Due:
Activity (in-lab): Monday, October 3, 2016 by the end of lab

Goals:

By the end of this activity you should be able to do the following:

- Understand the basics of the ternery *conditional operator*
- Understand the basics of the *for loop*, *for each loop*, and the *do-while loop*
- Understand the basics of the *switch* statement

Description:

In this activity you will create two classes. Temperatures will hold a set of integer values representing daily temperatures. TemperatureInfo will allow users to interact with the Temperatures class.

Directions:

Part 1: Temperatures: instance variable, method stubs

- Create a class called Temperatures, which will hold a set of integer values representing daily temperatures.
- Add an instance variable with the name *temperatures* to your class that is of type ArrayList with generic type Integer.
- Add method stubs for the following methods**.**
    - The constructor takes an ArrayList of integer values

```
public Temperatures(ArrayList<Integer> temperaturesIn) {

}
```

    - getLowTemp: takes no parameters; returns an integer value
    - getHighTemp: takes no parameters; returns an integer value
    - lowerMinimum: takes an int parameter; returns an integer value
    - higherMaximum: takes an int parameter; returns an integer value
    - toString: no parameters; returns a String

Part 2: Temperatures: constructor, getLowTemp

- In your constructor, set *temperatures* equal to *temperaturesIn*.
- In getLowTemp, first return 0 if the ArrayList is empty:

```
if(temperatures.isEmpty()) {
   return 0;
}
```

- Now iterate through the entire list and find the lowest temperature:

```
int low = temperatures.get(0);
for (int i = ______; i < _______________; i++) {
   if (temperatures.get(i) < low) {
      low = temperatures.get(i);
   }
}
```

- Finally, after the loop, return the lowest temperature:

```
return low;
```

Part 3: getHighTemp

- In getHighTemp, again return 0 if there are no temperatures in the ArrayList:

```
if (temperatures.isEmpty()) {
   return 0;
}
```

- This time, use a *for each* loop to iterate through the list of temperatures to find the highest temperature.

```
int high = temperatures.get(0);
for (Integer temp : temperatures) {
   if (_______ > high) {
      high = _______;
   }
}
return high;
```

- Add code to the toString method to return a string containing the low and high temperatures (hint: make a method call to getLowTemp and getHighTemp):

```
public String toString() {
   return "\tTemperatures: "+ temperatures
         + "\n\tLow: " + getLowTemp()
         + "\n\tHigh: " + getHighTemp();
}
```

Part 4: lowerMinimum & higherMaximum

- The lowerMinimum method takes an int value and returns the parameter if it is lower than the value returned by getLowTemp. Otherwise, it returns the return of getLowTemp.

```
public int lowerMinimum(int lowIn) {
   return lowIn < getLowTemp() ? lowIn : getLowTemp();
}
```

- The higherMaximum method takes an int value and returns the parameter if it is greater than than the value returned by getHighTemp. Otherwise, it returns the return of getHighTemp.

```
public int higherMaximum(int highIn) {
   return _____ ? _______ : _____;
}
```

- Test your methods in the interactions pane:
  Note that you should <u>not</u> use the generic <Integer> when declaring the ArrayList in interactions.

```
▶ import java.util.ArrayList;
▶ ArrayList tempList = new ArrayList();
▶ tempList.add(34);
```

```
tempList.add(52);
tempList.add(36);
tempList.add(65);
Temperatures temps = new Temperatures(tempList);
temps.getLowTemp()
34
temps.getHighTemp()
65
temps.lowerMinimum(33)
33
temps.lowerMinimum(35)
34
temps.higherMaximum(64)
65
temps.higherMaximum(67)
67
```

Part 5: TemperatureInfo

- Download the canvas file for this activity and save it in same folder as the classes for this activity. After you have created and compiled part or all the *main* method described below, use the canvas file in conjunction with debugger and/or run the program in canvas mode.
- Create a class called TemperatureInfo with a main method. Declare and instantiate a Scanner object called userInput that reads from System.in. Declare and instantiate an ArrayList with generic type Integer called tempsList. Remember to import the Scanner and ArrayList classes.
- Create the following *do while* loop that will read in temperature values, one per line, and add each to tempsList until the user presses enter with no value to indicate that there are no more temperatures to be input. After all of the temperatures have been read in and added to tempsList, create a Temperatures object with the tempsList:

```
String tempInput = "";
do {
   System.out.print("Enter a temperature (or nothing to end list): ");
   tempInput = userInput.nextLine().trim();
   if (!tempInput.equals("")) {
      tempsList.add(Integer.parseInt(tempInput));
   }
} while (!tempInput.equals(""));

Temperatures temps = new Temperatures(tempsList);
```

- Create a menu by using a *do while* loop that contains a *switch* statement to select among user choices for the following: `[L]ow temp, [H]igh temp, [P]rint, [E]nd` where 'L' prints the low temperature, 'H' prints the high temperature, 'P' prints the Temperatures object temps, 'E' ends the the program (i.e., ends the *do while* loop).

```
char choice= 'E';
do {
   System.out.print("Enter choice - [L]ow temp, [H]igh temp, [P]rint, [E]nd: ");
   choice = userInput.nextLine().toUpperCase().charAt(0);
   switch (choice) {
      case 'L':
         System.out.println("\tLow is " + temps._____________);
         break;

      case 'H':
         System.out.println("\tHigh is " + ________.getHighTemp());
         break;

      case 'P':
         System.out.println(temps);
         break;

      case 'E':
         System.out.println("\tDone");
         break;

      default:
         System.out.println("\tInvalid choice!");
   }

} while (choice != 'E');
```

Run your program as below to test its output:

```
  ----jGRASP exec: java -ea TemperatureInfo
▶▶ Enter a temperature (or space to end list): 34
▶▶ Enter a temperature (or space to end list): 56
▶▶ Enter a temperature (or space to end list): 78
▶▶ Enter a temperature (or space to end list): -10
▶▶ Enter a temperature (or space to end list): 95
▶▶ Enter a temperature (or space to end list):
▶▶ Enter choice - [L]ow temp, [H]igh temp, [P]rint, [E]nd: L
         Low is -10
▶▶ Enter choice - [L]ow temp, [H]igh temp, [P]rint, [E]nd: H
         High is 95
▶▶ Enter choice - [L]ow temp, [H]igh temp, [P]rint, [E]nd: P
         Temperatures: [34, 56, 78, -10, 95]
         Low: -10
         High: 95
▶▶ Enter choice - [L]ow temp, [H]igh temp, [P]rint, [E]nd: E
         Done


  ----jGRASP: operation complete.
```

- Finally, be sure to run your program in Canvas mode using the canvas file you downloaded. Remember that you can control the speed with the delay slider in canvas window or debug tab.