CLASS TEST 5

QUESTION - 1:

Create a student attendance system to record and manage student attendance. Implement methods to mark attendance, generate attendance reports, and calculate attendance percentages.

Methods:

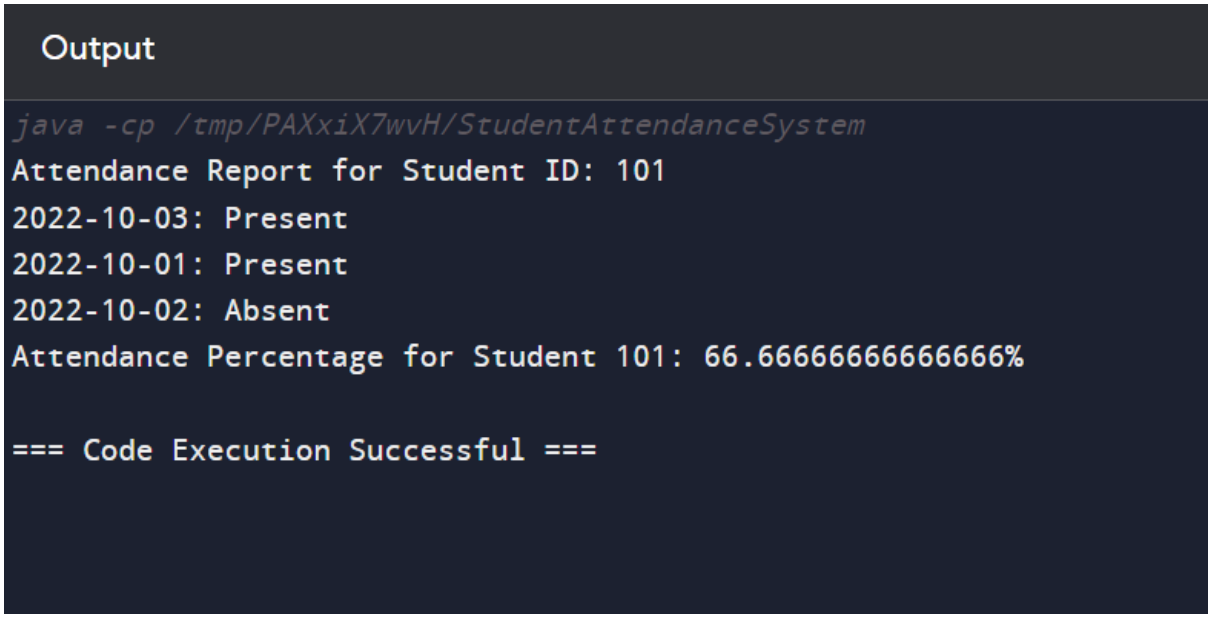
- markAttendance(int studentId, String date, boolean isPresent)
- generateAttendanceReport(int studentId)
- calculateAttendancePercentage(int studentId)

PROGRAM:

```
import java.util.HashMap;

public class StudentAttendanceSystem {

   private HashMap<Integer, HashMap<String, Boolean>> attendanceRecords =
new HashMap<>();


   public void markAttendance(int studentId, String date, boolean isPresent) {

      attendanceRecords.putIfAbsent(studentId, new HashMap<>());

      attendanceRecords.get(studentId).put(date, isPresent);

   }


   public void generateAttendanceReport(int studentId) {

      System.out.println("Attendance Report for Student ID: " + studentId);

      attendanceRecords.getOrDefault(studentId, new
HashMap<>()).forEach((date, isPresent) -> System.out.println(date + ": " +
(isPresent ? "Present" : "Absent")));
```

```
    }
    public double calculateAttendancePercentage(int studentId) {
        if (!attendanceRecords.containsKey(studentId) ||
attendanceRecords.get(studentId).isEmpty()) {
            return 0.0;
        }
        long totalDays = attendanceRecords.get(studentId).size();
        long presentDays =
attendanceRecords.get(studentId).values().stream().filter(Boolean::booleanVal
ue).count();
        return (double) presentDays / totalDays * 100;
    }
    public static void main(String[] args) {
        StudentAttendanceSystem system = new StudentAttendanceSystem();



        system.markAttendance(101, "2022-10-01", true);
        system.markAttendance(101, "2022-10-02", false);
        system.markAttendance(101, "2022-10-03", true);


        system.generateAttendanceReport(101);


        System.out.println("Attendance Percentage for Student 101: " +
system.calculateAttendancePercentage(101) + "%");
    }
}
```

OUTPUT:



QUESTION - 2:

Develop a weather forecast application that fetches and displays weather information. Implement methods to get current weather, forecast for the week, and display weather details.

