Assignment 1

**Question One. Total Marks /10**

**As a programmer, you are tasked to develop an object-oriented simulation of a heating system in**

**a smart home using the &quot;Room&quot; and &quot;Radiator&quot; classes.**

**The &quot;Radiator&quot; class should have the following features: Marks /4**

** RadiatorID and isOn member variables.**

** A default constructor that assigns a unique RadiatorID value (incremented by 15 for each**

**new object) with starting value on the last 3 digits of the student ID &quot;22K – 1345&quot; (e.g.**

**345). You are required to input your student ID and extract the digits.**

** The radiator starts in the off state.**

** An accessor method for RadiatorID.**

** A public method called &quot;heats&quot; that links the Room object with the Radiator object.**

**The &quot;Room&quot; class should have the following features: Marks /4**

** roomName, seatingCapacity, and numRadiators member variables.**

** A constructor that accepts roomName and sets seatingCapacity to a default value of 12.**

** A public method &quot;isHeatedBy&quot; that links the Radiator object with the Room object and**

**returns a message indicating the status of the radiator integration. This method returns a**

**String containing one of the following messages indicating whether the Radiator was**

**successfully added or not:**

**o &quot;Radiator already added to room.”**

**o &quot;Radiator successfully added to room.”**

**o &quot;Cannot add Radiator. Room has a maximum number of radiators.&quot;**

** A Room can be heated by 0 to 2 unique radiators.**

**In the main program do the following: Marks /2**

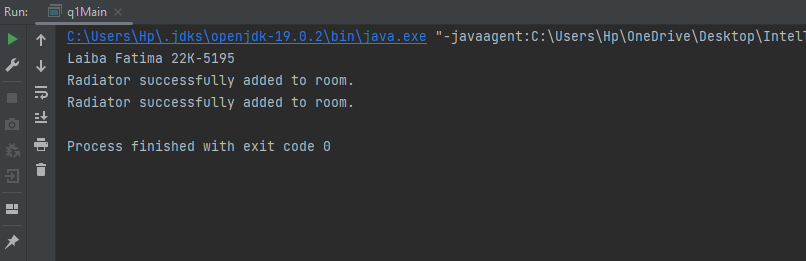
**Create a &quot;Room&quot; object &quot;room&quot; and two &quot;Radiator&quot; objects &quot;rd1&quot; and &quot;rd2&quot; and link them such**

**that the &quot;Room&quot; object &quot;room&quot; is heated by both &quot;Radiator&quot; objects.**

**Code:**

public class q1Main {  
 public static void main(String[] args) {  
 System.*out*.println("Laiba Fatima 22K-5195");  
 Room room = new Room("Drawing Room");  
 Radiator rd1 = new Radiator();  
 Radiator rd2 = new Radiator();  
 System.*out*.println(room.isHeatedBy(rd1));  
 System.*out*.println(room.isHeatedBy(rd2));  
 }  
  
  
}  
  
 class Radiator{  
 private int RadiatorID;  
 boolean isOn;  
 public static int *nextID* = 195;  
 public Radiator() {  
 this.RadiatorID = *nextID*;  
 *nextID* = *nextID* + 15;  
 this.isOn = false;  
 }  
 public int getRadiatorID() {  
 return RadiatorID;  
 }  
  
 public void heats (Room room){  
 room.isHeatedBy(this);  
 }  
 }  
  
 class Room{  
 String roomName;  
 int seatingCapacty;  
 int numRadiators;  
 Radiator r1;  
 Radiator r2;  
  
 public Room(){}  
 public Room(String roomName) {  
 this.roomName = roomName;  
 this.seatingCapacty = 12;  
 this.numRadiators = 0;  
 this.r1 = null;  
 this.r2 = null;  
 }  
  
 public String isHeatedBy(Radiator r){  
 if (r1 == r || r2 == r){  
 return "Radiator already added to room.";  
 }  
 if (numRadiators>=2){  
 return "Cannot add Radiator. Room has a maximum number of radiators.";  
 }  
 if (r1 == null) {  
 r1 = r;  
 numRadiators++;  
 return "Radiator successfully added to room.";  
 }  
 if (r2 == null) {  
 r2 = r;  
 numRadiators++;  
 return "Radiator successfully added to room.";  
 }  
 return "Cannot add Radiator. Room has a maximum number of radiators.";  
  
