CSN-103: Fundamentals of Object Oriented Programming

Assignment 04

General Instructions:

a) Whenever required, use **Scanner** class to accept inputs for the user at the runtime.
b) To submit this assignment: Create a single .zip file containing all the source code (.java) files. Rename the zip as 04_xxxxxxx where xxxxxxx is your enrollment number.
c) Send the zip file to oop2019.iitr@gmail.com
d) Follow indentation while writing programs.
e) All submissions will be checked for Plagiarism. ANY ATTEMPT TO CHEAT WILL BE SEVERELY PENALIZED.

Programming Problems

1) Write a JAVA program that creates two objects *ob1* and *ob2* of **Integer** *Type **Wrapper***. Store two *int* literals/values inside these **Integer** objects. Define a method **swap()** with two **Integer** parameters and **Integer** return type. Your program should swap *ob1* and *ob2* by calling **swap()** and by using the **Integer** object returned by it. Print the *int* values stored in the **Integer** objects before and after calling the **swap()** method.
   Note: Objects of type wrappers are immutable.

2) Write a JAVA program that accepts five random numbers from the user at the runtime and stores them in an array. Define a method **sortArray()** which accepts the array as an argument and sort the elements of the array in an increasing order. Finally, **return** the sorted array to the **main()** method where array elements are printed in the sorted order.
   Note: You can use any sorting algorithm of your choice (Quick, Bubble, Insertion etc.)

3) Write a Java program that emulates the **Stack** data structure. The program should use an array to store the **stack** elements. Elements can be added (pushed) or removed (popped) from **top** of the stack only. Define methods **push()** and **pop()** to add and remove top **stack** element. Use **public** access specifier for **push()** and **pop()** methods, and **private** access specifier for the array to prevent direct manipulation of **stack** elements.

4) Write a Java program to create a database (using an array) of students admitting in IIT Roorkee.
   a. Create a class **Student** which has three member variables viz., enroll_number (an int), name (a String), and branch (String "EC" or "CS").
   b. Create another class **Database** where the user is asked to enter name and branch only.
   c. Declare two **static** variables in **Student** class, one for EC and another for CS branch. Initialize these static variables with the first enrollment number for each branch i.e. 19116001 for EC and 19114001 for CS.
   d. When user enters the name and branch information at the runtime, an object of **Student** class should be created. The name and branch information are stored in the name and branch member variables of the object. Then, one out of two static variables is incremented (by one) based on whether the branch is EC or CS. The incremented value is then used to set the value of enroll_number for the newly created **Student** object. Finally, the object is stored inside an array to create the database (an array of Student objects).
   e. Display information of all the students saved in the array using **println().**
   f. **Access Specifiers:** Make sure static variables are private and they should be incremented automatically whenever a **Student** object is created.