Lesson – 3

Reading work: Do the reading of concepts discussed in this chapter.

Problem: 1

Create your own Customer class. A Customer has a firstName, lastName, socSecurityNum (which you can represent as a String), also it has billingAddress and a shippingAddress(which you can represent as a type of Address. Initialize billingAddress and shippingAddress through its setter from Customer class.
Create a constructor for your Customer class to initialize firstName, lastName and socSecurityNum. Create getter, and setter methods for all five attributes.
Create an Address class with the attributes such as street, city, state and zip(which you can represent as a String). Create a constructor to initialize fields of Address class.
Your Customer class should have a toString() method that provides a string representation of the customer. A typical toString() output would be "[Joe, Smith, ssn: 332-221-4444]". Just copy this code in your Customer class.

```
public String toString() {
        return "[" + firstName + ", " + lastName + ", " + "ssn: " + socSecurityNum
+ "]";
    }
```

In the main method of a Main class, create three instances of Customer (be sure to create instances of Address to populate their billingAddress and shippingAddress fields using setters). Add these instances to an array. Then loop through the array and print to the console all those Customers whose billingAddress is located in the city of Chicago (when you create instances of Customer initially, be sure to create at least one Customer whose billing address is in Chicago).

Problem – 2

(Target-Heart-Rate Calculator) While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the

American Heart Association (AHA), the formula for calculating your Maximum Heart Rate (MHR) in beats per minute is 220 minus your age in years. Your target heart rate is a range that's 50–85% of your maximum heart rate.

[Formula: To Calculate Target heart rate range is mentioned below]

a. Assign Resting Heart Rate (RHR) = 70. (Assign as a Constant)
b. Assign Maximum Heart Rate (MHR) = 220 – age.
c. Calculate Average Heart Rate (AHR) = MHR – RHR.
d. Assign Lower Boundary (LB) = 0.5 (Assign as a Constant)
e. Assign Upper Boundary (UB) = 0.85. (Assign as a Constant)
f. Calculate Lower Boundary Target Heart Rate (LBTHR)= (AHR*LB) + RHR,
g. Calculate Upper Boundary Target Heart Rate (UBTHR) = (AHR*UB) + RHR
h. The Result of Target Heart Rate Range is between LBTHR and UBTHR

Create a class called HeartRates. The class attributes should include the person's first name, last name and date of birth. Your class should have a constructor that receives this data as parameters. For each attribute provide set and get methods. The class also should include

a. A method that calculates and returns the person's age (in years), (Refer lesson3\dateapi\AgeCalculator.java)
b. A method that calculates and returns the person's maximum heart rate.
c. A method that calculates and prints the person's target heart rate range.
d. Override the toString() method to display the person's first name, last name and date of birth, person's age in (years), maximum heart rate(MHR).

Write a Java app that prompts for the person's information, instantiates an object of class HeartRates and prints the required information.

Expected Output

```
Please, enter your first name:
Tom
Please, enter your last name:
Bruck
Please, enter your birth date in the format-(yyyy-mm-dd)-Example 1989-4-14
1999-5-15
The Target Heart Rate Range is between 134.5 and 179.64999999999998
First Name : Tom
Last Name : Bruck
Age :21
Date of Birth : 1999-05-15
Maximum Heart Rate: 199
```

Problem 3:

Develop a Java application that helps a global team schedule meetings across different time zones. The application should allow users to create an event by specifying a date and time, view the event in a different time zone, and calculate the days until the event. Nicely show the date and time display.

Requirements:

1. Create an Event to the input the details and print the requested details:
    a. Allow users to input the event name, date and time for a new event.
    b. Display the day of the week (Eg: Sunday, Monday) for the event date and check if it is in a leap year.
    c. Calculate and display the number of days from the current date to the event date.
2. Write a function to Format Event Details.
    a. A Function, nicely format and display the event date and time, including the default system time zone.
    Eg: Sunday, October 20, 2024 @ 10:30 [America/Chicago]
3. Time Zone Conversion function:
    a. Convert the event's date and time from the system's default time zone to any other, specified time zone entered by the user.

