

/\* Saket Bakshi 2/3/19. Period 6

This program, for #3 of Ch 8, completes a cash register class.

\*/

public class PracticeExercisesCh8E3

{

private double purchase;

private double payment;

/\*\* Constructs a cash register with no money.

\*/

public PracticeExercisesCh8E3()

{

    this.purchase = 0;

    this.payment = 0;

}

/\*\* Records an item purchase.

@param amount the price of the item

\*/

public void recordPurchase(double amount)

{

    this.purchase = this.purchase + amount;

}

/\*\* Enters the payment received from the customer; should be called once  
for each monetary unit type

@param count the number of monetary units

@param coinType the type of the monetary units in the payment

\*/

public void enterPayment(int count, PracticeExercisesCh8E3Coin coinType)

{

    this.payment = this.payment + count \* coinType.getValue();

}

/\*\* Computes the change due and resets the machine for the next customer.

@return the change due to the customer

\*/

public double giveChange()

{

    double change = payment - purchase;

    purchase = 0;

    payment = 0;

    change = Math.round(change \* 100);

```

        change = change / 100;
        return change;
    }
}

/* Saket Bakshi 2/3/19. Period 6
This program, for #3 of Ch 8, completes a coin value class.
*/
public class PracticeExercisesCh8E3Coin
{
    private double value;
    private String name;

    /**    Constructs a coin.
    @param value the monetary value of the coin.
    @param name the name of the coin
    */
    public PracticeExercisesCh8E3Coin(double value, String name)
    {
        this.value = value;
        this.name = name;
    }

    /**    Gets the coin value.
    @return the value of the coin
    */
    public double getValue()
    {
        return value;
    }

    /**    Gets the coin name.
    @return the name of the coin
    */
    public String getName()
    {
        return name;
    }
}

```

```

/* Saket Bakshi 2/3/19. Period 6
This program, for #3 of Ch 8, tests a cash register class.
*/

```

```

public class PracticeExercisesCh8E3Tester
{
    public static void main(String[] args)
    {
        final double DOLLAR_VALUE = 1.0;
        final double QUARTER_VALUE = 0.25;
        final double DIME_VALUE = 0.1;
        final double NICKEL_VALUE = 0.05;

        PracticeExercisesCh8E3 test = new PracticeExercisesCh8E3();

        test.recordPurchase(5.03);

        test.enterPayment(4, new
PracticeExercisesCh8E3Coin(DOLLAR_VALUE, "dollar bill"));
        test.enterPayment(5, new
PracticeExercisesCh8E3Coin(QUARTER_VALUE, "quarter"));

        double myChange = test.giveChange();
        System.out.println("Change: " + myChange);
        System.out.println("Expected: 0.22");
    }
}

```

```

PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket> java PracticeExercisesCh8E3Tester
Change: 0.22
Expected: 0.22
PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket>

```

/\* Saket Bakshi 1/30/19. Period 6

This program, for #5 of Ch 8, calculates surface areas and volumes for certain 3D objects.

\*/

public class PracticeExercisesCh8E5

{

/\*\* Calculates cube volume

@param h the height of the cube

@return the volume of the cube

\*/

public static double cubeVolume(double h)

{

return h\*h\*h;

}

/\*\* Calculates cube surface area

@param h the height of the cube

@return the surface area of the cube

\*/

public static double cubeSurface(double h)

{

return 6\*h\*h;

}

/\*\* Calculates sphere volume

@param r the radius of the sphere

@return the volume of the sphere

\*/

public static double sphereVolume(double r)

{

return 4/3\*Math.PI\*r\*r\*r;

}

/\*\* Calculates sphere surface area

@param r the radius of the sphere

@return the surface area of the sphere

\*/

public static double sphereSurface(double r)

{

return 4\*Math.PI\*r\*r;

