Student Fees Management System

STEP01: Create a class Student (base class) with the following attributes:

first_name
last_name
previous_std
address
mail_id
mobile_no

Implement the following methods in the Student class:

- A constructor to initialize the attributes (first_name, last_name, previous_std, address, mobile_no). Ensure that the mobile_no entered is a valid 10-digit number. If not, prompt the user to enter a valid mobile number until a valid one is provided.
- A method named generate_mail_id that generates a mail id in the format: {first_name}.{last_name}@ljku.edu.in.
- A method named display to display the student's details, including first_name, last_name, previous_std, address, mail_id, and mobile_no.
- A method named calculate_percentage that takes input marks for subjects (Maths, Science, Social Science, English), calculates the overall percentage, and returns a discount percentage based on the calculated overall percentage. Refer to the following scholarship details for determining the discount percentage:
    - If percentage is greater than or equal to 90, scholarship will be 40% of the fixed fees.
    - If percentage is greater than or equal to 80 and less than 90, scholarship will be 30% of the fixed fees.
    - If percentage is greater than or equal to 70 and less than 80, scholarship will be 20% of the fixed fees.
    - If percentage is greater than or equal to 60 and less than 70, scholarship will be 10% of the fixed fees.
    - If percentage is less than 60, no scholarship will be given.

STEP02: Inherit the classes ScienceStudent, CommerceStudent, and SecondaryStudent from the Student class.

For each child class (ScienceStudent, CommerceStudent, and SecondaryStudent), implement the following:

- Define class variables:
    - For ScienceStudent: science_student_id_counter (starting from 1) and science_fixed_fees (50000).
    - For CommerceStudent: commerce_student_id_counter (starting from 31) and commerce_fixed_fees (40000).

    - For SecondaryStudent: secondary_student_id_counter (starting from 61) and secondary_fixed_fees (30000).
- Implement a constructor that calls the parent class constructor and generates a student id for each category.
- Implement a method named calculate_fees in each child class which will calculate the fees for each category of students. Each category of student has to pay 5% activity charges and 2% stationary charges of their decided fixed fees. The scholarship will be given based on the previous year percentage as per the scholarship details mentioned in STEP01. Display the student details, category, and fees details.
- Override the display method in each child class to display the values of its own instance variables in addition to calling the display method of the Student class.

STEP03: Create a main function to interact with the user.

In the main function:

- Display a menu with the following options:
    - Science Student
    - Commerce Student
    - Secondary Student
    - Exit/Quit
- Accept user input to choose the type of student for which they want admission (use a switch or if-else statements).
- For the chosen category, prompt the user to enter details (first name, last name, previous standard, address, and mobile number).
- Create an instance of the appropriate child class based on the user's choice and call the calculate_fees method to calculate and display the fees.
- Repeat these processes until the user selects the exit/quit option.

Note: Ensure to handle input validation, such as valid choices in the menu and appropriate input formats.