GIT Department of Computer Engineering
CSE 222/505 - Spring 2017
Homework
Due date: February 26 2017 – 23:55

SCENARIO:

Q1-) Design and implement a simple library management system.

The library holds only books for users to barrow.

The management system's users are library users and library staff. System users will log in to the system as either library users or library staff.

Library users can barrow or return books. Library staff can add a new book or remove a book from library. (Be careful for any unreasonable operation.)

Library staff can register a new library user.

Use csv files to keep records.

Q2-) (Add this part to your report) In the concept of ADT, Java provides encapsulation that ensures the access to data using only specific methods and the development of data structures independently from data type. How can C programming language provide these features?

RESTRICTIONS:

- Can be only one main class in project
- Don't use any other third part library

GENERAL RULES:

- For any question you have firstly use course news forum in moodle, and then the contact TA.
- You can submit assignment one day late with the penalty of 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 homework (your code must run on CSE222-VM), CSE222-VM download link is given on Moodle.
- Implement clean code standards;
    - Classes, methods, and variables names must be meaningful and related with the functionality.
    - Your functions and classes must be simple, general, reusable and focus on one topic.
    - Use standard 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:

Detailed system requirements	x
The Project usecase diagrams (extra points)	x
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 (nbalbayrak@gtu.edu.tr)