# *10 1Programming I (420-B10-HR)*

# *Lab 6 – Defining a Simple Class*

Date assigned: Tuesday, September 29, 2015

Date due: **Tuesday, September 29, 2015**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

1. create a Java program using stepwise refinement;
2. define a simple Java class;
3. declare instance variables;
4. define a simple Java method;
5. return a value from a method to a calling object;
6. declare a Java object;
7. instantiate a Java object;
8. execute an instance method for an object;
9. understand the difference between a reference variable and a primitive variable.

**To be handed in:**

1. Your ***username\_*B10\_L06\_Simple\_Class** foldershould be zipped and uploaded to **Moodle.**

**To Start:**

1. Log on to **Moodle**, go to the **Programming I** course page and complete the **Lab 6 Terminology and Review** quiz.
2. Start **Eclipse**. Use your **H:\420-B10\Labs** folder as the workspace.
3. Create a **New Java Project** called ***username*\_B10\_L06\_Simple\_Class**.

# Creating a Java Class

**Purpose**: To learn to create a Java class.

We are going to create and test the **TemperatureUser** program. The following class diagram shows the classes, instance variables and methods for the program. Refer to it as you work through the lab.

**To Do**:

## Create a new class called **Temperature** in the **simpleClass** package. It should **not** include a **main()** method. Add a comment to the closing brace as shown here.

public class Temperature

{

} // Temperature class

## Add a JavaDoc comment block at the top of the class before the class header. It should have the following layout:

/\*\*

\* <p>Course: 420-B10 Programming I</p>

\* <p>Lab 6: Defining a simple class</p>

\* <p>Description: represents a Temperature in both Celsius and Fahrenheit.</p>

\* @author ***your name***

\*/

***Add Instance Variables:***

## Add the following JavaDoc comment and **celsius** declaration after the opening brace of the **Temperature** class:

/\*\* The Celsius value of the temperature \*/

private double celsius;

**Format of an instance variable declaration:**

**private *typeid* *variableName*;**

## Add the **fahrenheit** instance variable with an appropriate JavaDoc comment after the **celsius** declaration.

***Add the methods:***

## Add the stub methods[[1]](#footnote-1) **setCelsius()** and **getCelsius()** methods to the **Temperature** class with the corresponding JavaDoc comments:

/\*\* Assign a Celsius temperature and calculate the Fahrenheit

\* temperature for the object

\* @param c the double precision Celsius temperature of the object

\*/

public void setCelsius(double c)

{

} // setCelsius(double)

/\*\* Return the Celsius temperature of the object

\* @return the double precision Celsius temperature \*/

public double getCelsius()

{

return 0.0;

} // getCelsius()

**Format of a method header:**

**public *resultTypeid* *methodName(parameter list)***

**Format of a method parameter:**

***typeid* *parameterName***

## Add the **setFahrenheit()** and **getFahrenheit()** stub methods to the Temperature class with appropriate JavaDoc comments.

## Correct any errors in your class.

# Creating a Java Method

***Purpose:*** To learn how to code a Java method.

To Do:

## Code the body of the **setCelsius** method:

**public void setCelsius(double c)**

**{**

**// Initialize the Celsius instance variable**

**celsius = c;**

**// Calculate the Fahrenheit equivalent**

**fahrenheit = 9 \* celsius / 5 + 32;**

**} // setCelsius(double)**

**Explanation:**

**Celsius = c;**

- The assignment statement is used to assign the value of the formal parameter (**c**) to the instance variable (**Celsius**).

**Fahrenheit = 9 \* Celsius / 5 + 32;**

- The expression on the right of the = sign (**9 \* Celsius / 5 + 32**) is calculated and assigned to the instance variable, **Fahrenheit**.

**The format of an assignment statement is:**

***variableName = expression;***

## Replace the statement to return 0.0 in the **getCelsius()** method with a statement to return the value of the **celsius** instance variable:

**public double getCelsius()**

**{**

**return celsius;**

**} // getCelsius()**

**Explanation:**

**return Celsius;**

- The **Celsius** instance variable is returned to the calling class.

**The format of a return statement is :**

**return *expression;***

## Save your changes and recompile the **Temperature** class. (Note: In Eclipse, a class is automatically compiled when you save it.)

## Code the **setFahrenheit()** method of the **Temperature** class. The **setFahrenheit()** method should set the **Fahrenheit** temperature to the parameter value and calculate the equivalent **Celsius** temperature. The formula is:



## Modify the **getFahrenheit()** method to return **fahrenheit**.

## Save your changes and recompile the **Temperature** class.

# Declaring and Instantiating Objects

Purpose: Learn how to declare and instantiate objects in Java.

To Do:

## Create a class called **TemperatureUser**.It should include a **main()** method. Add comment to the closing braces for main() and the class.

## Add a JavaDoc comment block at the top of the class similar to the one you added to the **Temperature** class. The description should say: “uses the Temperature class to convert from Celsius to Fahrenheit and vice versa.”

## In the **main()** method of the **TemperatureUser** class, declare an object of the **Temperature** class called **fahr:**

**public static void main(String[] args)**

**{**

**// Declare a Temperature object called fahr**

**Temperature fahr;**

**} // main()**

**Explanation:**

**Temperature fahr;**

- A memory location has been reserved for the reference variable **fahr**. This is the name that will be used for the object when it is instantiated.

