Assignment 2

Name: Priyanshu Tripathi

Roll number: 12022005

Subject: Java Programming

Code: CSPC-23

INDEX

1. Valid Identifiers

Which of the following are valid Java identifiers?

a. `Remainder`
b. `9_multiple`
c. `addTwoIntegers`
d. `PI_VALUE`
e. `highestof3`
f. `sum&product`
g. `diff_&_div`

> `Remainder`

> `addTwoIntegers`

> `PI_VALUE`

> `highestof3`

2. Display HI in large block letters

Write a program that displays the word “HI” in large block letters. Make each large letter out of the other character as shown below:

```
 I I I        I I I       H H H H H H H H H
 I I I        I I I                 H
 I I I        I I I                 H
 I I I        I I I                 H
 I I I I I I  I I I                 H
 I I I        I I I                 H
 I I I        I I I                 H
 I I I        I I I                 H
 I I I        I I I       H H H H H H H H H
```

```java
public class foobar {

    public static void main(String[] args) {
        for(int i = 0; i < 9; i++)
            switch (i) {
                case 0:
                case 8: System.out.println("III       III     HHHHHHHHH");
                        break;
                case 4: System.out.println("IIIIIIIIIIIII         H");
                        break;
                default:
                        System.out.println("III       III         H");
            }
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "2. HI.java"
III       III     HHHHHHHHH
III       III         H
III       III         H
III       III         H
IIIIIIIIIIIII         H
III       III         H
III       III         H
III       III         H
III       III     HHHHHHHHH
```

3. Metric Converter

Write a program that reads values representing the weight in kilograms, grams, and milligrams and then prints the equivalent weight in milligrams. (For example, 1 kilogram, 50 grams, and 42 milligrams is equivalent to 10,50,042 milligrams.)

```java
import java.util.Scanner;

public class foobar {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the weight --> ");
        int kg = scanner.nextInt();
        int g = scanner.nextInt();
        int mg = scanner.nextInt();

        String string = "";

        if (kg > 0)
            string += kg > 1 ? kg + " kilograms, " : kg + " kilogram, ";
        if (g > 0)
            string += g > 1 ? g + " grams, " : g + " gram, ";
        if (mg > 0)
            string += mg > 1 ? mg + " milligrams " : mg + " milligram ";

        System.out.println(String.format("%sis equivalent to %d milligrams", string,
convertToMg(kg, g, mg)));
    }

    public static int convertToMg(int kg, int g, int mg) {
        // Changes the value to 0 if entered value is negative
        kg = kg < 0 ? 0 : kg*1000000;
        g = g < 0 ? 0 : g*1000;
        mg = mg < 0 ? 0 : mg;

        return kg + g + mg;
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "3. Converter.java"
Enter the weight --> 1 50 42
1 kilogram, 50 grams, 42 milligrams is equivalent to 1050042 milligrams
[getpsyched@Manjaro Assignment 2]$ java "3. Converter.java"
Enter the weight --> -13 0 78
78 milligrams is equivalent to 78 milligrams
[getpsyched@Manjaro Assignment 2]$ java "3. Converter.java"
Enter the weight --> 0 21 69
21 grams, 69 milligrams is equivalent to 21069 milligrams
```

Ignores negative and 0 values ^

4. Value of coins

Write a program that determines the value of the coins in a jar and prints the total in dollars and cents. Read integer values that represent the number of quarters, dimes, nickels, and pennies.

```java
import java.util.Scanner;

public class foobar {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of quarters, dimes, nickels and pennies --> ");
        int quarters = scanner.nextInt();
        int dimes = scanner.nextInt();
        int nickels = scanner.nextInt();
        int pennies = scanner.nextInt();

        int cents = 0;
        cents += quarters < 0 ? 0 : quarters*25;
        cents += dimes < 0 ? 0 : dimes*10;
        cents += nickels < 0 ? 0 : nickels*5;
        cents += pennies < 0 ? 0 : pennies;

        int dollars = cents/100;
        cents -= dollars*100;

        System.out.println("\nValue --> " + dollars  + " dollars, " + cents + " cents");
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "4. Coins.java"
Enter the number of quarters, dimes, nickels and pennies --> 21 39 2 216

Value --> 11 dollars, 41 cents
[getpsyched@Manjaro Assignment 2]$ java "4. Coins.java"
Enter the number of quarters, dimes, nickels and pennies --> -12 -32 0 15

Value --> 0 dollars, 15 cents
```

