CSL 203: OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)

INTERNAL LAB EXAMINATION-SET A

Implement a java program for the following scenario

Define a class with data members ***username, password*** and ***phone_number*** and ***isValid ().*** The function *isValid ()* checks the validity of password and phone number. If the password and phone_number is valid, write the user's name and phone_number into **a file.** If it is not valid, **raise an exception** with appropriate error message indicating why the password or phone_number is invalid in the console.

***isValid ()* method:**

i. length of password should be in between 8 and 16

ii. Password should include a digit and an uppercase letter.

iii. Phone_number should be 10-digit number and all the elements should be digits.

(use the concept of this and constructor overloading)

CSL 203: OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)

INTERNAL LAB EXAMINATION-SET B

Design a base class named ***"Customer***" with attributes "***name"*** and "***phone-number."*** Derive a class called "***Depositor***" from the Customer class, adding attributes "***accno***" and "***balance.***" The Depositor class should contain methods for *depositing and withdrawing* funds.

Further, derive a class named "***Borrower***" from the Depositor class, introducing attributes "***loan-no" and "loan-amt."*** The Borrower class should have a method named "***check_eligibility***" that examines the eligibility for a loan by verifying the **balance**. If the balance is greater than 0, display "Eligible for loan" and write the customer details to **a file**. If the balance is not sufficient, throw a **custom exception.**

In the main method, utilize ***an array of objects*** to manage the list of customers. **Read the details of eligible customers from the file** and display the customers whose names start with the **letter 's' on the console.**

CSL 203: OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)

INTERNAL LAB EXAMINATION-SET C

Implement the following multithread program.

The main function is responsible for **reading the filename from the console.** The input file is expected to contain a **line of integers separated by colons (":").** The program should Three **threads:**

Thread1: This thread should read the file, tokenize the integers, and print the lists of odd and even numbers separately into an "odd.txt" and "even.txt"file.

Thread2: This thread should determine the second smallest and largest numbers in both file and write these values into the "smallest. Text" and "largest.txt" file.

Thread 3: Read "odd.txt" and "even.txt" and write the sorted numbers in "oddsort.txt" and "evensort.txt" file.

CSL 203: OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)

INTERNAL LAB EXAMINATION-SET D

Define a class named "Employee" with the following data members: Employee Name, Department, and Employee ID. Implement an "***ischeck ()"*** method within the class that performs the following tasks:

a. The method should throw an "**IDException**."

b. Return true if the Employee ID is valid, and false otherwise.

c. The method should check for the following errors and throw an "IDException" with the appropriate message:

i. The length of the Employee ID is not equal to 8 characters.

ii. The Employee ID does not start with the department code (e.g., "HR" for Human Resources, "IT" for Information Technology).

iii. Any non-alphanumeric characters are present in the Employee ID.

If the Employee ID is valid, the method should write the name, department, and Employee ID **into a file.** If an "IDException" is thrown, catch it and print the name, Employee ID, and the associated error message indicating why the Employee ID is invalid in the console. **(use the concept of this and constructor overloading)**

CSL 203: OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)

INTERNAL LAB EXAMINATION-SET E

Implement a java program for the following scenario

Define a class with data members Citizen *Name, Country, & Social Security Number (SSN)* and *isValid().* The Social Security number is a nine-digit number in the format *"**AA-GGG-SSSS**". Check* the social security number is valid by using an isValid() method. If the social security number is valid, **write the name, Country and SSN number into a file.** If SocSecException is thrown, it should catch it and print the name, SSN, and the associated error message indicating why the SSN is invalid in the console.

isValid() method:

a. This method throw SocSeException
b. Return true if the SSN number is valid, false otherwise
c. The method checks the following errors and throws a SocSecExceotion with the appropriate message
   i. Number of characters not equal to 11 characters
   ii. Dashes in the wrong spot
   iii. Any non-digit in the SSN