

CS 1073

FR04B

Assignment 7

Ebrahim Arefi

3621326

1) the source code for Question I:

```
/**
 * A class that calculate payments of different items.
 * @author Ebrahim Arefi, 3621326
 */

import java.util.Scanner;

public class As7Q1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the cost of the item:");
        int cost = sc.nextInt();

        while (cost <= 0) {
            System.out.println("Invalid cost. Enter the cost of the
item:");
            cost = sc.nextInt();
        }

        System.out.println("Enter the total value of the payment:");
        int payment = sc.nextInt();

        while (payment < cost) {
            System.out.println("Invalid payment. Enter the total value
of the payment:");
            payment = sc.nextInt();
        }

        if (payment == cost) {
            System.out.println("Exact payment – no change is due.");
        } else {
            int change = payment - cost;
            String Symbols = "";

            while (change >= 1000) {
                Symbols += "G";
                change -= 1000;
            }

            while (change >= 500) {
```

```

        Symbols += "S";
        change -= 500;
    }

    while (change >= 100) {
        Symbols += "B";
        change -= 100;
    }

    while (change >= 50) {
        Symbols += "C";
        change -= 50;
    }

    while (change >= 10) {
        Symbols += "W";
        change -= 10;
    }

    while (change >= 1) {
        Symbols += "F";
        change -= 1;
    }

    System.out.println(Symbols);
}
}
}

```

2) the sample output for Q1:

```
ebi@iPhone As7 % javac As7Q1.java
ebi@Ebis-MBP as7 % java As7Q1
```

```
-----
Enter the cost of the item:
222
Enter the total value of the payment:
12
Invalid payment. Enter the total value of the payment:
4567
GGGGBBBWWWWFFFFF
ebi@Ebis-MBP as7 % java As7Q1
```

```
-----
Enter the cost of the item:
-10
Invalid cost. Enter the cost of the item:
655
Enter the total value of the payment:
600
Invalid payment. Enter the total value of the payment:
700
WWWWFFFFF
ebi@Ebis-MBP as7 % java As7Q1
```

```
-----
Enter the cost of the item:
12
Enter the total value of the payment:
567
SCFFFFF
ebi@Ebis-MBP as7 %
```

3) Source code for Question II (parts a& b)

class As7Q2:

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

/**
 * Driver class for the Trick or Treat Bag App.
 * Allows users to enter the dimensions for Witch Hat and Frankenstein bags
 * and calculates their surface area and volume.
 *
 * @author Ebrahim Arefi, 3621326
 */
public class As7Q2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DecimalFormat df = new DecimalFormat("#,###.000");

        double smallestSurface = 0;
        double largestVolume = 0;
        String smallestType = "";
        String largestType = "";

        boolean running = true;
        while (running) {
            System.out.println("");
            System.out.println("-----");
            System.out.println("Select one of the following options:");
            System.out.println("1 - Enter the information for a Witch Hat bag");
            System.out.println("2 - Enter the information for a Frankenstein bag");
            System.out.println("3 - Quit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            if (choice == 3) {
                running = false;
                if (largestVolume > 0) {
                    System.out.println(
                        "The trick or treat bag that can hold the most candy is a "
                        + largestType + " bag.");
                    System.out.println("Highest volume: " + df.format(largestVolume) +
                        " cm^3");
                }
                if (smallestSurface > 0) {
                    System.out.println(
                        "The trick or treat bag with the smallest surface area is a "
                        + smallestType + " bag.");
                    System.out.println("Smallest surface area: " +
                        df.format(smallestSurface) + " cm^2");
                }
            }
        }
    }
}
```

```

    }
} else if (choice == 1) {
    System.out.print("Input radius of opening (in cm): ");
    double openingRadius = sc.nextDouble();
    while (openingRadius <= 0) {
        System.out.print("Invalid Input. Input radius of opening (in cm):
");
        openingRadius = sc.nextDouble();
    }

    System.out.print("Input radius to outside of brim (in cm): ");
    double outsideRadius = sc.nextDouble();
    while (outsideRadius <= 0 || outsideRadius < openingRadius) {
        System.out.print("Invalid Input. Input radius to outside of brim
(in cm): ");
        outsideRadius = sc.nextDouble();
    }