**The format of an object declaration is:**

***className objectName;***

## Instantiate the **fahr** object:

public static void main(String args[])

{

// Declare a Temperature object called fahr

Temperature fahr;

**fahr = new Temperature();**

} // main()

**Explanation:**

**fahr = new Temperature();**

- A new **Temperature** object has been created and its location stored in the reference variable **fahr**.

**The format of an object instantiation is:**

***objectName =* new *className*(*constructor arguments*);**

## Declare and instantiate a second **Temperature** object called **cels**.

## Run the **TemperatureUser** class. There should not be any output.

# Call methods

Purpose: Learn how to call methods for Java objects.

To Do:

## In the **main()** method of the **TemperatureUser** class, add the following statement after the object instantiation statements (before } // main()):

**fahr.setFahrenheit(212);**

**Explanation:**

**fahr.setFahrenheit(212);**

- asks the **fahr** object to change the Fahrenheit temperature to 212

**Format of a method call:**

***objectName.methodName*(*method arguments*)**

## Add a statement to set the **Celsius** temperature of the **cels** object to 20.

## Add the following statements after the last statement you added:

**// print the Fahrenheit temperature for fahr**

**System.out.println ("fahr object: ");**

**System.out.print ("\tFahrenheit: ");**

**System.out.println(fahr.getFahrenheit());**

**// print the Celsius temperature for fahr**

**System.out.print ("\tCelsius: ");**

**System.out.println (fahr.getCelsius());**

**Explanation:**

**System.out.print ("\tFahrenheit: ");**

- prints the word *Fahrenheit* in the output screen after a tab (\t).

**System.out.println (fahr.getFahrenheit());**

- asks the **fahr** object to return the Fahrenheit temperature by calling the **getFahrenheit()** method.

- prints the value returned in the output screen and moves to a new line after printing it.

## Rerun the program. There should be three lines of output.

## Add any statements necessary to the **main()** method to print out the following messages with the values from the **cels** object. Use the **print** and **println** statements and appropriate method calls. To print a blank line, code **System.out.println()**

fahr object:

Fahrenheit: 212.0

Celsius: 100.0

cels object:

Celsius: 20.0

Fahrenheit: 68.0

## Rerun the program.

## Add statements to the **main()** method to read in a Celsius temperature:

### add a **Scanner** object called **keyboard**

### add a double precision variable called **fahrTemperature**

### add statements before the closing brace of the **main()** method to prompt for and read a Fahrenheit temperature into **fahrTemperature**.

## Rerun your program. If you enter 0 at the prompt, it should look like:

fahr object:

Fahrenheit: 212.0

Celsius: 100.0

cels object:

Celsius: 20.0

Fahrenheit: 68.0

Enter a Fahrenheit temperature: 0

## Add statements to change the Fahrenheit temperature of the **fahr** object to **fahrTemperature** and display both the Fahrenheit and Celsius temperatures from the **fahr** object:

### call the **setFahrenheit()** method for the **fahr** object to set the Fahrenheit temperature to **fahrTemperature**.

### reprint the Fahrenheit and Celsius temperatures for the **fahr** object.

## Rerun the program. If you enter 0 at the prompt, the output should be:

fahr object:

Fahrenheit: 212.0

Celsius: 100.0

cels object:

Celsius: 20.0

Fahrenheit: 68.0

Enter a Fahrenheit temperature: 0

fahr object:

Fahrenheit: 0.0

Celsius: -17.77777777777778

## Run the program using different Fahrenheit temperatures.

## Right click on the source and select **Source🡪Format** to format the source layout.

**Finish This Section!!!**

# Practice

***Purpose:*** Practise creating a Java class with private instance variables and public methods and practice creating objects and calling methods for the objects.

***To Do:***

## Create new class called **NumberAdder** in the **simpleClass** package. It should not have a **main()** method.

## Add two private integer instance variables called **num1** and **num2** to the **NumberAdder** class.

## Add accessors and mutators for **num1** and **num2** to the **NumberAdder** class. (The accessors should be called **getNum1()** and **getNum2()** and the mutators should be called **setNum1()** and **setNum2()**. The accessors should return the appropriate instance variable. The mutators should set the value of the instance variable to the parameter value.)

## Add a method called **getSum()** that has no parameters and returns an integer. In the body of the method return the sum of **num1** and **num2**.

## Create another new class called **TestAdder** in the **simpleClass** package. It should contain a **main()** method.

## Add statements to the **main()** method to:

### Create a Scanner object called **keyboard**

### Create two instances of the **NumberAdder** class, named **adder1** and **adder2**

### Declare two integer variables, **number1** and **number2**

### Prompt the user for two integers and read them into **number1** and **number2**

### Use the **setNum1()** method for **adder1** to set **num1** to **number1** and the **setNum2()** method to set **num2** to **number2**

### Prompt the user for two more integers and assign them to **adder2**'s numbers

### print the result of the **getSum()** method for each object.

## Test your program. Your output should be similar to:

Enter two integers: 10 15

Enter two more integers: 100 200

Adder1: The sum of 10 and 15 is 25

Adder2: The sum of 100 and 200 is 300

# To Finish:

## Zip your ***username*\_B10\_L06\_Simple\_Class**folder and upload to **Moodle**.

## Do the **Lab 6 Review Quiz** on **Moodle**.

1. A stub method is a method with an empty body [↑](#footnote-ref-1)