 }  
 }

**Output:**

****

**Question Two. Total Marks /10**

**A data science company is trying to make a platform DataHub (such as StackOverflow) where**

**different data scientists can post their coding problems related to data science only. Other data**

**scientists can help in solving their problems. You are hired as an initial developer and your task**

**is to create and object-oriented paradigm for the DataHub in the following manner:**

**DataScientist Class: Marks /5**

**Your application must keep track of Data Scientist details including their first name, last name,**

**highest education, age, country, number of answers given, and number of questions asked.**

**Every user is going to have a unique ID which is issued automatically on their object creation,**

**and the generated ID will be used throughout the program for that particular object. The ID will**

**be generated using the second and the last digit of your student ID. You are required to**

**input your student ID and extract the digits. The class must have:**

**● A parameterized constructor that initializes the value of every Data Scientist user.**

**● These data members are read-only: number of answers given and number of questions asked.**

**● These data members can be accessed or updated at any given time: first name, last name, age,**

**and country.**

**● You may set the highest education but can’t access it.**

**● The class must have a member function AskQuestion in which any given problem is posted**

**by the user and it should increase the count of the number of questions asked by that user**

**automatically.**

**● The class must have a member function AnswerProblem in which the user solved any**

**queries posted by others and it should increase the count of the number of answers given by**

**that user automatically.**

**[Please note posting questions and posting answers are just used to increase the count right**

**now. No mechanisms are required for saving those questions or answers.]**

**Admin Class: Marks /3**

**Admin details include their first name, last name, age, country, and admin ID.**

**Admin class also keeps track of every new user created. A new user creation must increment the**

**user count known by only Admin Class. Admin class must have**

**● A parameterized constructor that initializes the value of admin and also assigns AdminID.**

**● These data members can be accessed or updated at any given time: first name, last name, age,**

**and country.**

**● The class must have a member function Total User. Calling this function will show the total**

**count of data scientists created but only the admin knows the total count.**

**In the main program do the following: Marks /2**

**Implement the above scenario by creating objects. Show the functionalities of both the classes by**

**invoking the functions appropriately.**

**Code:**

public class q2Main {  
 public static void main(String[] args) {  
 System.*out*.println("Laiba Fatima 22K-5195");  
 Admin admin = new Admin("laiba", "fatima", 19, "Pakistan");  
 DataScientist s1 = new DataScientist("Hani", "Masood", "Montessori", 65,"Nepal");  
 DataScientist s2 = new DataScientist("Umer", "Sheikh", "Matric", 56 , "Bangladesh");  
  
 s1.AskQuestion();  
 s1.AnswerProblem();  
 s2.AnswerProblem();  
 s2.AnswerProblem();  
 s2.AnswerProblem();  
  
 System.*out*.println(s1);  
 System.*out*.println(s2);  
  
 s1.display();  
 s2.display();  
  
 admin.TotalUser();  
 }  
}  
class DataScientist{  
 private String firstname;  
 private String lastname;  
 private String highesteducation;  
 private int age;  
 private String country;  
 private int numOfAnswersGiven = 0 ;  
 private int numOfQuestionsAsked = 0;  
 public static int *uniqueID*;  
  
 public DataScientist(String firstname, String lastname, String highesteducation, int age, String country) {  
 this.firstname = firstname;  
 this.lastname = lastname;  
 this.highesteducation = highesteducation;  
 this.age = age;  
 this.country = country;  
 this.numOfAnswersGiven = 0;  
 this.numOfQuestionsAsked = 0;  
 int studentid = 5195;  
 this.*uniqueID* = (((5195/100)%10)\*10) + (5195%10);  
 *uniqueID*++;  
 }  
  
 void display(){  
 System.*out*.println("First name " +firstname);  
 System.*out*.println("Second name " +lastname);  
 System.*out*.println("Highest education " + highesteducation);  
 System.*out*.println("age " +age);  
 System.*out*.println("Country " + country);  
 }  
 public String getFirstname() {  
 return firstname;  
 }  
  
 public String getLastname() {  
 return lastname;  
 }  
  
 public String getHighesteducation() {  
 return highesteducation;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public int getNumOfAnswersGiven() {  
 return numOfAnswersGiven;  
 }  
  
