Assignment 2 – Object Oriented Programming and Design

1. The program with the program is that the variable, completelyFilledJars is of type integer, trying to hold a floating point number (double). This can easily be fixed by either:

Casting milk/jarCapacity as an integer.

```
2 * Devante Wilson - 100554361
 3 * October 5, 2015
    * Program determines how many completely filled jars of
 6 * milk can be had.
 8
 9 public class MilkJarCalculator
11⊝
       public static void main(String[] args)
12
           // declare variables
          // gallons
15
          double milk = 5.5;
17
          // gallons
18
          double jarCapacity = 0.75;
          // cast value as type integer
          int completelyFilledJars = (int) (milk / jarCapacity);
          // output result
23
           System.out.println(completelyFilledJars);
24
       }
```

Making completelyFilledJars a variable of type double and rounding the result.

```
19/*
 2 * Devante Wilson - 100554361
    * October 5, 2015
 5
    * Program determines how many completely filled jars of
    * milk can be had.
 6
 9 public class MilkJarCalculator
 10 {
11⊝
       public static void main(String[] args)
 12
 13
           // declare variables
 14
           // gallons
 15
 16
           double milk = 5.5;
 17
           // gallons
 18
          double jarCapacity = 0.75;
          double completelyFilledJars = (milk / jarCapacity);
 19
21 // output result by rounding
 22
           System.out.println(Math.round(completelyFilledJars));
 23
       }
24 }
```

```
2.

☑ MilkJarCalculator.java
☑ SmallerNumber.java ☒ ☑ DistanceConverter.java
       1⊖/*
        2 * Devante Wilson - 100554361
       3 * October 5, 2015
        5 * Program prompts the user for two integers.
        6 * The smaller of the two will be printed.
       9 // import packages
      10 import java.util.Scanner;
      11
      12 public class SmallerNumber
      13 {
      149
             public static void main(String[] args)
      15
              {
       16
                  // declare variables/objects
      17
                  int num1, num2;
      18
                  Scanner scan = new Scanner (System.in);
      19
      20
                  // prompt user for input
      21
                 System.out.println("Please your first integer.");
      22
                 num1 = scan.nextInt();
      23
                 System.out.println("Please your second integer.");
       24
                  num2 = scan.nextInt();
       25
      26
                  // close Scanner object
      27
                  scan.close();
      28
      29
                // determine smaller integer using min method
      30
                  System.out.println("The smaller integer is " + Math.min(num1, num2));
      31
              }
      32 }
```

```
2 * Devante Wilson - 100554361
 3 * October 5, 2015
 5 * Program determines how many feet are in 3.5 yards
 * as well as how many inches are in 3.5 yards.
 9 public class DistanceConverter
 10 {
 119
       public static void main(String[] args)
 12
 13
           // declare variables
 14
           final int FEET CONVERSION FACTOR = 3, INCHES CONVERSION FACTOR = 12;
 15
           final double YARDS = 3.5;
 16
           double feet = YARDS * FEET CONVERSION FACTOR;
           double inches = feet * INCHES CONVERSION FACTOR;
 17
 18
 19
           // output results
 20
          System.out.println(YARDS + " yards are " + feet + " feet");
 21
           System.out.println(YARDS + " yards are " + inches + " inches");
 22
        }
 23 }
```

4.1

```
■ MilkJarCalculator.java
                     SmallerNumber.java
                                          Distance
  10/*
  2 * Devante Wilson - 100554361
  3 * October 5, 2015
  4
     * Program models a door object
  6
  7
  8 public class Door
  9 {
 10⊖
        public static void main(String[] args)
 11
         {
 12
13
         }
14 }
```

4.2

```
10/*
    * Devante Wilson - 100554361
    * October 5, 2015
 5
    * Program models a door object
 6
    */
 7
 8 public class Door
 9 {
 10
        // declare objects/variables
11
       private String name, state;
12
13⊖
        public static void main(String[] args)
14
        {
15
16
17 }
```

```
2 * Devante Wilson - 100554361
 3 * October 5, 2015
 5 * Program models a door object
 8 public class Door
 9 {
10
       // declare objects/variables
k11
       private String name, state;
12
      // declare state methods
13
149
      public void close()
15
16
17
18
19⊖
       public void open()
20
21
22
<u>23</u>
24⊖
       public static void main(String[] args)
25
26
27
28 }
```

