

Your Name: \_\_\_\_\_

### Lab 1a: to be done in class

For this lab, you will create a class called Model. Include a Javadoc comment above the class including a description, @author tag. Here are the attributes of a Model that we are concerned with for the purpose of this project:

- first name
- last name
- height            (inches – use int)
- weight            (pounds – use double)
- can travel        (true or false)
- smokes            (true or false)
- occupation        (static; value is "modeling")

Choose appropriate data types and descriptive names for the fields (instance variables), and declare them. Be sure to specify that they are private.

Write **three** constructors for this class (please see the NOTE FOR CONSTRUCTORS on the next page first). The first is the no-arg constructor which does nothing. The second constructor will expect *the first six fields* to be passed in as parameters. Choose descriptive names for the parameters. Use the parameters to initialize the fields, but only if they follow these rules (i.e., do not store parameter values which fail to meet the following criteria):

- first name and last name must be 3 to 20 characters long or else they won't be stored
- height must be 24 to 84 inches or it won't be stored
- weight must be 80 to 280 pounds or it won't be stored

The third constructor takes parameters for only the first name, last name, height in inches, and weight in pounds; this constructor will set `canTravel` to **true** and `smokes` to **false**. Include a Javadoc comment with `@param` tags for each of the parameters above the three constructors.

#### NOTE FOR CONSTRUCTORS:

Make the mutator methods **final**, as follows:

```
public final void setCanTravel(boolean canTravel) {
    this.canTravel = canTravel;
}

public final void setFirstName(String firstName) {
    if((firstName != null) && (firstName.length() >= 3) && (firstName.length() <= 20)) {
        this.firstName = firstName;
    }
}
```

In the second and third constructors, **call the mutator method for each field instead of using an assignment statement**, as in this example:

```
public Model(String firstName, String lastName, int heightInches,
double weightKg, boolean traveler, boolean smoker) {

    setFirstName(firstName);
    setCanTravel(traveler);           // or setCanTravel(true);
    etc...
}
```

Write an accessor (“get”) method **for each instance variable**. Include Javadoc comments with `@return` tags above each method. Write a mutator (“set”) method **for each field**.

Include Javadoc comments with @param tags above each method, and only store the parameter if it meets the respective criteria listed above.

Also add the following accessor methods, which do exactly what they say:

```
public String getHeightInFeetAndInches() // e.g. "5 feet", or
                                           // e.g. "5 feet 1 inch", or
                                           // e.g. "5 feet 6 inches", or

public long getWeightKg();                // e.g. 103 (Math.round() the number)

public static String getOccupation()
```

Also add the following mutator methods:

```
public final void setWeight(long kilograms)

public final void setWeight(double pounds)

public final void setHeight(int feet, int inches)

public final void setHeight(int inches)
```

Also add the following method: **public void printDetails():**

Which prints in the following format, exactly:

```
Name: Susan Smith
Height: 70 inches
Weight: 120 pounds
Does not travel
Does smoke
```

or

```
Name: Tiger Woods
Height: 72 inches
Weight: 190 pounds
```

Does travel  
Does not smoke  
etc...

**NOTE:** this method *must* call your own object's accessor methods; do not access the instance variables directly. For example:

```
public void printDetails() {  
    System.out.println("Name: " + getFirstName() + " " + getLastName());  
    System.out.println("Height: " + getHeightInches() + " inches");  
    System.out.println(" Weight: " + Math.round(getWeightPounds()) + " pounds");  
    if(canTravel == true){  
        System.out.println("Does travel ");  
    }else{  
        System.out.println("Does not travel ");  
    }  
  
    if(smokes == true){  
        System.out.println("Does smoke ");  
    }else{  
        System.out.println("Does not smoke ");  
    }  
}
```

Create a **Driver** class (class with the main method) to test your project by compiling your class and then creating a Model object. Call each mutator method to be sure it is storing (or ignoring, if the criteria is not met) the correct value. Call each accessor method to be sure it is returning the correct value.

Demonstrate your completed project to your instructor. After that, your paper will be signed and you can go home. Lab 1b (below) is due at the next lesson.

Checked by: \_\_\_\_\_

**Lab 1b:**

Continuing from lab 1a (above), add some data and methods as described below.

Add the constants below for conversion factors and bonus rates. There must be no “magic numbers” in your code.

```
public static final int INCHES_PER_FOOT           = 12;
public static final int BASE_RATE_DOLLARS_PER_HOUR = 60;
public static final int TALL_INCHES              = 67;
public static final double THIN_POUNDS           = 140.0;
public static final int TALL_THIN_BONUS_DOLLARS_PER_HOUR = 5;
public static final int TRAVEL_BONUS_DOLLARS_PER_HOUR = 4;
public static final int SMOKER_DEDUCTION_DOLLARS_PER_HOUR = 10;
```

Also add the following methods:

- **public int calculatePayDollarsPerHour()**

Calculates a model's per-hour pay rate and returns the amount. Everyone receives the hourly BASE\_RATE\_DOLLARS\_PER\_HOUR. Tall, thin models (both tall and thin) receive an hourly bonus of TALL\_THIN\_BONUS\_DOLLARS\_PER\_HOUR. Tall means TALL\_INCHES inches or more, thin means less than THIN\_POUNDS pounds. Models who are willing to travel receive an extra TRAVEL\_BONUS\_DOLLARS\_PER\_HOUR per hour. However, models who smoke have SMOKER\_DEDUCTION\_DOLLARS\_PER\_HOUR per hour deducted from their pay.

- **public void displayModelDetails()**

Displays all the model's information on the screen. It must invoke methods `convertHeightToFeetInches()` and `calculatePayDollarsPerHour()`. Example display:

Name: Susan Smith  
Height: 5 feet 10 inches  
Weight: 120.0 pounds  
Travels: yep  
Smokes: nope  
Hourly rate: \$69

**or**

Name: Susan Smith  
Height: 5 feet 10 inches  
Weight: 120.0 pounds  
Travels: yep  
Smokes: nope  
Hourly rate: \$69

In addition to the above method, create an overloaded method called **public void displayModelDetails(boolean metricUnits)** that takes a boolean, which specifies whether the weight should be in metric (kg and cm) or imperial (lb and inches) units when it is displayed. The overloaded method therefore could display as above, or else as:

Name: Susan Smith  
Height: 178 cm  
Weight: 54 kg  
Travels: yep  
Smokes: nope  
Hourly rate: \$69

(note: rounded from 177.8: use **Math.round()**)  
(note: rounded from 54.4)

**or**

Name: Susan Smith  
Height: 178 cm  
Weight: 54 kg  
Travels: yep  
Smokes: nope  
Hourly rate: \$69

This take-home lab is due next class. Finish it before next class (on your own) and bring it in person so your instructor can review it with you during the lab period after the lecture.

Checked by: \_\_\_\_\_