CHRIST (Deemed to be University)
BANGALORE CENTRAL CAMPUS
DEPARTMENT OF COMPUTER SCIENCE
CIA Component -II
COURSE CODE/ COURSE NAME: MCA272 – PROGRAMMING IN JAVA

PROGRAM – II MCA B   Time Duration – 1.30 hrs Marks – 30 Marks

Social Media

Imagine you are tasked with implementing a simplified version of a social media platform in Java. The platform involves different types of users, posts, and comments. The goal is to model these entities using Java classes and demonstrate the use of inheritance, interfaces, packages, and string manipulation.

Requirements:

1. **User Class:**
   - Create an abstract class named `User` with the following attributes:
     - `userId` (String)
     - `userName` (String)
     - `email` (String)
   - Include a parameterized constructor to initialize the attributes.
   - Define an abstract method named `displayProfile()` that prints the user's profile details.
2. **Post Interface:**
   - Create an interface named `Post` with the following methods:
     - `createPost(String content)`: Takes a content string and returns a post ID.
     - `displayPost(String postId)`: Takes a post ID and displays the details of the post.
3. **RegularUser Class:**
   - Create a subclass named `RegularUser` that inherits from `User` and implements the `Post` interface.
   - Include an additional attribute `postList` (ArrayList<String>) to store post IDs.
   - Implement the `displayProfile()` method to show user details.
   - Implement the methods of the `Post` interface to create and display posts.
4. **AdminUser Class:**
   - Create another subclass named `AdminUser` that inherits from `User`.
   - Include an additional attribute `permissions` (String) to store admin permissions.
   - Implement the `displayProfile()` method to show admin user details.
5. **SocialMediaPackage:**
   - Organize the classes and interface into a package named `socialmedia`.
   - Provide appropriate access modifiers to ensure encapsulation.
6. **StringManipulation:**
   - Create a separate Java class outside the `socialmedia` package.

- Implement a method named `manipulateString(String input)` that performs the following:
    - Counts the number of vowels in the input string.
    - Converts the input string to uppercase.
    - Reverses the input string using `StringBuffer`.

Note: You can add additional functions in the classes wherever required.