 public int getNumOfQuestionsAsked() {  
 return numOfQuestionsAsked;  
 }  
  
 public static int getUniqueID() {  
 return *uniqueID*;  
 }  
  
 public void setFirstname(String firstname) {  
 this.firstname = firstname;  
 }  
  
 public void setLastname(String lastname) {  
 this.lastname = lastname;  
 }  
  
 public void setHighesteducation(String highesteducation) {  
 this.highesteducation = highesteducation;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
  
 public void setNumOfAnswersGiven(int numOfAnswersGiven) {  
 this.numOfAnswersGiven = numOfAnswersGiven;  
 }  
  
 public void setNumOfQuestionsAsked(int numOfQuestionsAsked) {  
 this.numOfQuestionsAsked = numOfQuestionsAsked;  
 }  
  
 public void AskQuestion() {  
 numOfQuestionsAsked++;  
 }  
 public void AnswerProblem() {  
 numOfAnswersGiven++;  
 }  
}  
class Admin{  
 private String firstname;  
 private String lastname;  
 private int age;  
 private String country;  
 public static int *adminID*;  
 int userCount;  
  
 public Admin(String firstname, String lastname, int age, String country) {  
 this.firstname = firstname;  
 this.lastname = lastname;  
 this.age = age;  
 this.country = country;  
 this.userCount = 0;  
  
 this.*adminID* = 5195;  
 this.*adminID* = (((5195/100)%10)\*10) + (5195%10);  
 }  
  
 public String getFirstname() {  
 return firstname;  
 }  
  
 public String getLastname() {  
 return lastname;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public String getCountry() {  
 return country;  
 }  
  
 public void setFirstname(String firstname) {  
 this.firstname = firstname;  
 }  
  
 public void setLastname(String lastname) {  
 this.lastname = lastname;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public void setCountry(String country) {  
 this.country = country;  
 }  
 public void addUser() {  
 userCount++;  
 }  
  
 public int TotalUser() {  
 return userCount;

}  
  
}

**Output:**

****

**Question Three. Total Marks /10**

**A smartwatch is a wearable computer in the form of a watch; modern smartwatches provide a**

**local touchscreen interface for daily use and calculate your health essentials.**

**While exercising, you can use a heart-rate monitor installed in your smartwatch to see that your**

**heart rate stays within a safe range.**

**The formula for calculating your maximum heart rate in beats per minute is 220 minus your**

**age in years. Your target heart rate is a range that is 50–85% of your maximum heart rate.**

**Your task is to create a class called HeartRates.**

**Marks /2**

** The class attributes should include the person’s first name, a unique ID, and date of birth**

**(consisting of separate attributes for the month, day, and year of birth). The ID will be**

**the third digit of your student ID. You are required to input your student ID and**

**extract the third digit.**

** Your class should have a constructor that receives this data as a parameter. For each**

**attribute, provide set and get functions.**

**The class has the following functions: Marks /6**

**o a function getAge that calculates and returns the person’s age (in years),**

**o a function getMaxiumumHeartRate that calculates and returns the person’s**

**maximum heart rate**

**o a function getTargetHeartRate that calculates and returns the person’s target heart**

**rate.**

**Marks /2**

** Write an application that prompts for the person’s information, instantiates an object of**

**the class HeartRates, and prints the information from that object including the person’s**

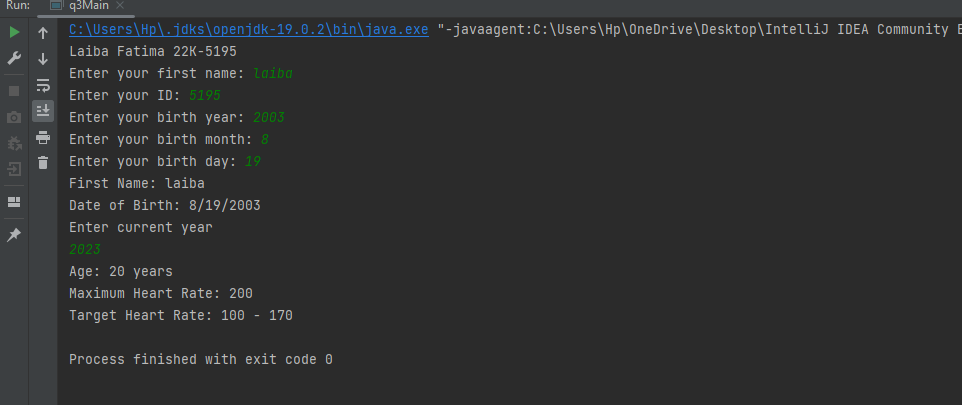
**first name, and date of birth—then calculates and prints the person’s age in (years),**