Methods:

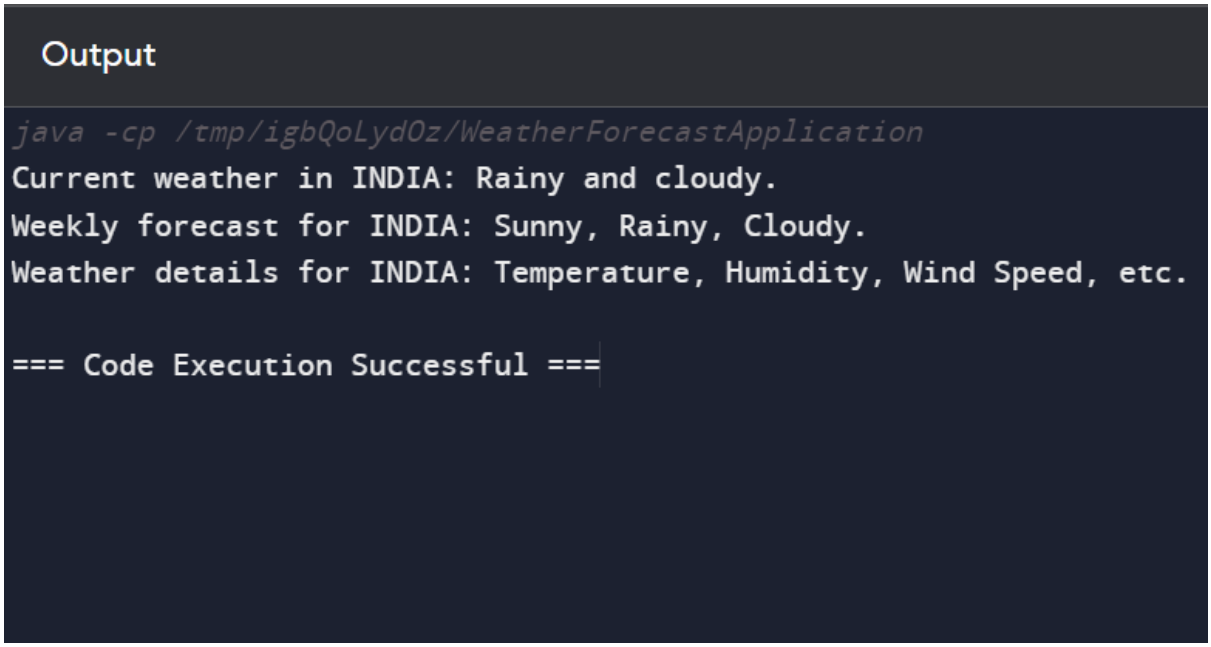
- getCurrentWeather(String location)
- getWeeklyForecast(String location)
- displayWeatherDetails(String location)

PROGRAM:

```
public class WeatherForecastApplication {
    public static void main(String[] args) {
        WeatherForecastApplication App = new
WeatherForecastApplication();
        String location = "INDIA";


        App.getCurrentWeather(location);
        App.getWeeklyForecast(location);
        App.displayWeatherDetails(location);
    }


    public void getCurrentWeather(String location) {
        System.out.println("Current weather in " + location + ": Rainy
and cloudy.");
    }


    public void getWeeklyForecast(String location) {
        System.out.println("Weekly forecast for " + location + ": Sunny,
Rainy, Cloudy.");
    }


    public void displayWeatherDetails(String location) {
        System.out.println("Weather details for " + location + ":
Temperature, Humidity, Wind Speed, etc.");
```

```
    }
}
```

OUTPUT:


Output
java -cp /tmp/igbQoLydOz/WeatherForecastApplication
Current weather in INDIA: Rainy and cloudy.
Weekly forecast for INDIA: Sunny, Rainy, Cloudy.
Weather details for INDIA: Temperature, Humidity, Wind Speed, etc.
=== Code Execution Successful ===