

CSCI 145 PA __8__ Submission

Due Date: __Apr 24, 2023__ Late (date and time): _____

Name(s): _____ Ivan Leung _____ & _____

Exercise 1 -- need to submit source code and I/O

-- check if completely done __x__ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

```
package pa8;
```

```
/* Java Class: CSCI 145  
Modified by: Ivan Leung  
Class: Mon/Wed  
Date: Apr 17 2023  
Description:
```

I certify that the code below is my own work.

Exception(s): N/A

```
*/
```

```
//*****  
//Sales.java  
//  
//Reads in and stores sales for each of 5 salespeople. Displays  
//sales entered by salesperson id and total sales for all  
//salespeople.  
//  
//*****
```

```
import java.util.Scanner;  
import java.text.DecimalFormat;
```

```
public class Sales {
```

```

    public static void main(String[] args) {
        final int SALESPEOPLE = 5;
//        int[] sales = new int[SALESPEOPLE];
        int[] sales;
        int sum, min, max, salesTarget,
totalSalesTargetPeople, totalSalesPeople;
        double mean;
        Scanner scan = new Scanner(System.in);
        DecimalFormat decimal = new DecimalFormat("#.00");

        System.out.print("Enter total number of salesperson:
");
        totalSalesPeople = scan.nextInt();

        sales = new int[totalSalesPeople];

        System.out.println();
        for (int i = 0; i < sales.length; i++) {
            System.out.print("Enter sales for salesperson " +
(i + 1) + ": ");
            sales[i] = scan.nextInt();
        }
        System.out.println("\nSalesperson Sales");
        System.out.println(" ----- ");

        sum = 0;
        min = sales[0];
        max = sales[0];

        for (int i = 0; i < sales.length; i++) {
            System.out.println(" " + (i + 1) + " " +
sales[i]);
            sum += sales[i];
            if (sales[i] > max)
                max = sales[i];
            else if (sales[i] < min)
                min = sales[i];
        }
        mean = (double) sum / SALESPEOPLE;

        System.out.println("\nTotal sales: " + sum);
        System.out.println("Average sales: " +
decimal.format(mean));
        System.out.println("Maximum sales: " + max);
        System.out.println("Minimum sales: " + min);
    }

```

```

        System.out.print("\nEnter sales target: ");
        salesTarget = scan.nextInt();
        System.out.println();

        scan.close();

        totalSalesTargetPeople = 0;

        System.out.println("\nSalesperson who hit the sales
target");
        System.out.println(" -----
---- ");
        for (int i = 0; i < sales.length; ++i) {
            if (sales[i] >= salesTarget) {
                System.out.println("Salesperon " + (i + 1)
+ " " + sales[i]);
                ++totalSalesTargetPeople;
            }
        }
        System.out.println("Total salespeople hit the sales
tartget: " + totalSalesTargetPeople);
    }
}

```

Input/output below:

Enter total number of salesperson: 6

Enter sales for salesperson 1: 3829

Enter sales for salesperson 2: 8265

Enter sales for salesperson 3: 7385

Enter sales for salesperson 4: 2347

Enter sales for salesperson 5: 6589

Enter sales for salesperson 6: 4892

Salesperson Sales

1 3829

2 8265

3 7385

4 2347

5 6589

6 4892

Total sales: 33307
Average sales: 6661.40
Maximum sales: 8265
Minimum sales: 2347

Enter sales target: 4000

Salesperson who hit the sales target

Salesperson 2 8265
Salesperson 3 7385
Salesperson 5 6589
Salesperson 6 4892
Total salespeople hit the sales target: 4

Exercise 2 -- need to submit source code and I/O
-- check if completely done ___x___ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

```
package pa8;  
  
import java.text.NumberFormat;
```

```
/* Java Class: CSCI 145  
Author: Ivan Leung  
Class: Mon/Wed  
Date: Apr 17 2023  
Description:
```

```
I certify that the code below is my own work.
```

```
Exception(s): N/A
```

```
*/
```

