Object Oriented Programming Lab (JAVA)
BCSE 2nd year 2nd Semester

Analyze and design each application with the help of class diagram (in all cases) and sequence or interaction diagram (in some special cases) for the following assignments. *Build a package named java_assignments to store all the assignments. Use separate sub package for each assignment.*

Beginners Level

Assignment 1:

Design a STUDENT class to store roll, name, course, admission date and marks in 5 subjects taken from user. Create an array of STUDENT objects. Provide methods corresponding to admission date and receiving marks, preparing mark sheet. Support must be there to show the number of students who have taken admission.

Inherit Student class and override the input method so as to input the department of each student. Search and display a **sorted list** of students of **one department** or **students** based on scoring criteria. Create an arraylist of students and remove a student based on certain criterion and then call **gc()** and check for free memory.

NOTE: Student roll/ID should be fixed length which include department name, admission year and roll and it should be auto generated (no random roll no will be accepted). Consider at least four departments. **1st four characters: dept, 2nd two characters : year, last three characters: roll no**

Example: if a student is in dept. Of Computer Science and Engg.

1st admission : **BCSE*18*001** (considering max 100 students in each dept.)

2nd admission : **BCSE*18*002**

if a student is in dept. Of Electronics and Telecommunication Engg.

1st admission : **ETCE*18*001** (considering max 100 students in each dept.)

3rd admission : **ETCE*18*003**

Assignment 2:

Design a system for the following scenario:

1. An item list contains item code, name, rate, and quantity for several items.
2. Whenever a new item is added in the list uniqueness of item code is to be checked. Register a new product with its price.
3. Time to time rate of the items may change.
4. Whenever an item is issued or received existence of the item is checked and quantity is updated.
5. In case of issue, availability of quantity is also to be checked.
6. User may also like to know price/quantity available for an item.
7. Find how many items cost more than a given amount. The amount will be a parameter.
8. Remember that the methods have to return an error code if for example an invalid item code is given

NOTE:

- The system should be maintained by two types of user, one is Stock entry operator(SEO) and other is Shopkeeper (SK) and SEO will be the first operator in default case.
- The SEO primarily maintain first 3 operations but SEO users can also maintain all operations (1 to 8)
- SK users can only operates on 4 to 8.
- System should be used for a specific shop type. Ex. Electronics, Book, Grocer etc.. You can design your system for any one.
- Item Code should be auto generated that includes item name and entry order(1,2,3...)

Example: for Electronics shop

Item name	entry order	Item Code
Laptop	3	LAP003
Mobile	2	MOB002
Monitor	10	MON010
Mouse	1	MOU001

Assignment 3:
Write a program Parentheses.java that reads in a text stream from standard input and uses a ***stack*** to determine whether or not its parentheses are properly balanced. For example, your program should print true for [()]{}{[()()]()} and false for [(]). You need to implement the ***stack*** class by yourself.

Assignment 4:
Create a class diagram and Java code for the following system and scenario, taking into account the possibility of future extensions. "The system is a command line utility that prints a short 'quote of the day' on the user's terminal when run. To begin with the quote is selected randomly from a set of hard-coded strings within the program itself, but that might change later on -- the quotes might be based on the **user's history, the time of day, the date,** etc..

Scenario

1. User types "java QuoteOfTheDay" on the command line.
2. System prints out a quote of the day, with an attribution.

Examples:

> javac QuoteOfTheDay
God helps them that help themselves. -- Benjamin Franklin
> javac QuoteOfTheDay
Happiness is not a reward - it is consequence. Suffering is not a punishment - it is a result. -- Robert Green Ingersoll
> javac QuoteOfTheDay
Future. That period of time in which our affairs prosper, our friends are true and our happiness is assured. -- Ambrose Bierce
> javac QuoteOfTheDay
Honesty is the first chapter of the book of wisdom. --Thomas Jefferson

Note:

- **user's history:** *within a fixed time duration it will not print any similar type of quote (considering different execution)*
- **the time of day:** *morning, noon evening , night*
- **the date: on 15th August – Independence day, 5th September – Teachers' day**

Assignment 5:
Indexing a book. Write a program that reads in a text file from standard input and compiles an alphabetical index of which ***words/phrases*** appear on which lines, as in the following input. Ignore case and punctuation. For each word maintain a list of location on which it appears. Try to use HashTable and/or HashMap class (of java.util).
Note: key of HashMap : word or phrase
Value of HashMap can be single or multiple(if multiple time occurs)--

Assignment 6:
Design and create a hospital information system with the following scenarios.

- Register a new patient.
- Each patient is assigned to one doctor, but a doctor can have any number of patients. Patients check in to the hospital and assigned a doctor if they don't already have one.
- While in the hospital, doctors record various observations about each patient at various times. Examples of observations are blood pressure and temperature. Record test results for a patient.
- The hospital keeps track of all the observations for a given patient until they check out of the hospital. Obtain all of a patient's information given the social security number.

NOTE: Patients id will be auto generated.

Assignment 7:

Write a program that lists all of the files and subdirectories in a specified directory, along with their sizes and modification dates. By default, the output should be sorted by name. If invoked with the `-s` option, however, output should be sorted by size, from largest to smallest. If invoked with the `-d` option, output should be sorted by date, from most recent to least. Use the `sort( )` method of `java.util.Collections` to help with the sorting.

Assignment 8:

Write a `Compress` class defines two static methods, `gzipFile( )`, which compresses a file using GZIP compression format, and `zipDirectory( )`, which compresses the files (but not directories) in a directory using the ZIP archive and compression format. `gzipFile( )` uses the `GZIPOutputStream` class, while `zipDirectory( )` uses the `ZipOutputStream` and `ZipEntry` classes, all from `java.util.zip`. Use these methods to compress a directory given by the user.