Ignores negative and 0 values ^

5. Commas to colons

Write a line of code that changes all commas(,) in a String names sentence into colons(:). Store the modified String in a new variable named **`corrected`**.

```java
import java.util.Scanner;

public class foobar {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the string --> ");
        String string = scanner.nextLine();

        System.out.println("\nCorrected string --> " + correctString(string));
    }

    public static String correctString(String s) {
        String corrected = "";
        for (int i = 0; i < s.length(); i++)
            corrected += s.charAt(i) == ',' ? ':' : s.charAt(i);
        return corrected;
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "5. Commas to colons.java"
Enter the string --> Hello, World!

Corrected string --> Hello: World!
[getpsyched@Manjaro Assignment 2]$ java "5. Commas to colons.java"
Enter the string --> He said: "Hello, World!"

Corrected string --> He said: "Hello: World!"
```

6. Determine the range

Assuming that a Random object has been created called generator, what is the range of the result of each of the following expressions?

a. `generator.nextInt(20)`
b. `generator.nextInt(8) + 1`
c. `generator.nextInt(12) + 2`
d. `generator.nextInt(35) + 10`
e. `generator.nextInt(100) – 50`

a) Range: 0 to 19

b) Range: 1 to 8

c) Range: 2 to 13

d) Range: 10 to 44

e) Range: -50 to 49

7. Random class

Write code to declare and instantiate an object of the `Random` class (call the object reference variable `rand`). Then write a list of expressions using the `nextInt` method that generates random numbers in the following specified ranges, including the endpoints. Use the version of the `nextInt` method that accepts a single integer parameter.

a. `0 to 10`
b. `0 to 400`
c. `1 to 10`
d. `1 to 400`
e. `25 to 50`
f. `-10 to 15`

```java
import java.util.Random;

public class foobar {
    public static void main(String[] args) {
        Random rand = new Random();

        System.out.println(rand.nextInt(11));
        System.out.println(rand.nextInt(401));
        System.out.println(rand.nextInt(10) + 1);
        System.out.println(rand.nextInt(400) + 1);
        System.out.println(rand.nextInt(26) + 25);
        System.out.println(rand.nextInt(26) - 10);
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "7. Random.java"
0
79
2
126
44
2
[getpsyched@Manjaro Assignment 2]$ java "7. Random.java"
1
83
3
41
38
1
```

8. Password Generation

Write a program that generates a random password that meets certain criteria. The password should start with 7 or 8 or 9, and the next five digits can be any digit from 0 to 9. This should be followed by a dash (-) and then three random uppercase letters. Hint: The integers from 65 to 90 represent the uppercase letters from A to Z. You can cast an integer to a `char` type like this: `(char)65` is 'A' and `(char)66` is 'B'

```java
import java.util.Random;

public class foobar {
    public static void main(String[] args) {
        String password = "";
        Random rand = new Random();

        password += rand.nextInt(3) + 7;
        for(int i = 0; i < 5; i++)
            password += rand.nextInt(10);
        password += "-";
        for(int i = 0; i < 5; i++)
            password += (char)(rand.nextInt(27) + 65);

        System.out.println("Password --> " + password);
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 729647-ODTCE
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 923709-XCLSU
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 926496-NGMUP
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 991300-QFZBJ
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 763095-YSGHN
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 729538-YUYYE
[getpsyched@Manjaro Assignment 2]$ java "8. Password Generation.java"
Password --> 747875-QFPDE
```