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

```

public class Shopping {

    public static void main(String[] args) {
        ShoppingCart shoppingCart = new ShoppingCart();
        String itemName;
        double itemPrice;
        int itemQty;
        double totalPrice = 0;
        String continueShopping = "y";
        Scanner scan = new Scanner(System.in);
        NumberFormat currency =
NumberFormat.getCurrencyInstance();

        while(continueShopping.trim().equalsIgnoreCase("y")) {
            System.out.println("\nAdding item to the
cart...\n");

            System.out.print("What is name of the item: ");
            itemName = scan.next();
            System.out.print("How much the " + itemName + "
is? ");

            itemPrice = scan.nextDouble();
            System.out.print("How many " + itemName + " do
you want? ");

            itemQty = scan.nextInt();
            shoppingCart.addToCart(itemName, itemPrice,
itemQty);

            totalPrice += (itemPrice * itemQty);

            System.out.println(shoppingCart);
            System.out.print("Do you want to add more items
(y or n)? ");

            continueShopping = scan.next();
        }
        scan.close();
        System.out.println("\nPlease pay...");
        System.out.println("Total: " +
currency.format(totalPrice));
    }

}

```

Input/output below:

Adding item to the cart...

What is name of the item: T-shirt

How much the T-shirt is? 8.99

How many T-shirt do you want? 3

Shopping Cart

Item	Unit Price	Quantity	Total
T-shirt	\$8.99	3	\$26.97

Total Price: \$26.97

Do you want to add more items (y or n)? y

Adding item to the cart...

What is name of the item: Pants

How much the Pants is? 14.59

How many Pants do you want? 2

Shopping Cart

Item	Unit Price	Quantity	Total
T-shirt	\$8.99	3	\$26.97
Pants	\$14.59	2	\$29.18

Total Price: \$56.15

Do you want to add more items (y or n)? y

Adding item to the cart...

What is name of the item: Socks

How much the Socks is? 2.99

How many Socks do you want? 10

Shopping Cart

Item	Unit Price	Quantity	Total
T-shirt	\$8.99	3	\$26.97
Pants	\$14.59	2	\$29.18
Socks	\$2.99	10	\$29.90

Total Price: \$86.05

Do you want to add more items (y or n)? y

Adding item to the cart...

What is name of the item: PS5

How much the PS5 is? 699.99

How many PS5 do you want? 1

Shopping Cart

Item	Unit Price	Quantity	Total
T-shirt	\$8.99	3	\$26.97
Pants	\$14.59	2	\$29.18
Socks	\$2.99	10	\$29.90
PS5	\$699.99	1	\$699.99

Total Price: \$786.04

Do you want to add more items (y or n)? n

Please pay...

Total: \$786.04

Exercise 3 -- need to submit source code and I/O

-- check if completely done __x__ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

```
package pa8;
```

```
/* Java Class: CSCI 145
```

```
Author: Ivan Leung
```

```
Class: Mon/Wed
```

```
Date: Apr 17 2023
```

```
Description:
```

I certify that the code below is my own work.

Exception(s): N/A

*/

```
import java.text.NumberFormat;
```

```
public class RollingDice {
```

```
    public static void main(String[] args) {
        Die die = new Die();
        int maxSum = 12;
        int maxRoll = 1000;
        int[] results = new int[maxSum];
        NumberFormat percent =
NumberFormat.getPercentInstance();
        percent.setMinimumFractionDigits(1);

        for (int i = 0; i < maxRoll; ++i) {
            ++results[die.roll() + die.roll() - 1];
        }

        System.out.println("Value\tCount\tPercentages");
        for (int i = 0; i < 12; ++i) {
            System.out.println(i + 1 + "\t" + results[i] +
"\t" + percent.format((double) results[i] / maxRoll));

        }

    }

}
```

Input/output below:

Value Count Percentages

1	0	0.0%
2	27	2.7%
3	49	4.9%
4	84	8.4%
5	117	11.7%
6	154	15.4%
7	153	15.3%
8	134	13.4%
9	118	11.8%

10	88	8.8%
11	48	4.8%
12	28	2.8%

Add more exercises as needed

Exercise 4 -- need to submit source code and I/O

-- check if completely done ___x___ ; otherwise, discuss issues below

Pseudocode below if applicable:

Source code below:

```
package pa8;
```

```
/* Java Class: CSCI 145
```

```
Author: Ivan Leung
```

```
Class: Mon/Wed
```

```
Date: Apr 17 2023
```

```
Description:
```

```
I certify that the code below is my own work.
```

```
Exception(s): N/A
```

```
*/
```