**maximum heart rate and target-heart-rate range.**

**Code:**

import java.util.Scanner;  
  
public class q3Main {  
 public static void main(String[] args) {  
 System.*out*.println("Laiba Fatima 22K-5195");  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter your first name: ");  
 String firstName = scanner.nextLine();  
  
 System.*out*.print("Enter your ID: ");  
 int id = scanner.nextInt();  
  
 id = (id / 10) % 10;  
  
 System.*out*.print("Enter your birth year: ");  
 int birthYear = scanner.nextInt();  
  
 System.*out*.print("Enter your birth month: ");  
 int birthMonth = scanner.nextInt();  
  
 System.*out*.print("Enter your birth day: ");  
 int birthDay = scanner.nextInt();  
  
 HeartRates heartRates = new HeartRates(firstName, id, birthYear, birthMonth, birthDay);  
  
 System.*out*.println("First Name: " + heartRates.getFirstName());  
 System.*out*.println("Date of Birth: " + heartRates.getBirthMonth() + "/" + heartRates.getBirthDay() + "/" + heartRates.getBirthYear());  
 System.*out*.println("Enter current year");  
 int curryear;  
 curryear = scanner.nextInt();  
 System.*out*.println("Age: " + heartRates.getAge(curryear, birthYear) + " years");  
 int age;  
 age = heartRates.getAge(curryear, birthYear);  
 System.*out*.println("Maximum Heart Rate: " + heartRates.getMaximumHeartRate(age));  
 int maximumHeartRate = heartRates.getMaximumHeartRate(age);  
 int[] targetRange = heartRates.getTargetHeartRate(maximumHeartRate);  
 System.*out*.println("Target Heart Rate: " + targetRange[0] + " - " + targetRange[1]);  
 }  
}  
  
//formula for maximum heart rate = 220 - age(years)  
//target heart rate = 50-80% of max heart rate  
class HeartRates {  
 Scanner A = new Scanner(System.*in*);  
 private String firstName;  
 private int id;  
 private int birthYear;  
 private int birthMonth;  
 private int birthDay;  
  
 public HeartRates(String firstName, int id, int birthYear, int birthMonth, int birthDay) {  
 this.firstName = firstName;  
 this.id = id;  
 this.birthYear = birthYear;  
 this.birthMonth = birthMonth;  
 this.birthDay = birthDay;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public int getBirthYear() {  
 return birthYear;  
 }  
  
 public void setBirthYear(int birthYear) {  
 this.birthYear = birthYear;  
 }  
  
 public int getBirthMonth() {  
 return birthMonth;  
 }  
  
 public void setBirthMonth(int birthMonth) {  
 this.birthMonth = birthMonth;  
 }  
  
 public int getBirthDay() {  
 return birthDay;  
 }  
  
 public void setBirthDay(int birthDay) {  
 this.birthDay = birthDay;  
 }  
  
 public int getAge(int currentyear,int birthyear) {  
  
 int age = currentyear - birthyear;  
 return age;  
 }  
  
 public int getMaximumHeartRate(int a) {  
 return 220 - a;  
 }  
  
 public int[] getTargetHeartRate(int m) {  
 int targetMin = (int) Math.*round*(m \* 0.5);  
 int targetMax = (int) Math.*round*(m \* 0.85);  
 int[] targetRange = {targetMin, targetMax};  
 return targetRange;  
 }  
}