Devante Wilson October 5th, 2015 100554361

4.4

```
10/*
 2 * Devante Wilson - 100554361
 3 * October 5, 2015
 5 * Program models a door object
 8 public class Door
 9 {
10
        // declare objects/variables
11
       private String name, state;
 12
       // state method
 13
 14⊖
      public void close()
 15
16
           state = "close";
 17
 18
 19
       // state method
 20⊖
       public void open()
 21
       {
 22
           state = "open";
 23
       }
 24
 25
       // constructor for Door class
26⊖
        public Door(String name, String state)
 27
 28
 29
        }
 31⊖
       public static void main(String[] args)
 32
33
 34
        }
35 }
```

```
5 * Program models a door object
8 public class Door
9 {
10
      // declare objects/variables
11
      private String name, state;
12
13
      // state method
149
     public void close()
15
16
          state = "close";
18
      // state method
19
20⊝
      public void open()
21
22
          state = "open";
23
      }
24
2.5
      // constructor for Door class
26⊖
       public Door(String name, String state)
27
28
```

// accessor method for door name

// accessor method for door state

public static void main(String[] args)

public String getName()

public String getState()

return state;

return name;

29 30

31 32[©]

33

34

35 36 37

38⊖

40

41 42 43⊖ {

2 * Devante Wilson - 100554361 3 * October 5, 2015

```
8 public class Door
 9 {
10
       // declare objects/variables
11
       private String name, state;
12
       // state method
13
140
     public void close()
15
16
          state = "close";
17
       }
18
19
       // state method
20⊝
       public void open()
21
22
          state = "open";
23 }
24
       // constructor for Door class
25
260
       public Door(String name, String state)
27
28
29
       }
30
      // accessor method for door name
31
32⊖
       public String getName()
33
       {
34
           return name;
35
36
      // accessor method for door state
37
38⊖
     public String getState()
39
       -{
40
          return state;
       }
41
42
43
       // mutator method for door name
     public void setName(String newName)
440
45
46
         name = newName;
47
       }
48
49
       // mutator method for door state
50⊖
       public void setState(String newState)
51
      {
52
          state = newState;
53
       }
54
     public static void main(String[] args)
55⊖
56
       {
57
58
       }
59 }
```

```
2 * Devante Wilson - 100554361
 3 * October 5, 2015
   * Program models a car's fuel usage
6 */
8 public class Car
9 {
10
       // declare variables/objects
       double gasLeft, efficiency, distance, spentGas;
       double initialGas = 0;
12
13
       // constructor - specifies fuel efficiency
14
       public Car(int e)
15⊜
16
17
           this.efficiency = e;
18
19
20⊖
       /* method simulates driving the car for certain distance,
          reducing the amount of gasoline in the fuel tank.*/
       public void drive (double distance)
24
           spentGas = distance / efficiency;
26
       // method returns the current amount of gasoline in the fuel tank
28⊖
       public double getGasInTank()
29
30
           gasLeft = initialGas - (distance/efficiency);
31
           return gasLeft;
32
33
34
      // adds gasoline in the tank
35⊜
       public void addGas(double a)
36
37
           this.initialGas = a;
38
39
       public static void main(String[] args)
40⊖
41
42
43
44 }
```

```
2
    * Devante Wilson - 100554361
 3
    * October 5, 2015
 4
    * Program is tester class for Car.java
 5
 6
 7
 8 public class CarTester
 9 {
10⊖
       public static void main(String[] args)
11
12
           Car myHybrid = new Car (50); // 50 miles per gallon
13
           myHybrid.addGas (20); // Tank 20 gallons
14
           myHybrid.drive (100); // Drive 100 miles
15
16
           // initialize gasLeft
17
           double gasLeft = myHybrid.getGasInTank();
18
19
           // print result
20
           System.out.println(gasLeft);
21
       }
22 }
```

```
19/*
 2 * Devante Wilson - 100554361
 3 * October 5, 2015
 4