9. Trigonometric values of generated number

Write a program that generates a random integer in the range 20 to 40, inclusive, and displays the sine, cosine, and tangent of that number.

```java
import java.util.Random;
import java.lang.Math;

public class foobar {
    public static void main(String[] args) {
        Random rand = new Random();
        int number = rand.nextInt(21) + 20;

        double sin = Math.sin(number);
        double cos = Math.cos(number);
        double tan = Math.tan(number);

        System.out.println("Generated number is --> " + number);
        System.out.println("Sine --> " + sin);
        System.out.println("Cosine --> " + cos);
        System.out.println("Tangent --> " + tan);
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "9. Trigonometric values.java"
Generated number is --> 23
Sine --> -0.8462204041751706
Cosine --> -0.5328330203333975
Tangent --> 1.5881530833912738
[getpsyched@Manjaro Assignment 2]$ java "9. Trigonometric values.java"
Generated number is --> 35
Sine --> -0.428182669496151
Cosine --> -0.9036922050915067
Tangent --> 0.473814720414451
[getpsyched@Manjaro Assignment 2]$ java "9. Trigonometric values.java"
Generated number is --> 20
Sine --> 0.9129452507276277
Cosine --> 0.40808206181339196
Tangent --> 2.237160944224742
```

10. Number of goals

Write a class called `NumberOfGoals` that represents the total number of goals scored by a football team. The `NumberOfGoals` class should contain a single integer as instance `data`, representing the number of goals scored. Write a constructor to initialize the number of goals to zero. Write a method called `setGoal` that increments the value by one whenever a goal is scored, and another method called `getGoal` that returns the total number of goals scored so far. Finally, create a driver class called `GoalTracker` that creates a few `NumberOfGoals` objects and tests their methods.

```
public class GoalTracker {
    public static void main(String[] args) {
        NumberOfGoals obj1 = new NumberOfGoals();
        NumberOfGoals obj2 = new NumberOfGoals();

        System.out.println("Initial values --> " + obj1.getGoal() + " " + obj2.getGoal());
        obj1.setGoal(); obj2.setGoal();
        System.out.println("After 1st iteration --> " + obj1.getGoal() + " " +
obj2.getGoal());
        obj1.setGoal();
        System.out.println("After 2nd iteration --> " + obj1.getGoal() + " " +
obj2.getGoal());
        obj2.setGoal(); obj2.setGoal();
        System.out.println("After 3rd iteration --> " + obj1.getGoal() + " " +
obj2.getGoal());
    }
}

class NumberOfGoals {
    int goals;

    NumberOfGoals() { goals = 0; }
    void setGoal() { goals++; }
    int getGoal() { return goals; }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "10. Goals.java"
Initial values --> 0 0
After 1st iteration --> 1 1
After 2nd iteration --> 2 1
After 3rd iteration --> 2 3
```

11. Duration test

Write a class called `Duration` that represents a duration of time in hours and minutes. It should contain instance data that represents the start hour, end hour, start minute, and end minute. Define the Duration constructor to accept and initialize all instance data. Include getter and setter methods for all instance data. Include a `toString` method that returns a String in the format [HH:MM,HH:MM]. Write a method called length that returns the length of the duration in minutes. Assume that a 24-hour format is used for all durations of time and no duration will span over multiple days. Create a driver class called `DurationTest`, whose main method instantiates and updates several Duration objects.

```
public class DurationTest {
    public static void main(String[] args) {
        Duration dr = new Duration(02, 50, 4, 50);
        assert dr.length() == 120;
        for (int i = 0; i < 24; i++) {
            Duration dri = new Duration(0, 0, i, 50);
            assert dri.length() == i * 60 + 50;
            dri.setEndHr(i / 2);
            assert dri.length() == i / 2 * 60 + 50;
            dri.setEndMin(i);
            assert dri.length() == i / 2 * 60 + i;
            dri.setStartHr(i / 3);
            assert dri.length() == i / 2 * 60 + i - i / 3 * 60;
        }
    }
}

class Duration {
    int startHr;
    int startMin;
    int endHr;
    int endMin;

    public Duration(int startHr, int startMin, int endHr, int endMin) {
        this.startHr = startHr;
        this.startMin = startMin;
        this.endHr = endHr;
        this.endMin = endMin;
    }

    public int length() {
        return 60 * (endHr - startHr) + (endMin - startMin);
    }

    @Override
    public String toString() {
        return String.format("[%02d:%02d, %02d:%02d]", startHr, startMin, endHr, endMin);
    }

    public int getStartHr() { return startHr; }

    public int getStartMin() { return startMin; }

    public void setStartHr(int startHr) { this.startHr = startHr; }
```

```java
    public void setStartMin(int startMin) { this.startMin = startMin; }

    public int getEndHr() { return endHr; }

    public int getEndMin() { return endMin; }

    public void setEndHr(int endHr) { this.endHr = endHr; }

    public void setEndMin(int endMin) { this.endMin = endMin; }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java -ea "11. Duration.java"
[getpsyched@Manjaro Assignment 2]$
```