**Output:**

****

**Question Four. Total Marks /10**

**Atrium Cinemas need to automate their system to keep track of the tickets sold for the movies**

**that are screened at the cinema halls.**

**Create a class named as “Movie”. Marks /3**

**● The class contains member variables that include movie name and showtime.**

**● Users can only see which movies are being shown and at what times. Include member**

**functions that display what movies are being shown and their showtimes.**

**Create a class named as “Ticket”. Marks /4**

**● The class contains member variables that include the row number, seat number, ticketID,**

**and a boolean / flag variable that maintains the status of whether the seat has been sold or**

**not.Status (boolean / flag) is a private member variable.**

**● The ticketID will be the first two digits of your studentID. Generate unique integer values**

**for the ticketID using the first 2 digits of your student ID “22K – 1345” so 13 would be**

**considered as the ticketID. You are required to input your student ID and extract the**

**digits.**

**● Include member functions that check if the ticket has been sold, update the ticket status to**

**sold and a display function that displays the sold status, row number, ticketID and the**

**seat number for the sold ticket.**

**In the main program do the following: Marks /3**

**Implement the above scenario by creating objects. Show the functionalities of both the classes**

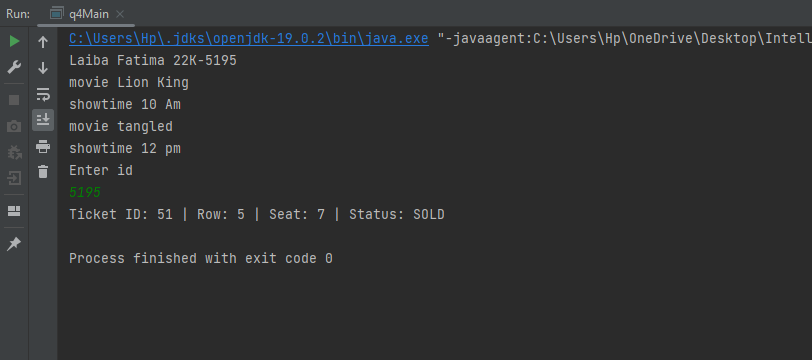
**where a user can buy a ticket for a particular movie. Additional functions and variables can be**

**added as needed.**

**Code:**

import java.util.Scanner;  
  
public class q4Main {  
 public static void main(String[] args) {  
 System.*out*.println("Laiba Fatima 22K-5195");  
 Scanner A = new Scanner(System.*in*);  
 Movie m1 = new Movie("Lion King", "10 Am");  
 Movie m2 = new Movie("tangled", "12 pm");  
 m1.display();  
 m2.display();  
 int id;  
 System.*out*.println("Enter id");  
 id = A.nextInt();  
 int digits = id /100;  
  
 Ticket ticket = new Ticket(5,7,digits);  
  
 if(!ticket.isSold()){  
 ticket.sellTicket();;  
 ticket.displayTicket();  
 }  
 }  
}  
class Movie{  
  
 private String moviename;  
 private String showtime;  
  
 public Movie(String name, String showTime) {  
 this.moviename = name;  
 this.showtime = showTime;  
 }  
 void display(){  
 System.*out*.println("movie " +moviename);  
 System.*out*.println("showtime "+showtime);  
 }  
 public String getName() {  
 return moviename;  
 }  
  
 public String getShowTime() {  
 return showtime;  
 }  
}  
class Ticket {  
 Scanner A= new Scanner(System.*in*);  
 int rowNumber;  
 int seatNumber;  
 int ticketID;  
 private boolean isSold;  
  
 public Ticket(int rowNumber, int seatNumber, int ticketID) {  
 this.rowNumber = rowNumber;  
 this.seatNumber = seatNumber;  
 this.ticketID = ticketID;  
 this.isSold = false;  
 }  
 public boolean isSold() {  
 return isSold;  
 }  
  
 public void sellTicket() {  
 this.isSold = true;  
 }  
 public void displayTicket() {  
 System.*out*.println("Ticket ID: " + ticketID + " | Row: " + rowNumber + " | Seat: " + seatNumber + " | Status: " + (isSold ? "SOLD" : "AVAILABLE"));  
 }  
}