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

```
import java.text.DecimalFormat;
```

```
public class ProcessValues {
```

```
    public static void main(String[] args) {
```

```
        int totalVal;
```

```
        int inRangeQty;
```

```
        int[] list;
```

```
        int inRangeSum;
```

```
        int min;
```

```
        int max;
```

```
        Scanner scan = new Scanner(System.in);
```

```
        DecimalFormat decimal = new DecimalFormat("#.0");
```

```
        System.out.print("How many values? ");
```

```

totalVal = scan.nextInt();
list = new int[totalVal];

System.out.print("Input " + totalVal + " values: ");
for (int i = 0; i < totalVal; ++i) {
    list[i] = scan.nextInt();
}

System.out.print("Input minimum and maximum: ");
min = scan.nextInt();
max = scan.nextInt();
scan.close();

inRangeSum = 0;
inRangeQty = 0;
System.out.print("Values between " + min + " and " +
max + ":" );
for (int i = 0; i < totalVal; ++i) {
    if (list[i] >= min && list[i] <= max) {
        System.out.print(" " + list[i]);
        inRangeSum += list[i];
        ++inRangeQty;
    }
}
System.out.println("\nTheir average: " +
decimal.format((double) inRangeSum / inRangeQty));
}
}

```

Input/output below:

```

How many values? 8
Input 8 values: 89 32 90 54 101 31 77 10
Input minimum and maximum: 32 89
Values between 32 and 89: 89 32 54 77
Their average: 63.0

```

Answer for Question 1

Yes, it is reasonable. According to the probability of rolling two dice, the middle number, 7, has the highest probability of 16.67%. Starting from the number 2 and 12, the

probability increases as the number goes toward the middle number, 7. Comparing to the results of rolling two dice from exercise 3, they are very close to each other. If we roll the two dice enough time, the results will get even closer to the probability of rolling two dice.

Answer for Question 2

One important reason for choosing an array over an ArrayList is that ArrayList cannot hold primitive type while array can hold both primitive type and reference type. ArrayList also has dynamic size which we do not need for our purposes, so it is better to use array performance wise.

Extra Credit – provide if applicable

Pseudocode below if applicable:

Source code below:

```
package pa8;
```

```
/* Java Class: CSCI 145  
Author: Ivan Leung  
Class: Mon/Wed  
Date: Apr 17 2023  
Description:
```

```
I certify that the code below is my own work.
```

```
Exception(s): N/A
```

```
*/
```

```
public class Dice {  
    private Die[] die;  
    private int numberOfDie;  
    private int faceValue;  
  
    public Dice(int numberOfDie) {  
        this.numberOfDie = numberOfDie;  
        die = new Die[this.numberOfDie];  
        for (int i = 0; i < this.numberOfDie; ++i) {  
            die[i] = new Die();  
        }  
    }  
}
```

```

    }

    public int roll() {
        faceValue = 0;
        for (int i = 0; i < numberOfDie; ++i) {
            faceValue += die[i].roll();
        }
        return faceValue;
    }

    public void setFaceValue(int value) {
        faceValue = value;
    }

    public int getFaceValue() {
        return faceValue;
    }

    public String toString() {
        return Integer.toString(faceValue);
    }
}

```

package pa8;

/* Java Class: CSCI 145
 Author: Ivan Leung
 Class: Mon/Wed
 Date: Apr 17 2023
 Description:

I certify that the code below is my own work.

Exception(s): N/A

*/

import java.text.NumberFormat;

```

public class RollingDice {

    public static void main(String[] args) {
        Dice twoDice = new Dice(2);
        Dice fiveDice = new Dice(5);
    }
}

```

```

        int maxSum = 12;
        int maxRoll = 1000;
        int[] results = new int[maxSum];
        NumberFormat percent =
NumberFormat.getPercentInstance();
        percent.setMinimumFractionDigits(1);

        for (int i = 0; i < maxRoll; ++i) {
            ++results[twoDice.roll() - 1];
        }

        System.out.println("Value\tCount\tPercentages");
        for (int i = 0; i < 12; ++i) {
            System.out.println(i + 1 + "\t" + results[i] +
"\t" + percent.format((double) results[i] / maxRoll));
        }

        System.out.println("\nRolling five dice a few
times...");
        for (int i = 0; i < 5; ++i) {
            System.out.println("Rolled " + fiveDice.roll());
        }
    }
}

```

Input/output below:

Value Count Percentages

1	0	0.0%
2	21	2.1%
3	50	5.0%
4	82	8.2%
5	112	11.2%
6	153	15.3%
7	148	14.8%
8	139	13.9%
9	115	11.5%
10	74	7.4%
11	63	6.3%
12	43	4.3%

Rolling five dice a few times...

Rolled 15

Rolled 20
Rolled 16
Rolled 16
Rolled 15