**Lab 7**

**Static Methods and Debugging**

**The following exercises are to be completed during lab class. If you do not have time to finish during lab, they must be completed before the beginning of the following lab session.**

***Set-Up***

* Create a new project in your Eclipse workspace named: **Lab07**
* In the *src* folder, create a package named: **edu.ilstu.lab07**
* Import into this new package all of the .java files in the lab07 folder in the T:\it168 Matsuda\Labs folder.

**Part I**

Implement the **TemperatureConverter** class you designed for pre-lab. Make sure to create the proper Javadoc comments. Use the following equations for your calculations:

|  |  |  |  |
| --- | --- | --- | --- |
| **From** | **To Fahrenheit** | **To Celsius** | **To Kelvin** |
| Celsius (C or o) | (C \* 9/5) + 32 | N/A | C + 273.15 |
| Fahrenheit (F) | N/A | (F - 32) \* 5/9 | (F - 32) \* 5/9 + 273.15 |
| Kelvin (K) | (K - 273.15) \* 9/5 + 32 | K - 273.15 | N/A |

Demonstrate your understanding of static methods by writing a **TemperatureApp** class that has a main method. You will ask the user for a temperature in Celsius, Fahrenheit and Kelvin. Then you will display the original temperatures entered and the converted temperatures with proper labels.

**Sample run (values in bold are inputs):**

Enter temperature in Celsius: **32**

In Fahrenheit: 89.6

In Kelvin: 305.15

Enter temperature in Fahrenheit: **28**

In Celsius: -2.22

In Kelvin: 270.93

Enter temperature in Kelvin: **55**

In Celsius: -218.15

In Fahrenheit: -360.67

**Part II - Debugging**

We're going to learn about the debugger by using it to fix a broken program, so we need to start with a broken program. You will use what you learned in the debugging document you read for prelab, i.e. “Using the Eclipse Debugger.docx”.

Read through the TestBankAccount.java code. Note that the comments specify what should be happening in the program.

Run the TestBankAccount program. Answer the questions in this section as you go through the steps. You will upload this document updated with the answers.

**1) How does what actually happens differ from what should happen?**

1. What should be the balance for start100 object?

100 should be the balance of start100 object.

1. What should be the balance for startZero?

It should be zero.st

**2) What kind of error is this (syntax, run-time, or logic)?**

We're now going to track down the cause of the problem using the debugger. **Do not** try to solve the problem just by looking at the code. The point of this exercise is to learn how to use the debugger to solve more difficult problems than this one.

We want to **step through** the program one step at a time, in order to figure out what is wrong with the code.

1. Set the breakpoint on the first line of executable code, line 12.
2. Begin debugging. Run the application in Debug mode using one of the ways from the debugging document.
3. We're pretty sure that the creation of the BankAccount object works so let's step over that using **Step Over** button. **Click on the Step Over button twice**.

**3) What line is highlighted now in the code window?**

Up in the Variables pane of the window, which is the top right pane, there should now be a startZero variable and start100 variable of type BankAccount. Notice that there's an arrow next to each variable. **Click on the arrow sign for start100**. You will see the variable balance (with red square) having the value 0.

**4)** Click the green Arrow  to finish running the application. **What is shown as the balance for the account that should have a starting balance of $100?**

**A value of 0 is shown for the account that should have a starting balance of 100.**

**5) Let us look inside the Bank Account class.**

1. Your application should still be in Debug perspective. From this perspective, run the application again in debug mode.
2. Step over line 12 that creates the startZero instance of BankAccount.
3. Line 13 should be highlighted. This time, click on **Step Into.** You are taken into constructor inside BankAccount class. The method header is highlighted.
4. Look in variables window.
   1. There is a variable balance with 100 assigned to it. This is the parameter from the method header.
   2. You will also see the keyword this (with green circle). Expand the arrow. You will see the instance variable balance which is assigned a value 0. Keep this expanded.
5. Click the **Step Over**, line 21 is highlighted. This line is not executed yet. Click **Step Over** again to execute line 21.

**6) The instance variable should have been assigned the value from the parameter. What is happening here? How should this be fixed?**

1. Stop the application by clicking the red square in the console window. While on Debug perspective, correct the problem in the constructor. Click Save button (or use Control-S, or select File 🡪 Save).
2. Run the application again in Debug mode again. You will be back in TestBankAccount.
3. Repeat steps 2 - 5 from section **5).**
4. **What is the value now for the instance variable balance? Is this correct?**

**The value for the Instance variable now is 100 and yes it is the correct value.**

**7) You will debug the rest of the application and fix the problems you find.**

1. Figure out why the third withdrawal does not work.

It does not work because it is not updating the instance variable balance after making the withdrawal.

1. Why is there not enough money in the balance to withdraw $74.50? Where is the problem?

In my view there was a logical error in the program as the math for calculating the withdrawal should be done in the if statement rather than before the if statement as if the newBalance is greater than zero which holds true, it withdraws the amount, if it does not holds true it displays that we have insufficient balance.

**8)** Once all the problems are fixed, go back into Java perspective and run the application as a Java Application (click the green arrow) to make sure all the outputs are correct.

***To Be Submitted***

The following files should be zipped together into a file called Lab07.zip and submitted to ReggieNet by the beginning of your next lab.

* TemperatureConverter.java
* TemperatureApp.java
* Update “Lab 7.docx”

|  |  |
| --- | --- |
|  |  |
|  |  |