nodes with iID = 101.</p> <p>Node with memberID =1225 has an investment = 30000 Node with memberID =1227 has an investment = 50000</p> <p>Hence total investment = 50000+30000=80000 80000 will be displayed on the console</p>
<pre>2 103</pre>
<p>The line starting with integer 2 indicates that the first node with iID=103 will be removed from the list. No output is generated. This removes the node that contains the following information</p> <pre>1226 103 2 20000 Mohammad Othman</pre>
<pre>4 1227 30000</pre>
<p>The line starting with integer 4 indicates that the node with memberID =1227 will be searched and the given investment of 30000 will be <u>added</u> to the previous investment. No output is generated.</p>

```
5 101
```

The line starting with integer 5 indicates that All visitor details for iID=101 will be printed on console separate by new line character at the end of each line. The membership status would also be calculated and printed for each node.

```
1227 101 1 80000 Ahmed Ali Gold
1225 101 3 30000 Ahmed Ali Silver
```

Similar to the above, your program implements the following commands:

Legend for the acceptable commands:

1. insert details for a new node
2. removes node from list
3. computes the total investments for this iID and displays the total.
4. adds investments and updates it.
5. prints All nodes with iID.
6. prints ALL nodes with SGI_ID.
7. prints the node with memberID.

NOTE: If no output is generated, your program prints 0 on the console.

Evaluation:

You are allowed to work as a group with maximum 2 members in a group. Your work's evaluation would be based the following criterion.

Step 1: Hackerrank

You have unlimited chances to submit your code to hackerrank before the deadline. When you submit your code, the hackerrank platform tests your program against pre-built (hidden) testcases and grades your work. You need to fix all the mistakes in the project and try to earn a full score. For this project, there are no limitations on time and memory usage. Upload your work to <https://www.hackerrank.com/232cs210p1>

Step 2: Upload your submission to LMS. The submission will consist of

- A zip file containing "ALL code"
- A jar file exported from Netbeans
- Upload the [coversheet](#) with personal details and a screenshot from Hackerrank.

The instructor reserves the right to determine the scores of each test case. Test-cases will be posted on hackerrank, students will have unlimited number of opportunities to post and test their project until the due date. The system will not take any submissions after the due date.

Code Inspection and plagiarism:

The code would be inspected by the instructor. The instructor would use the MOSS tool (<https://theory.stanford.edu/~aiken/moss/>) to determine the originality of submission.

If the code similarity is above 50%, the students would earn ZERO score on the project. This includes **all** group members for all teams involved as well (i.e. all other groups with similar code)

Additional Note:

Any Student would be requested to present their work. The instructor reserves the right to “**interview**” any student on their submission to see the understanding of the submission. The instructor may also ask the student to modify the code to satisfy any test-case(s) there in.

This project is weighted 10 points. The score is determined by the correct number of test-cases passed on the hackerrank portal, the code inspection and a possible interview.

Submission Dead-Line:

The submission deadline is final. Late Submissions will be awarded ZERO points.

Important Notes:

- It is the student’s responsibility to check/test/verify/debug the code before submission.
- It is the student’s responsibility to check/test/verify all submitted work (including jar files)
- It is the student’s responsibility to verify that all files have been uploaded to the LMS.
- Incomplete or wrong file types that do not execute will NOT be graded.
- For each project, instructor will provide a few sample test-cases to verify the execution of your program.
- After an assignment/project has been graded, re-submission with an intention to improve an assignments score will not be allowed.
- After the assignment/project has been graded, the instructor will post test-cases used for grading on the website.
- The instructor has the right to share project execution reports that may have been auto-generated on the course website.

TUTORIAL:

Submission on Hacker Rank

Step 1. Register on <https://www.hackerrank.com/>

Make sure your username as your PSU ID number.



I'm here to practice and prepare
Solve problems and learn new skills

Create account

HackerRank

For Developers

Practice coding, prepare for interviews, and get hired.

Sign up

Log in



Muhammad Ahmed



220110999@psu.edu.sa



Your password



I agree to HackerRank's [Terms of Service](#) and [Privacy Policy](#).

Create An Account

2. Join contest "232CS210p1"

Go to <https://www.hackerrank.com/232CS210p1>. Start working on your project.