    System.out.print("Input length of the bag (in cm): ");
    double length = sc.nextDouble();
    while (length <= 0) {
        System.out.print("Invalid Input. Input length of the bag (in cm):
");
        length = sc.nextDouble();
    }

    WitchHatBag bag = new WitchHatBag(openingRadius, outsideRadius,
length);

    double surface = bag.calculateOutside();
    double volume = bag.calculateVolume();

    System.out.println("Surface area: " + df.format(surface) + " cm^2");
    System.out.println("Volume: " + df.format(volume) + " cm^3");

    if (smallestSurface == 0 || surface < smallestSurface) {
        smallestSurface = surface;
        smallestType = "Witch Hat";
    }
    if (volume > largestVolume) {
        largestVolume = volume;
        largestType = "Witch Hat";
    }
} else if (choice == 2) {
    System.out.print("Input width of opening (in cm): ");
    double width = sc.nextDouble();
    while (width <= 0) {
        System.out.print("Invalid Input. Input width of opening (in cm):
");
        width = sc.nextDouble();
    }

    System.out.print("Input depth of opening (in cm): ");
    double depth = sc.nextDouble();

```

```

        while (depth <= 0) {
            System.out.print("Invalid Input. Input depth of opening (in cm):
");
            depth = sc.nextDouble();
        }

        System.out.print("Input length of the bag (in cm): ");
        double length = sc.nextDouble();
        while (length <= 0) {
            System.out.print("Invalid Input. Input length of the bag (in cm):
");
            length = sc.nextDouble();
        }

        FrankensteinBag bag = new FrankensteinBag(width, depth, length);
        double surface = bag.calculateOutside();
        double volume = bag.calculateVolume();

        System.out.println("Surface area: " + df.format(surface) + " cm^2");
        System.out.println("Volume: " + df.format(volume) + " cm^3");

        if (smallestSurface == 0 || surface < smallestSurface) {
            smallestSurface = surface;
            smallestType = "Frankenstein";
        }
        if (volume > largestVolume) {
            largestVolume = volume;
            largestType = "Frankenstein";
        }
    } else {
        System.out.println("Invalid choice - try again!");
    }
}
}
}

```

class WitchHatBag:

```
/**
 * This class represents a Witch Hat trick-or-treat bag shaped like a cone
with
 * a circular brim.
 * It can calculate the outside material required (surface area) and the
volume
 * of candy the bag can hold.
 *
 * @author Ebrahim Arefi, 3621326
 */
public class WitchHatBag {

    /**
     * The radius of the circular opening (in cm).
     */
    private double openingRadius;

    /**
     * The radius to the outer edge of the brim (in cm).
     */
    private double outsideRadius;

    /**
     * The length of the bag (in cm).
     */
    private double length;

    /**
     * Constructs a WitchHatBag with a given opening radius, outside radius,
and
     * length.
     *
     * @param openingRadiusIn the radius of the circular opening (in cm).
     * @param outsideRadiusIn the radius to the outer brim edge (in cm).
     * @param lengthIn       the length of the bag (in cm).
     */
    public WitchHatBag(double openingRadiusIn, double outsideRadiusIn, double
lengthIn) {
        openingRadius = openingRadiusIn;
        outsideRadius = outsideRadiusIn;
        length = lengthIn;
    }

    /**
     * Retrieves the radius of the circular opening.
```



```

    *
    * @return the radius of the circular opening (in cm).
    */
    public double getOpeningRadius() {
        return openingRadius;
    }

    /**
     * Retrieves the radius to the outer edge of the brim.
     *
     * @return the radius to the outer brim (in cm).
     */
    public double getOutsideRadius() {
        return outsideRadius;
    }

    /**
     * Retrieves the length of the bag.
     *
     * @return the length (in cm).
     */
    public double getLength() {
        return length;
    }

    /**
     * Calculates the total outside surface area of the Witch Hat bag.
     *
     * @return the surface area in cm2.
     */
    public double calculateOutside() {
        double slantHeight = Math.sqrt(Math.pow(length, 2) +
Math.pow(openingRadius, 2));
        double lateralArea = Math.PI * openingRadius * slantHeight;
        double brimArea = Math.PI * (Math.pow(outsideRadius, 2) -
Math.pow(openingRadius, 2));
        return lateralArea + brimArea;
    }

