Q1)

Create a Java project called Lab6-Q1. Inside this project create a class named "Q1" with a main function. Also create a class named "Patient". The "Patient" class should have the following instance variables:

private String name;
private double weight;
private int age;

The "Patient" class should have the following accessor methods:

public String getName() - returns the patient's name
public double getWeight() - returns the patient's weight
public int getAge() - return the patient's age

and the following method

public boolean patientEquals(Patient otherPatient) - Returns a true bool if the two patients are identical. False otherwise

Create 3 instances of a patient. Make 2 of the 3 patients identical, and 1 patient unique.

In the main function use the Accessor methods to print the patients attributes. Then use the *patientEquals()* to compare if the patients are identical or different and print the results.

Ex)

Standard Output: Patient 1: Ryan Tannehill is 220.0lbs and is 26 years old

Standard Output: Patient 2: Ryan Tannehill is 220.0lbs and is 26 years old

Standard Output: Patient 3: Jarvis Landry is 200.0lbs and is 25 years old

Standard Output: Ryan Tannehill is the same as Ryan Tannehill

Standard Output: Ryan Tannehill is different from Jarvis Landry

Q2)

Create a Java project called Lab6-Q2. Inside this project create a class named "Q2" with a main function. Also create a class named "Car". The "Car" class should have the following instance variables:

private String name;
private double horsePower;
private double weight;

Create the appropriate accessor methods.

Create the below methods.

public void readInput() - This method asks the user for the name of the car, the horse power of the car, and the weight of the car. The users input is assigned to the instance variables.

public String whoFaster(Car otherCar) - This method will compute which car is faster and return the name of the faster car. Which ever car has the higher ratio of horsepower/weight is the faster car.

Create two instances of different cars and use the readInput() method to assign the cars information. Then call the whoFaster() method and print which car is faster.

Ex)

Standard Output: What is the car's name?

Standard Input: Mustang

Standard Output: What is the car's horsepower?

Standard Input: 300

Standard Output: What is the car's weight?

Standard Input: 2000

Standard Output: What is the car's name?

Standard Input: Civic

Standard Output: What is the car's horsepower?

Standard Input: 100

Standard Output: What is the car's weight?

Standard Input: 2000

Standard Output: The Mustang is the faster car

Q3)

Consider a class RatingScore that represents a numeric rating for something such as a movie. Its attributes are

-The description of what is being rated
-The maximum possible rating
-The rating

It will have methods to

-Get the rating from a user
-Return the maximum rating possible
-Return the rating
-Return a string showing the rating in a format suitable for display

Note: The program should have logic to handle the case if the user enters an input greater then the max possible rating.

Write a driver program that displays the functionality of the program.

hint

RatingScore movieRating = new RatingScore();
movieRating.initialize("Joe's excellent movie", 5);

RatingScore restaurantRating = new RatingScore();
restaurantRating.initialize("Food quality", 10);

Ex)

Standard Output: What is your rating for Joe's excellent movie?
Standard Output: Please enter an integer from 0 to 5

Standard Input: 3

Standard Output: Displaying the rating for the movie.
Standard Output: The rating is 3/5 for Joe's excellent movie

Standard Output: What is your rating for Food quality?
Standard Output: Please enter an integer from 0 to 10

Standard Input: 9

Standard Output: Displaying the rating for the restaurant.
Standard Output: The rating is 9/10 for Food quality