No AssertionError raised

12. Guessing game

Write a program that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive), then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit. Use a sentinel value to determine whether the user wants to quit. Count the number of guesses and report that value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop.

```
import java.util.Random;
import java.util.Scanner;

public class foobar {
    public static void main(String[] args) {
        char playing = 'y';
        Scanner sc = new Scanner(System.in);

        while (playing == 'y') {
            playGame(sc);
            System.out.print("Do you want to keep playing? [y/n] ");
            playing = sc.next().charAt(0);
        }
    }

    static void playGame(Scanner sc) {
        char quit;
        int answer = new Random().nextInt(101);
        int guess, tries = 0;
        do {
            System.out.print("Choose a number --> ");
            guess = sc.nextInt();
            if (guess == answer) {
                System.out.println("You won in " + tries + " tries!");
                break;
            }
            if (guess < answer)
                System.out.println("You guess is too low");
            else if (guess > answer)
                System.out.println("You guess is too high");

            System.out.print("Do you want to quit? [y/n] ");
            quit = sc.next().charAt(0);
            tries++;
        } while (quit == 'n');
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "12. Guessing Game.java"
Choose a number --> 50
You guess is too high
Do you want to quit? [y/n] n
Choose a number --> 25
You guess is too low
Do you want to quit? [y/n] n
Choose a number --> 38
You guess is too high
Do you want to quit? [y/n] n
Choose a number --> 32
You guess is too low
Do you want to quit? [y/n] n
Choose a number --> 35
You guess is too high
Do you want to quit? [y/n] n
Choose a number --> 34
You guess is too high
Do you want to quit? [y/n] n
Choose a number --> 33
You won in 6 tries!
Do you want to keep playing? [y/n] n
[getpsyched@Manjaro Assignment 2]$
```