Sample Time Zones
America/Panama
America/Chicago
America/Indiana/Indianapolis
America/Santiago
America/Phoenix

Problem 4: Pizza Order Management with Enums and Switch Expressions

A) Define an enum named **PizzaSize** representing different pizza sizes (SMALL, MEDIUM, LARGE).
B) Define an enum named **PizzaType** representing different pizza types (VEGGIE, PEPPERONI, CHEEZE, BBQCHICKEN).
C) Create a Class
    a. An attribute for the pizza size of type PizzaSize.
    b. An attribute for the pizza type of type PizzaType.
    c. An attribute for the quantity
    d. An attribute to calculate the price based on PizzaSize and PizzaType using calculatePrice() private method
    e. A constructor to initialize PizzaSize, PizzaType and quantity.

D) Create a private void calculatePrice() method which assign the price instance using Switch expression and enum types. You can assign some default values based on the pizza sizes (SMALL($8), MEDIUM($10), LARGE($12) using Switch expression to return the sizeprice. In the same way another switch return the default price based on the types of Pizza(VEGGIE($1), PEPPERONI($2), CHEEZE($1.5), BBQCHICKEN($2)). Finally assign the price instance by adding the sizeprice and typeprice multiplied by quantity.

E) A method printOrderSummary() that returns a string summarizing the pizza order, including size, type, qty, price, tax amount, and total price. Use String.format() to print the receipt as mentioned below.

Pizza Order:
Size: SMALL
Type: VEGGIE
Qty: 2
Price: $18.00
Tax: $0.54
Total Price: $18.54

Add 3% of tax from the price calculated from the Task D. Total Price is price + tax.

F) Use the below code

```
public class PizzaTest {
   public static void main(String[] args) {
      Pizza pizza1 = new Pizza(PizzaSize.SMALL, PizzaType.VEGGIE,2);
      Pizza pizza2 = new Pizza(PizzaSize.MEDIUM, PizzaType.PEPPERONI,1);
      Pizza pizza3 = new Pizza(PizzaSize.LARGE, PizzaType.BBQ_CHICKEN,2);
```

```
        System.out.println(pizza1.printOrderSummary());
        System.out.println(pizza2.printOrderSummary());
        System.out.println(pizza3.printOrderSummary());
    }
}
```

Problem – 5

Reference : ImmutableDemo.java

Create Java classes for Triangle, Rectangle, and Circle. Provide each class with a method

public double computeArea()

Make all of these classes immutable. (Follow the guidelines in the slides for creating this type of class.) Provide one constructor for each class; the constructor should accept the data necessary to specify the figure, and to compute its area. The values accepted by the constructor should be stored in (private) instance fields of the class. For example, Rectangle should have instance fields width and length, and the constructor should look like this

public Rectangle(double width, double length)

For Triangle, you may use arguments base and height. And for Circle, use radius as the constructor argument.

Whenever you create instance fields for one of these classes, provide public accessors for them (but do not provide mutators since the class is supposed to be immutable – for instance, the dimensions of a Rectangle should be read-only). For example, you will have in the Rectangle class:

```
private double width;

public double getWidth() {
        return width;
}
```

Create a fourth class Main that will, in its main method, test these three figure classes as follows: It will create one instance of each (you can make your own choice for the dimensions of your figures and get the input from the console) and then print to the

console the area of each. Typical output would be:

A Sample Output looks like this:

Enter C for Circle
Enter R for Rectangle
Enter T for Triangle
R
Enter the width of the Rectangle
120
Enter the height of the Rectangle
200
The area of Rectangle is: 24000.00
Do you want to continue(y/n) : y
Enter C for Circle
Enter R for Rectangle
Enter T for Triangle
R
Enter the width of the Rectangle
20
Enter the height of the Rectangle
20
The area of Rectangle is : 400.00

Here are some area formulas, in case you do not remember them:

Area of a rectangle = width * height
Area of a triangle = 1/2 * base * height
Area of a circle = PI * radius * radius