ST. XAVIER'S COLLEGE
KOLKATA
(AUTONOMOUS)

1st SEMESTER EXAMINATION
NOVEMBER - DECEMBER 2016
M.Sc. COMPUTER SCIENCE

CMSM4157 (SET-I)

LABORATORY 2: OBJECT ORIENTED PROGRAMMING LAB

Friday, December 9, 2016
10:00 AM to 02:00 PM
4 hours
Full Marks : **80**

READ THESE INSTRUCTIONS FIRST:

- Of the questions attempted, the answers to only the first required number of questions (as stipulated in the question paper) will be evaluated. **So please do not attempt extra questions**.
- Use fountain pen or ball-point pen of **blue** or **black ink**.
- Answer in your own words as far as practicable.
- Do not write anything on the Question paper other than your Roll No.
- Answer each Group in a separate Answer Script.

At the end of the examination, fasten all your work securely together.

The marks are given in **brackets [ ]** at the end of each question or part question.

Of the questions attempted, the answers to only the first required number of questions (as stipulated in the question paper) will be evaluated.
So, PLEASE DO NOT ATTEMPT EXTRA QUESTIONS.

SET: I

PROGRAM CODE: 45; SAMPLE OUTPUT(S): 15; VIVA: 20

(Only the program code and the sample output(s) are to be written in the answer script.)

(One question to be chosen by random draw.)

1. Write a class to create and initialize an integer array. Write proper methods to sort the elements of the array in the descending order. Implement exception handling mechanisms whenever required.

2. Write an abstract base class to initialize complex numbers. Create proper sub classes to perform the following operations on the complex numbers:
   (a) Addition
   (b) Subtraction
   (c) Multiplication

3. Create a package called Centigrade which will have a class to initialize and display temperature in the centigrade scale. Similarly, create another package called Farenheit which will have a class to initialize and display temperature in the farenheit scale. Both of these packages are to be created in the same path. Now create another class named as Compare in the parent directory of the previously created packages and it will be used to compare the temperature values stored in the previously defined classes and to display suitable messages.

4. (a) Create a new exception class which will throw an exception when any negative number is taken as an input to an integer array.
   (b) Create a class to find the largest string present in the list of strings passed as the command line arguments.

5. Create a class to replace all occurrences of a particular string in a sentence by another string. The sentence will be input by the user and the string to be replaced and the new string should be passed as command line arguments. If the string to be replaced is not present in the input sentence then generate proper user-defined exception.

****

CMSM4157
SET: I

1. Write a class to create and initialize an integer array. Write proper methods to sort the elements of the array in the descending order. Implement exception handling mechanisms whenever required.

CMSM4157
SET: I

2. Write an abstract base class to initialize complex numbers. Create proper sub classes to perform the following operations on the complex numbers:
   (a) Addition
   (b) Subtraction
   (c) Multiplication

CMSM4157
SET: I

3. Create a package called Centigrade which will have a class to initialize and display temperature in the centigrade scale. Similarly, create another package called Farenheit which will have a class to initialize and display temperature in the farenheit scale. Both of these packages are to be created in the same path. Now create another class named as Compare in the parent directory of the previously created packages and it will be used to compare the temperature values stored in the previously defined classes and to display suitable messages.

CMSM4157
SET: I

4. (a) Create a new exception class which will throw an exception when any negative number is taken as an input to an integer array.
   (b) Create a class to find the largest string present in the list of strings passed as the command line arguments.

CMSM4157
SET: I

5. Create a class to replace all occurrences of a particular string in a sentence by another string. The sentence will be input by the user and the string to be replaced and the new string should be passed as command line arguments. If the string to be replaced is not present in the input sentence then generate proper user-defined exception.