}

/\*\* Calculates cylinder volume

@param h the height of the cylinder

```

    @param r the radius of the cylinder
    @return the volume of the cylinder
    */
    public static double cylinderVolume(double r, double h)
    {
        return Math.PI*r*r*h;
    }

    /** Calculates cylinder surface area
    @param h the height of the cylinder
    @param r the radius of the cylinder
    @return the surface area of the cylinder
    */
    public static double cylinderSurface(double r, double h)
    {
        return (2*Math.PI*r*r) + (2*Math.PI*r*h);
    }

    /** Calculates cone volume
    @param h the height of the cone
    @param r the radius of the cone
    @return the volume of the cone
    */
    public static double coneVolume(double r, double h)
    {
        return Math.PI*r*r*h/3;
    }

    /** Calculates cone surface area
    @param h the height of the cone
    @param r the radius of the cone
    @return the surface area of the cone
    */
    public static double coneSurface(double r, double h)
    {
        return Math.PI*r*(r+Math.sqrt(h*h + r*r));
    }
}

/* Saket Bakshi 12/10/18. Period 6
This program, for #5 of Ch 8, tests the #5 class.
*/
import java.util.Scanner;

```

```

public class PracticeExercisesCh8E5Tester
{
    public static void main(String[] args) {
        Scanner key = new Scanner(System.in);

        PracticeExercisesCh8E5 test = new PracticeExercisesCh8E5(); //introduces
object
        System.out.println("I can calculate volumes and surface areas for certain 3D
objects after you give me a certain radius and height.");
        System.out.print("Give me a radius: "); //prompts for dimensions
        double r = key.nextDouble();
        System.out.print("Give me a height: ");
        double h = key.nextDouble();

        System.out.printf("A cube's volume is: %25s", test.cubeVolume(h) + "\n");
//returns and prints all calculations
        System.out.printf("A cube's surface area is: %19s", test.cubeSurface(h) + "\n");
        System.out.printf("A sphere's volume is: %23s", test.sphereVolume(r) + "\n");
        System.out.printf("A sphere's surface area is: %10s", test.sphereSurface(r) +
"\n");
        System.out.printf("A cylinder's volume is: %21s", test.cylinderVolume(r, h) + "\n");
        System.out.printf("A cylinder's surface area is: %9s", test.cylinderSurface(r, h) +
"\n");
        System.out.printf("A cone's volume is: %25s", test.coneVolume(r, h) + "\n");
        System.out.printf("A cone's surface area is: %19s", test.coneSurface(r, h) + "\n");
    }
}

```

```

PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket> java PracticeExercisesCh8E5Tester
I can calculate volumes and surface areas for certain 3D objects after you give me a certain radius and height.
Give me a radius: 1
Give me a height: 2
A cube's volume is: 8.0
A cube's surface area is: 24.0
A sphere's volume is: 3.141592653589793
A sphere's surface area is: 12.566370614359172
A cylinder's volume is: 6.283185307179586
A cylinder's surface area is: 18.84955592153876
A cone's volume is: 2.0943951023931953
A cone's surface area is: 10.166407384630519
PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket>

```

/\* Saket Bakshi 1/30/19. Period 6

This program, for #12 of Ch 8, tests whether an integer is between a minimum or maximum.

\*/

import java.util.Scanner;

public class PracticeExercisesCh8E12

{

/\*\* Prompts with a given prompt for an integer. Reads the integer and sees if it's within a given minimum or maximum. Subsequently prints out given error messages if needed.

\*/

public static void readInt(Scanner in, String prompt, String error, int min, int max)

{

System.out.println("The prompt is: \"\" + prompt + "\"");

int tested = in.nextInt();

if(tested >= min && tested <= max)

{

System.out.println("The integer is between your given maximum and minimum.");

}

else

{

System.out.println(error);

}

}

}

/\* Saket Bakshi 1/30/19. Period 6

This program, for #12 of Ch 8, tests whether an integer is between a minimum or maximum.