**Output:**

****

**Question Five. Total Marks /10**

**Let’s go over a very simple scenario:**

**Scenario: An application dedicated to providing the right shoes for its users**

**(But we’ll skip the searching part for now)**

**Let’s consider the User class for the application. Each person(user) can be represented by a**

**specific set of attributes, for example, their user ID, their name, their age, their height, their**

**gender, their shoe size, etc. Each person has some actions they can perform on the application.**

**One such function is notifying whether the user is an infant, toddler, child, teenager or adult**

**(We’ll assume that the parents handle the accounts for children under 13). Furthermore, the user**

**attributes can be accessed and updated by the application by some functionality provided by the**

**user class.**

**Similarly, a Shoe can be represented by its size, width, style (running shoes, sneakers, stiletto,**

**etc.), brand, color and demographic (infant, toddler, child, adult). Similar to the user class, the**

**attributes for any shoe can be accessed and updated by the functionality provided by the shoe**

**class.**

**Your task for this question is to create an application that asks the user to enter their information.**

**Your program should create an object for the user and store the information inside the object**

**using the functionality that you have created for these classes.**

**Afterwards, your program should ask the user about which type of shoes they want to purchase.**

**Once again, after you have taken the input from the user, you should create an object for the shoe**

**class, and store values in that object.**

**Your program should have the following checks for ages, when assigning the value to the**

**demographic attribute in the shoe object:**

**· infant: ages 0 -2 years (0 assumes that infant is some months old)**

**· toddler: 3-5 years**

**· child: 10-12 years**

**· teenager: 13-19 years**

**· adult: 19+ years**

**Following things are required for this question:**

**1. Marks /2**

**User and Shoe classes (along with all of the attributes and functions). User ID should be an**

**integer array of length 2. It should hold your roll number within its indices in this manner: If your**

**roll number is 12k-2034, it should be stored in the array as: {12, 2034}**

**2. Marks /2**

**Default and Parameterized constructors for these classes**

**3. Marks /1**

**Accessor and Mutator functions for these classes**

**For questions 4-6, provide a menu-driven interface for the user.**

**4. Marks /2**

**Objects for User and Shoe should be created in your main function. This should be done by using**

**user provided values.**

**5. Marks /2**

**Users should be allowed to update the information that they have provided as many times as they**

**would like. Values should always be updated using the setter functions.**

**6. Marks /1**

**Users should be allowed to view the updated details as many times as they would like. Values**

**should always be retrieved by using the getter functions.**

**7. Marks /???**

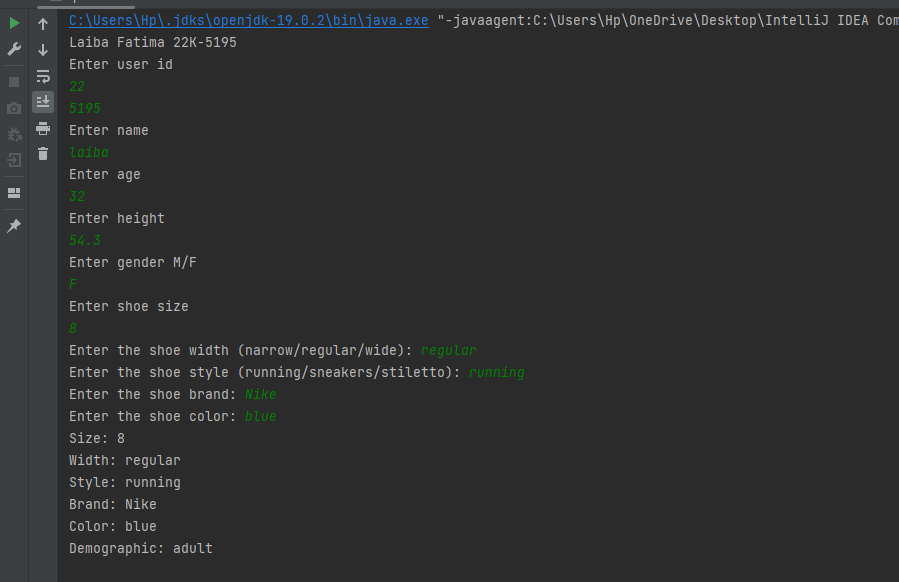
**Bonus(?): Create a global function that accepts a User and Shoe Object and displays all their**

**information by using the getter functions.**

**Code:**

import java.util.Scanner;  
public class q5Main{  
public static void main(String[]args){  
 Scanner A = new Scanner(System.*in*);  
 int i = 0;  
 int[] userID= new int[2];  
 System.*out*.println("Laiba Fatima 22K-5195");  
 System.*out*.println("Enter user id");  
 int[] userId = { A.nextInt(), A.nextInt() };  
 A.nextLine();  
 System.*out*.println("Enter name");  
 String name = A.nextLine();  
 System.*out*.println("Enter age");  
 int age = A.nextInt();  
 System.*out*.println("Enter height");  
 double height = A.nextDouble();  
 System.*out*.println("Enter gender M/F ");  
 String gender = A.next();  
  
 System.*out*.println("Enter shoe size");  
 int shoeSize = A.nextInt();  
  
 User user = new User(userId, name, age, height, gender, shoeSize);  
  