13. Rock paper scissor

Write a program that plays the Rock-Paper-Scissors game against the computer. When played between two people, each person picks one of three options (usually shown by a hand gesture) at the same time, and a winner is determined. In the game, Rock beats Scissors, Scissors beats Paper, and Paper beats Rock. The program should randomly choose one of the three options (without revealing it), then prompt for the user's selection. At that point, the program reveals both choices and prints a statement indicating if the user won, the computer won, or if it was a tie. Continue playing until the user chooses to stop, then print the number of user wins, losses, and ties.

```
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

public class foobar {
    public static void main(String[] args) {
        char playing = 'y';
        Scanner sc = new Scanner(System.in);

        int lost = 0;
        int won = 0;
        int tie = 0;
        while (playing == 'y') {
            switch (playGame(sc)) {
                case 1: won++;
                        break;
                case -1:
                        lost++;
                        break;
                case 0: tie++;
                        break;
            }
            System.out.print("\nDo you want to keep playing? [y/n] ");
            playing = sc.next().charAt(0);
            System.out.println();
        }

        System.out.println("Final stats:" +
        "\nWon  - " +   won +
        "\nLost - " + lost +
        "\nTie  - " + tie + "\n");
    }

    static String[] choices = {"rock", "paper", "scissor"};

    static int playGame(Scanner sc) {
        int myChoice = new Random().nextInt(3);

        System.out.print("Choose [rock/paper/scissor] --> ");
        String userChoice = sc.next();

        int userChoiceInt = Arrays.asList(choices).indexOf(userChoice.toLowerCase());
        if (userChoiceInt == -1) {
            System.out.println("Invalid Choice!");
            return -1;
        }
```

```
        System.out.print("I choose " + choices[myChoice] + ", ");

        switch (userChoiceInt - myChoice) {
            case 1:
            case -2:
                    System.out.println("You win!");
                    return 1;

            case -1:
            case 2: System.out.println("You lose");
                    return -1;

            case 0: System.out.println("Tie");
                    return 0;
            default:
                    return 0;
        }
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "13. Rock Paper Scissor.java"
Choose [rock/paper/scissor] --> paper
I choose scissor, You lose

Do you want to keep playing? [y/n] y

Choose [rock/paper/scissor] --> rock
I choose paper, You lose

Do you want to keep playing? [y/n] y

Choose [rock/paper/scissor] --> paper
I choose paper, Tie

Do you want to keep playing? [y/n] y

Choose [rock/paper/scissor] --> scissor
I choose rock, You lose

Do you want to keep playing? [y/n] y

Choose [rock/paper/scissor] --> hello
Invalid Choice!

Do you want to keep playing? [y/n] y

Choose [rock/paper/scissor] --> rock
I choose scissor, You win!

Do you want to keep playing? [y/n] n

Final stats:
Won  - 1
Lost - 4
Tie  - 1
```

14. Count the occurrence of vowels in a string

Write a program that reads a string from the user, then determines and prints how many of each lowercase vowel (a, e, i, o, and u) appear in the entire string. Have a separate counter for each vowel. Also count and print the number of non-vowel characters.

```
import java.util.*;

public class vowels {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a string --> ");
        String input = sc.nextLine();
        sc.close();

        int a = 0;
        int e = 0;
        int i = 0;
        int o = 0;
        int u = 0;

        int nonVowel = 0;

        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);
            switch (ch) {
                case 'a':
                        a++;
                        break;
                case 'e':
                        e++;
                        break;
                case 'i':
                        i++;
                        break;
                case 'o':
                        o++;
                        break;
                case 'u':
                        u++;
                        break;

                case 'A':
                case 'E':
                case 'I':
                case 'O':
                case 'U':
                        break;

                default:
                        if (Character.isLetter(ch))
                            nonVowel++;
            }
        }
        System.out.println("a = " + a);
        System.out.println("e = " + e);
```

```
        System.out.println("i = " + i);
        System.out.println("o = " + o);
        System.out.println("u = " + u);
        System.out.println("Non vowels = " + nonVowel);
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "14. Vowels.java"
Enter a string --> Hello, world!
a = 0
e = 1
i = 0
o = 2
u = 0
Non vowels = 7
```

15. Pad

Write a method called pad that accepts a string called `str` and an integer called `length`. Return a string with length number of characters by adding an appropriate number of '`0`' characters to the end of the string. If the string is longer than length, then return the original string unchanged. For example, pad ("`aaa`",6) will return "`aaa000`".

```java
import java.util.*;

public class foobar {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a string --> ");
        String input = sc.nextLine();
        System.out.print("Enter length --> ");
        int length = sc.nextInt();
        sc.close();

        String padded = pad(input, length);
        System.out.println("Padded String is --> " + padded);
    }

    public static String pad(String s, int length) {
        if (length < s.length())
            return s;

        return s + "0".repeat(length - s.length());
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "15. Pad.java"
Enter a string --> hello
Enter length --> 10
Padded String is --> hello00000
[getpsyched@Manjaro Assignment 2]$ java "15. Pad.java"
Enter a string --> world12
Enter length --> 12
Padded String is --> world1200000
```

