Project 1: Temperature Converter

For the first project, you must simply type in a Java program and submit it. While this may seem mundane, there are five specific goals for this project:

- Gain access to a working installation of the BlueJ IDE (preferably at home ***and*** in the lab);
- Become familiar with how to use the BlueJ IDE;
- Become familiar with how to use the Curator online submission system;
- Gain exposure to actual Java code – how it looks;
- Gain an appreciation for how precise Java demands that you be in your typing.

Hopefully, by dealing with these issues now, they should be less of a shock when you have to actually author your first program in Project 2. For now, do not worry about trying to figure out how the program works. Some features of this program will be covered early in the semester, some very late in the semester, and some not at all. Do not expect to understand everything.

This program converts temperature scales. ***It must be named*** **"Program1.java".** It accepts input from a file, called **"celsius_in.txt"**, which holds five temperatures in the Celsius scale. It then converts the temperatures to Celsius and writes appropriate output to another file, called **"fahrenheit_out.txt"**. There are a few points in the program where you must substitute your personal information for what is written in the code below. Those places in the code are highlighted. For example, you should replace occurrences of my name (Mir Farooq Ali) with your own name. Similarly, replace the dates highlighted in the code with the dates when you work on your project, etc. Test files will be posted on the website so that you can test your program's correctness. I also suggest that you create your own test input files and check your program with them. The posted test files are not meant to be a guarantee of a completely correct program.

Evaluation:

Your program will be submitted to the Curator system for evaluation of correctness (details are given below). You will have five submissions to allow you an opportunity to correct mistakes that your own testing has not found. **You will not be given more submissions for any reason!** Your submitted program will be assigned a grade based on the correctness of your program and evaluation of your program by a TA. The grade you receive in the email from the Curator is your correctness grade. The grade for the project for future assignments be computed by deducting any points for style and design from the correctness grade. Therefore, the correctness grade is the *maximum* that your program will receive. **The TA will always grade the *first* submission that has the highest points.** No exceptions will be made, so be sure that you have done everything you need to before submitting. Note: Check your grade a few minutes after submitting, **If you get "N/A" as a grade, do not submit again until you get a numeric grade.**

Submitting your Program

You will submit this assignment to the Curator System (read the *Student Guid*e), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guid*e. You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted. The *Student Guide* can be found at:

`http://www.cs.vt.edu/curator/`

Use of Comments and Program Format

The comments used in this assignment are typical of what will be expected for all your projects.

Program Compilation

Your program must compile and run under JDK 1.4.

Pledge

Every program submission for this class must include an Honor Code pledge. Specifically, you **must always** include the following pledge statement in the header comment for your program:

```
/****************************************************************************
 *
 * On my honor:
 *
 * - I have not discussed the Java language code in my program with
 * anyone other than my instructor or the teaching assistants
 * assigned to this course.
 *
 * - I have not used Java language code obtained from another student,
 * or any other unauthorized source, either modified or unmodified.
 *
 * - If any Java language code or documentation used in my program
 * was obtained from another source, such as a text book or course
 * notes, that has been clearly noted with a proper citation in
 * the comments of my program.
 *
 * - I have not designed this program in such a way as to defeat or
 * interfere with the normal operation of the Curator System.
 *
 * Your Name
 * Your PID
****************************************************************************/
```

Failure to include this pledge in a submission is a violation of the Honor Code.