 A.nextLine();  
 System.*out*.print("Enter the shoe width (narrow/regular/wide): ");  
 String shoeWidth = A.nextLine();  
 System.*out*.print("Enter the shoe style (running/sneakers/stiletto): ");  
 String shoeStyle = A.nextLine();  
 System.*out*.print("Enter the shoe brand: ");  
 String shoeBrand = A.nextLine();  
 System.*out*.print("Enter the shoe color: ");  
 String shoeColor = A.nextLine();  
  
  
 Shoe shoe = new Shoe(shoeSize, shoeWidth, shoeStyle, shoeBrand, shoeColor, user.getDemographic());  
 shoe.displayShoeInfo(shoe);  
 }  
}  
class User {  
 private int[] userId;  
 private String name;  
 private int age;  
 private double height;  
 private String gender;  
 private int shoeSize;  
  
 public User( ) {  
 this.userId = new int[2];  
 this.name = "";  
 this.age = 0;  
 this.height = 0.0;  
 this.gender = "";  
 this.shoeSize = 0;  
 }  
 public User(int[] userId, String name, int age, double height, String gender, int shoeSize) {  
 this.userId = userId;  
 this.name = name;  
 this.age = age;  
 this.height = height;  
 this.gender = gender;  
 this.shoeSize = shoeSize;  
 }  
  
 public int[] getUserId() {  
 return userId;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public double getHeight() {  
 return height;  
 }  
  
 public String getGender() {  
 return gender;  
 }  
  
 public int getShoeSize() {  
 return shoeSize;  
 }  
  
 public void setUserId(int[] userId) {  
 this.userId = userId;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public void setHeight(double height) {  
 this.height = height;  
 }  
  
 public void setGender(String gender) {  
 this.gender = gender;  
 }  
  
 public void setShoeSize(int shoeSize) {  
 this.shoeSize = shoeSize;  
 }  
 public String getDemographic() {  
 if (age <= 2) {  
 return "infant";  
 } else if (age >= 3 && age <= 5) {  
 return "toddler";  
 } else if (age >= 10 && age <= 12) {  
 return "child";  
 } else if (age >= 13 && age <= 19) {  
 return "teenager";  
 } else {  
 return "adult";  
 }  
 }  
  
}  
  
class Shoe {  
 private int size;  
 private String width;  
 private String style;  
 private String brand;  
 private String color;  
 private String Demographic;  
  
 public Shoe() {  
 size = 0;  
 width = "";  
 style = "";  
 brand = "";  
 color = "";  
 Demographic = "";  
 }  
  
 public Shoe(int size, String width, String style, String brand, String color, String demographic) {  
 this.size = size;  
 this.width = width;  
 this.style = style;  
 this.brand = brand;  
 this.color = color;  
 this.Demographic = demographic;  
 }  
  
 public int getSize() {  
 return size;  
 }  
  
 public String getWidth() {  
 return width;  
 }  
  
 public String getStyle() {  
 return style;  
 }  
  
 public String getBrand() {  
 return brand;  
 }  
  
 public String getColor() {  
 return color;  
 }  
  
 public String getDemographic() {  
 return Demographic;  
 }  
  
 public void setSize(int size) {  
 this.size = size;  
 }  
  
 public void setWidth(String width) {  
 this.width = width;  
 }  
  
 public void setStyle(String style) {  
 this.style = style;  
 }  
  
 public void setBrand(String brand) {  
 this.brand = brand;  
 }  
  
 public void setColor(String color) {  
 this.color = color;  
 }  
  
 public void setDemographic(int age) {  
 if (age >= 0 && age <= 2) {  
 Demographic = "infant";  
 } else if (age >= 3 && age <= 5) {  
 Demographic = "toddler";  
 } else if (age >= 10 && age <= 12) {  
 Demographic = "child";  
 } else if (age >= 13 && age <= 19) {  
 Demographic = "teenager";  
 } else {  
 Demographic = "adult";  
 }  
 }  
 public void displayShoeInfo(Shoe shoe) {  
 System.*out*.println("Size: " + shoe.getSize());  
 System.*out*.println("Width: " + shoe.getWidth());  
 System.*out*.println("Style: " + shoe.getStyle());  
 System.*out*.println("Brand: " + shoe.getBrand());  
 System.*out*.println("Color: " + shoe.getColor());  
 System.*out*.println("Demographic: " + shoe.getDemographic());  
 }  
  
  
}

**Output:**

****