JADAVPUR UNIVERSITY ENGINEERING FACULTY
B.COMP.SC.ENGG-SECOND YEAR
SESSION: 2016 - 2017

Object Oriented Programming using Java
Assignment I

Name: Submission Deadline:20/03/2017

Roll: Group: **A2**

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.

Remarks	Problem
	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.
	2. Design a system for the following scenario: ● An item list contains item code, name, rate, and quantity for several items. ● Whenever a new item is added in the list uniqueness of item code is to be checked. Register a new product with its price. ● Time to time rate of the items may change. ● Whenever an item is issued or received existence of the item is checked and quantity is updated. ● In case of issue, availability of quantity is also to be checked. ● User may also like to know price/quantity available for an item. ● Find how many items cost more than a given amount. The amount will be a parameter. ● Remember that the methods have to return an error code if for example an invalid item code is given
	3. 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 • User types "java QuoteOfTheDay" on the command line. • 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 happinesss is assured. -- Ambrose Bierce

> javac QuoteOfTheDay

Honesty is the first chapter of the book of wisdom. --Thomas Jefferson

4. Design an applet version of the same.

5. Indexing a book. Write a program that reads in a text file from standard input and compiles an alphabetical index of which words 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).

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.

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.

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.