 5 * Program designs a tuning circuit for a given frequency.
 6 * It prints the required inductance value
 7
    * and the range of frequencies to which a circuit
 8
   * can be tuned by varying capacitance
9 */
10
11 // import packages
12 import java.util.*;
13
14 public class TuningCircuit
15 {
16⊖
       public static void main(String[] args)
17
18
           // declare and initialize variables
19
           double capacitance = 0,
20
              capacitanceMin = 0,
               capacitanceMax = 0,
21
22
              frequency = 0,
23
              frequencyMin = 0,
24
               frequencyMax = 0,
25
              inductance = 0;
27
           Scanner scan = new Scanner(System.in);
28
29
           // prompt user for values
30
           System.out.println("Please enter a frequency:");
31
           frequency = scan.nextDouble();
32
33
          System.out.println("Please enter a minimum capacitance:");
34
           capacitanceMin = scan.nextDouble();
35
36
           System.out.println("Please enter a maximum capacitance:");
37
           capacitanceMax = scan.nextDouble();
38
39
           // close Scanner object
40
           scan.close();
41
           // calculate results
42
43
           capacitance = Math.sqrt(capacitanceMin * capacitanceMax);
           inductance = 2 * Math.PI / (Math.pow(frequency, 2) * capacitance);
44
           frequencyMin = (2 * Math.PI) / (Math.sqrt(inductance * capacitanceMax));
45
46
          frequencyMax = (2 * Math.PI) / (Math.sqrt(inductance * capacitanceMin));
47
48
           // output results
49
           System.out.println("The required inductance is: " + inductance + " H");
50
           System.out.println("The range of frequencies is from: " + frequencyMin + " Hz to " + frequencyMax + " Hz");
51
       }
52 }
```

```
10/*
 2 * Devante Wilson - 100554361
3 * October 8, 2015
 4 *
5 * Program calculates the force on a pair
 6 * of charged particles.
 7 */
 8
 9 // import packages
10 import java.util.Scanner;
12 public class Coulombs
13 {
        public static void main(String[] args)
14⊖
15
16
            // declare and initialize variables
17
           Scanner scan = new Scanner(System.in);
           double q1 = 0, q2 = 0, r = 0, force = 0;
18
19
20
           // prompt user for input
           System.out.println("Please enter charge Q1 in Coulombs");
21
           q1 = scan.nextDouble();
23
24
           System.out.println("Please enter charge Q2 in Coulombs");
           q2 = scan.nextDouble();
26
27
          System.out.println("Enter the distnace between the charges (in metres)");
28
          r = scan.nextDouble();
29
           // close Scanner object
30
31
            scan.close();
32
33
            // calculate result
34
           force = (q1*q2)/(4*Math.PI*8.854e-12*Math.pow(r,2));
35
36
            // output result
37
           System.out.println("The electric force between the two charged particles is\n" + Math.round(force));
38
        }
39 }
```

```
10/*
 2 * Devante Wilson - 100554361
 3 * October 10, 2015
 4 *
 5 * Program calculates the liquid temperature of liquid
 6 * contained in copper pipes based on a thermistor.
 7
 8
 9 //import packages
10 import java.util.Scanner;
11
12 public class Thermistor
13 {
14⊖
       public static void main(String[] args)
15
16
           // declare/initialize variables
17
           Scanner scan = new Scanner(System.in);
18
           final int R_NAUGHT = 1075; // ohms
19
           final int T_NAUGHT = 85; // celsius
20
           final int BETA = 3696; // 3969 kelvin -> celsius (-273)
21
           double resistance = 0, liquidTemp = 0;
22
23
           //prompt user for resistance
24
           System.out.println("Please enter the thermistor resistance (in ohms):");
25
           resistance = scan.nextDouble();
26
27
           // close Scanner object
28
           scan.close();
29
30
           // calculate results
31
           liquidTemp = (BETA*T NAUGHT)/(T NAUGHT*Math.log(resistance/R NAUGHT)+BETA);
32
33
           //output result
34
           System.out.println("The liquid temperature in degrees C is: " + liquidTemp);
35
36 }
```