\*/

import java.util.Scanner;

public class PracticeExercisesCh8E12Tester

{

public static void main(String[] args) {

Scanner key = new Scanner(System.in);

String prompt, error;

int min, max;

System.out.print("What do you want as your prompt for asking for an integer? ");

prompt = key.nextLine();

System.out.println();

```

        System.out.print("Give me an error message, a minimum integer, and a
maximum integer. ");
        error = key.nextLine();

        System.out.println();

        System.out.print("Give me a minimum: ");
        min = key.nextInt();

        System.out.print("Give me a maximum: ");
        max = key.nextInt();

        PracticeExercisesCh8E12 test = new PracticeExercisesCh8E12();
        test.readInt(key, prompt, error, min, max);
    }
}

```

```

PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket> java PracticeExercisesCh8E12Tester
What do you want as your prompt for asking for an integer? please give an integer

Give me an error message, a minimum integer, and a maximum integer. error!!

Give me a minimum: 2
Give me a maximum: 5
The prompt is: "please give an integer"
4
The integer is between your given maximum and minimum.
PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket> java PracticeExercisesCh8E12Tester
What do you want as your prompt for asking for an integer? please give an integer

Give me an error message, a minimum integer, and a maximum integer. error!!

Give me a minimum: 2
Give me a maximum: 5
The prompt is: "please give an integer"
7
error!!
PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket>

```



/\* Saket Bakshi 2/3/19. Period 6

This program, for Project 1 of Ch 8, creates a class to open a three digit lock.

\*/

public class PracticeExercisesCh8P1

{

private boolean open;

private int number1, number2, number3;

private int currentPlace;

private int[] attempt;

private int order;

/\*\* Creates a lock object

@param secret1 the first lock digit

@param secret2 the second lock digit

@param secret3 the third lock digit

\*/

public PracticeExercisesCh8P1(int secret1, int secret2, int secret3)

{

    this.open = false;

    this.number1 = secret1;

    this.number2 = secret2;

    this.number3 = secret3;

    this.currentPlace = 0;

    this.attempt = new int[3];

    this.attempt[0] = 0;

    this.attempt[1] = 0;

    this.attempt[2] = 0;

    this.order = 0;

}

/\*\* Resets the lock.

\*/

public void reset()

{

    this.attempt[0] = 0;

    this.attempt[1] = 0;

    this.attempt[2] = 0;

    this.currentPlace = 0;

```

        this.order = 0;
    }

    /** Turns the lock to the right.
    */
    public void turnRight(int ticks)
    {
        this.currentPlace = this.currentPlace - ticks;
        while(this.currentPlace < 0)
        {
            this.currentPlace = this.currentPlace + 40;
        }
        this.attempt[this.order] = this.currentPlace;
        this.order++;
    }

    /** Turns the lock to the right.
    */
    public void turnLeft(int ticks)
    {
        this.currentPlace = this.currentPlace + ticks;
        while(this.currentPlace > 39)
        {
            this.currentPlace = this.currentPlace - 40;
        }
        this.attempt[this.order] = this.currentPlace;
        this.order++;
    }

    /** Tries to open the lock.
    */
    public void open()
    {
        if(this.attempt[0] == this.number1 && this.attempt[1] == this.number2 &&
this.attempt[2] == this.number3)
        {
            this.open = true;
            System.out.println("The lock is open!");
        }
        else
        {
            this.open = false;

```

```

        System.out.println("The lock was not opened. You inputted \" +
this.attempt[0] + ", " + this.attempt[1] + ", " + this.attempt[2] + "\" \nTry again.\n");
    }
}
}
/* Saket Bakshi 2/3/19. Period 6
This program, for Project 1 of Ch 8, tests a class that opens a three digit lock.
*/
public class PracticeExercisesCh8P1Tester
{
    public static void main(String[] args)
    {
        PracticeExercisesCh8P1 test = new PracticeExercisesCh8P1(10, 20, 5);
        System.out.println("Code is set to \"10, 20, 5\"");

        test.turnLeft(23);
        test.turnRight(30);
        test.turnLeft(5);
        System.out.println("\tExpected: Lock should not open. Code inputted was 23, 33,
38.");

        test.open();
        test.reset();

        test.turnLeft(10);
        test.turnRight(30);
        test.turnLeft(25);
        System.out.println("\tExpected: Lock should open.");
        test.open();
    }
}

```

```

PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket> java PracticeExercisesCh8P1Tester
Code is set to "10, 20, 5"
    Expected: Lock should not open. Code inputted was 23, 33, 38.
The lock was not opened. You inputted "23, 33, 38."
Try again.

    Expected: Lock should open.
The lock is open!
PS C:\Users\saket\Git\CSWork\JAVA\ChapterAssignments\C8EXBakshiSaket>

```