16. Find the sum of even numbers

Write a program that reads an integer value and prints the sum of all even integers between 2 and the input value, inclusive. Print an error message if the input value is less than 2. Prompt accordingly.

```java
import java.util.Scanner;

public class foobar {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number --> ");
        int num = sc.nextInt();
        sc.close();

        if (num < 2) {
            System.out.println("Invalid number!");
            return;
        }

        int sum = 0;
        for (int i = 2; i <= num; i += 2)
            sum += i;

        System.out.println("Sum --> " + sum);
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "16. Sum of Even.java"
Enter a number --> 2
Sum --> 2
[getpsyched@Manjaro Assignment 2]$ java "16. Sum of Even.java"
Enter a number --> 4
Sum --> 6
[getpsyched@Manjaro Assignment 2]$ java "16. Sum of Even.java"
Enter a number --> 10
Sum --> 30
[getpsyched@Manjaro Assignment 2]$ java "16. Sum of Even.java"
Enter a number --> 0
Invalid number!
[getpsyched@Manjaro Assignment 2]$ java "16. Sum of Even.java"
Enter a number --> -6
Invalid number!
```

17. Splice

Write a method called splice that takes a String and two integers named `startingIndex` and `deletelength`. Remove `deleteLength` characters from the String, starting with index `startingIndex`. Return the resulting string. If `startingIndex` is less than 0, then start removing characters from index 0. If `deleteLength` is 0, then return the original string.

```java
import java.util.Scanner;


public class foobar {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a string --> ");
        String str = sc.nextLine();
        System.out.print("Enter start index --> ");
        int startingIndex = sc.nextInt();
        System.out.print("Enter delete count --> ");
        int deleteLength = sc.nextInt();

        String string = splice(str, startingIndex, deleteLength);

        System.out.println("Result is " + string);
    }

    public static String splice(String str, int start, int del) {
        if(start < 0)
            start = 0;
        return str.substring(0, start) + str.substring(start + del);
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java "17. Splice.java"
Enter a string --> hello, world
Enter start index --> 2
Enter delete count --> 4
Result is he world
[getpsyched@Manjaro Assignment 2]$ java "17. Splice.java"
Enter a string --> hello, world
Enter start index --> 5
Enter delete count --> 3
Result is helloorld
[getpsyched@Manjaro Assignment 2]$ java "17. Splice.java"
Enter a string --> hello, world
Enter start index --> -2
Enter delete count --> 9
Result is rld
```

18. Sales person

Create a class called `SalesPerson` that represents a salesperson in an organization. The `SalesPerson` class should have the name, the phone number, and the assigned district of a salesperson. Each salesperson acquires a certain sale amount every day. Provide a constructor that sets all instance values based on parameter values. Overload the constructor such that each daily sale amount for a week is set to zero. Provide a method called `setDailyAmount` that accepts a day of a week as a number (0 for Sunday, 1 for Monday, and so on) and an amount and sets that amount as the daily sale amount. Provide another method called `getDailyAmount` that accepts a day of a week and returns the amount for that day. Provide a method called total that calculates the total sale amount for the week a method called average that calculates the average daily sale amount for the week. Write a `toString` method that prints all the details of the salesperson. Write a `toString` method that prints all the details of the salesperson. Write a driver class to exercise the `SalesPerson` class.

```
import java.util.*;

public class foobar {
    public static void main(String[] args) {
        SalesPerson sp = new SalesPerson("someName", "9999999999", "someDistrict");
        sp.setDailyAmount(0, 1000);
        sp.setDailyAmount(1, 2000);
        sp.setDailyAmount(2, 3000);
        sp.setDailyAmount(3, 4000);
        sp.setDailyAmount(4, 5000);

        assert sp.getDailyAmount(2) == 3000;
        assert sp.getDailyAmount(4) == 5000;

        assert sp.total() == 15000;
        assert (int)sp.avg() == (int)(15000/7);

        System.out.println("All tests passed");
    }
}

class SalesPerson {
    String name;
    String phoneNum;
    String district;
    int[] sales;

    public SalesPerson(String name, String phoneNum, String district) {
        this.name = name;
        this.phoneNum = phoneNum;
        this.district = district;
        sales = new int[] {0, 0, 0, 0, 0, 0, 0};
    }

    public int getDailyAmount(int day) { return sales[day]; }

    public void setDailyAmount(int day, int amt) { this.sales[day] = amt; }

    public int total() {
        int sum = 0;
```

```
        for (int val : sales)
            sum += val;
        return sum;
    }

    public double avg() { return total() / 7.0; }

    @Override
    public String toString() {
        return "\nName --> " + name + "\nDistrict --> " + district +
        "\nPhone --> " + phoneNum + "\nTotal sales --> " + total();
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java -ea "18. Sales Person.java"
All tests passed
[getpsyched@Manjaro Assignment 2]$
```