    /**
     * Calculates the volume of the Witch Hat bag.
     *
     * @return the volume in cm3.
     */
    public double calculateVolume() {
        return (Math.PI * Math.pow(openingRadius, 2) * length) / 3.0;
    }
}

```

class FrankensteinBag:

```
/**
 * This class represents Frankenstein bag.
 * Calculate the surface area and volume of the bag.
 *
 * @author Ebrahim Arefi, 3621326
 */
public class FrankensteinBag {

    /**
     * The width of the rectangular opening (in cm).
     */
    private double width;

    /**
     * The depth of the rectangular opening (in cm).
     */
    private double depth;

    /**
     * The length of the bag (in cm).
     */
    private double length;

    /**
     * Constructs a FrankensteinBag with a given width, depth, and length.
     *
     * @param widthIn the width of the bag (in cm).
     * @param depthIn the depth of the bag (in cm).
     * @param lengthIn the length (in cm).
     */
    public FrankensteinBag(double widthIn, double depthIn, double lengthIn) {
        width = widthIn;
        depth = depthIn;
        length = lengthIn;
    }

    /**
     * Retrieves the width of the bag.
     *
     * @return the width of the bag (in cm).
     */
    public double getWidth() {
        return width;
    }
}
```

```

/**
 * Retrieves the depth of the bag.
 *
 * @return the depth of the bag (in cm).
 */
public double getDepth() {
    return depth;
}

/**
 * Retrieves the length of the bag.
 *
 * @return the length of the bag (in cm).
 */
public double getLength() {
    return length;
}

/**
 * Calculates the surface area of the bag.
 * The top of the bag is open.
 *
 * @return the surface area in cm2.
 */
public double calculateOutside() {
    double bottom = width * depth;
    double sides = 2 * (width * length + depth * length);
    return bottom + sides;
}

/**
 * Calculates the volume of the bag.
 *
 * @return the volume in cm3.
 */
public double calculateVolume() {
    return width * depth * length;
}
}

```

4) Sample output: (Question II)

```
ebi@Ebis-MBP as7 % javac As7Q2.java
ebi@Ebis-MBP as7 % java As7Q2
```

```
-----
Select one of the following options:
```

- 1 - Enter the information for a Witch Hat bag
- 2 - Enter the information for a Frankenstein bag
- 3 - Quit

```
Enter your choice: 4
```

```
Invalid choice - try again!
```

```
-----
Select one of the following options:
```

- 1 - Enter the information for a Witch Hat bag
- 2 - Enter the information for a Frankenstein bag
- 3 - Quit

```
Enter your choice: 1
```

```
Input radius of opening (in cm): -6.7
```

```
Invalid Input. Input radius of opening (in cm): 3.4
```

```
Input radius to outside of brim (in cm): 9.95
```

```
Input length of the bag (in cm): 14.53
```

```
Surface area: 434.102 cm^2
```

```
Volume: 175.894 cm^3
```

```
-----
Select one of the following options:
```

- 1 - Enter the information for a Witch Hat bag
- 2 - Enter the information for a Frankenstein bag
- 3 - Quit

```
Enter your choice: 2
```

```
Input width of opening (in cm): 6.5
```

```
Input depth of opening (in cm): -5.4
```

```
Invalid Input. Input depth of opening (in cm): 5.4
```

```
Input length of the bag (in cm): 16.8
```

```
Surface area: 434.940 cm^2
```

```
Volume: 589.680 cm^3
```

```
-----
Select one of the following options:
```

- 1 - Enter the information for a Witch Hat bag
- 2 - Enter the information for a Frankenstein bag
- 3 - Quit

```
Enter your choice: 1
```

```
Input radius of opening (in cm): 8.5
```

```
Input radius to outside of brim (in cm): 14.3
```

```
Input length of the bag (in cm): 18.9
```

```
Surface area: 968.833 cm^2
```

Volume: 1,429.974 cm³

Select one of the following options:

- 1 - Enter the information for a Witch Hat bag
- 2 - Enter the information for a Frankenstein bag
- 3 - Quit

Enter your choice: 3

The trick or treat bag that can hold the most candy is a Witch Hat bag.

Highest volume: 1,429.974 cm³

The trick or treat bag with the smallest surface area is a Witch Hat bag.

Smallest surface area: 434.102 cm²