19. Sales team

Write a class called `SalesTeam` that represents a team of salespeople. Each salesperson on the team is represented by the `SalesPerson` class as in the previous question. Each team has a name, and the constructor needs only to accept the name of the team. Use and `ArrayList` to store the team members. Provide a method called `addSalesPerson` that accepts a `SalesPerson` object. Provide a method called `weeklyReport` that prints the name and the total sale amount of each team member and the total amount for the entire team. Write a driver class that fully exercise the `SalesTeam` class.

```
import java.util.*;

public class foobar {
    public static void main(String[] args) {
        SalesTeam team = new SalesTeam("team");

        SalesPerson sp = new SalesPerson("someName", "9999999999", "someDistrict");
        sp.setDailyAmount(0, 1000);
        sp.setDailyAmount(1, 2000);
        sp.setDailyAmount(2, 3000);
        sp.setDailyAmount(3, 4000);
        sp.setDailyAmount(4, 5000);

        assert sp.getDailyAmount(2) == 3000;
        assert sp.getDailyAmount(4) == 5000;

        assert sp.total() == 15000;
        assert (int)sp.avg() == (int)(15000/7);

        team.addSalesPerson(sp);
        team.weeklyReport();
        System.out.println("All tests passed");
    }
}

class SalesTeam {
    ArrayList<SalesPerson> members;
    String name;

    public SalesTeam(String name) {
        this.name = name;
        members = new ArrayList<>();
    }

    public void addSalesPerson(SalesPerson p) { members.add(p); }

    public void weeklyReport() {
        System.out.println("Sales team --> " + name);
        int sales = 0;
        for (SalesPerson per : members)
            sales += per.total();

        System.out.println("Total sales --> " + sales);

        for (SalesPerson per : members)
            System.out.println(per.toString());
```

```
    }
}

class SalesPerson {
    String name;
    String phoneNum;
    String district;
    int[] sales;

    public SalesPerson(String name, String phoneNum, String district) {
        this.name = name;
        this.phoneNum = phoneNum;
        this.district = district;
        sales = new int[] {0, 0, 0, 0, 0, 0, 0};
    }

    public int getDailyAmount(int day) { return sales[day]; }

    public void setDailyAmount(int day, int amt) { this.sales[day] = amt; }

    public int total() {
        int sum = 0;
        for (int val : sales)
            sum += val;
        return sum;
    }

    public double avg() { return total() / 7.0; }

    @Override
    public String toString() {
        return "\nName --> " + name + "\nDistrict --> " + district +
        "\nPhone --> " + phoneNum + "\nTotal sales --> " + total();
    }
}
```

```
[getpsyched@Manjaro Assignment 2]$ java -ea "19. Sales Team.java"
Sales team --> team
Total sales --> 15000

Name --> someName
District --> someDistrict
Phone --> 9999999999
Total sales --> 15000
All tests passed
[getpsyched@Manjaro Assignment 2]$
```