Program starts here

```
import java.io.*;
import java.text.*;

/****************************************************************************
 * Project 1 for CS 1054 Fall 2003
 *  Temperature Converter Program
 *
 *  File:           Program1.java
 *  Programmer:     Mir Farooq Ali
 *  Java Platform: JDK 1.4.2_01
 *  Compiler:       BlueJ - Version 1.3.0
 *  Last modified: September 11, 2003
 *
 *  Purpose:   This program takes input from a file named "celsius_in.txt",
 *   which contains five lines, each containing a single decimal
 *   value representing a temperature in the Celsius scale.
 *   This program converts those five temperatures to Fahrenheit
 *   scale, and sends results to a file called "fahrenheit_out.txt".
 ****************************************************************************/
/****************************************************************************
 *
 * On my honor:
 *
 * - I have not discussed the Java language code in my program with
 * anyone other than my instructor or the teaching assistants
 * assigned to this course.
 *
 * - I have not used Java language code obtained from another student,
 * or any other unauthorized source, either modified or unmodified.
 *
 * - If any Java language code or documentation used in my program
 * was obtained from another source, such as a text book or course
 * notes, that has been clearly noted with a proper citation in
 * the comments of my program.
 *
 * - I have not designed this program in such a way as to defeat or
 * interfere with the normal operation of the Curator System.
 *
 * Mir Farooq Ali
 * miali
 ****************************************************************************/

public class Program1 {

    /*************************************************************************
    *  Project 1 for CS 1054 Fall 2003
    *  Temperature Converter Program
    *
    *  Programmer:    Mir Farooq Ali
    *  Java Platform: JDK 1.4.2_01
    *  Compiler:      BlueJ - Version 1.3.0
    *  Last modified: September 11, 2003
    *
    *  Purpose:  This program takes input from a file named "celsius_in.txt",
    *      which contains five lines, each containing a single decimal
    *      value representing a temperature in the Celsius scale.
    *      This program converts those five temperatures to Fahrenheit
    *      scale, and sends results to a file called "fahrenheit_out.txt".

    * Method:        main
    * Programmer     Mir Farooq Ali
    * Arguments:     String array; ignored
    * Returns:       void
    * Throws:        FileNotFound & IO Exceptions
    * Last Modified: September 11, 2003
    *
    * Purpose: This method declares variables, reads and decodes
    *          input, and generates and writes appropriate output.
    ***********************************************************************/

    public static void main(String args[]) throws Exception{

        // String objects holding the input and output filenames
        String myInFileName = "celsius_in.txt";
        String myOutFileName = "fahrenheit_out.txt";

        // objects to read input and write output
        BufferedReader myInput;
        PrintStream myOutput;

        // String objects representing the five lines read from the input
        // stream.
        String cTempStr1, cTempStr2, cTempStr3, cTempStr4, cTempStr5;

        // five decimal Celsius temperatures, as determined from input
        double cTemp1, cTemp2, cTemp3, cTemp4, cTemp5;

        // the five result Strings to be written as output
        String msg1, msg2, msg3, msg4, msg5;


        //**********************************************************************
        //*               SET UP THE BUFFERED INPUT STREAM READER              *
        //**********************************************************************
        // if input is from the *keyboard* use this line......
        // myInput = new BufferedReader(new InputStreamReader(System.in) );
        //
        // ....but if input is from a *file* use these lines
        FileInputStream myFileIS = new FileInputStream(new File(myInFileName) );
        myInput = new BufferedReader(new InputStreamReader(myFileIS) );
        //**********************************************************************

        //**********************************************************************
        //*              SET UP THE PRINT STREAM / OUTPUT STREAM              *
        //**********************************************************************
        // if output if to the *screen* us this line......
        // myOutput = System.out;
        //
        // ....but if output is to a *file* use these lines
        FileOutputStream myFileOS=new FileOutputStream(new File(myOutFileName));
        myOutput = new PrintStream(myFileOS);
        //**********************************************************************

        // read the lines
        cTempStr1 = myInput.readLine();
        cTempStr2 = myInput.readLine();
        cTempStr3 = myInput.readLine();
        cTempStr4 = myInput.readLine();
        cTempStr5 = myInput.readLine();

        // decode the lines; store values as doubles
        cTemp1 = Double.valueOf(cTempStr1).doubleValue();
        cTemp2 = Double.valueOf(cTempStr2).doubleValue();
        cTemp3 = Double.valueOf(cTempStr3).doubleValue();
        cTemp4 = Double.valueOf(cTempStr4).doubleValue();
        cTemp5 = Double.valueOf(cTempStr5).doubleValue();

        // determine output messages from the temperature values
        msg1 = generateMsg(cTemp1);
        msg2 = generateMsg(cTemp2);
        msg3 = generateMsg(cTemp3);
        msg4 = generateMsg(cTemp4);
        msg5 = generateMsg(cTemp5);

        // write the headers out
        myOutput.println("Programmer: Mir Farooq Ali");
        myOutput.println("CS1054: Project 1 Fall 2003");
        myOutput.println();

        // write the messages out
        myOutput.println(msg1);
        myOutput.println(msg2);
        myOutput.println(msg3);
        myOutput.println(msg4);
        myOutput.println(msg5);

        // close the streams
        myFileIS.close();
        myFileOS.close();
    }

    /**************************************************************
    * Method:         generateMsg
    * Programmer      Mir Farooq Ali
    * Arguments:      a Celsius temperature
    * Returns:        a string telling the Celsius and
    *                 corresponding Fahrenheit temperatures
    * Last Modified:  September 11, 2003
    *
    * Purpose:      This method takes a Celsius temperature
    *        and uses it to produce a message telling
    *        how that temperature relates to its corresponding
    *        Fahrenheit temperature.
    ***************************************************************/

    static String generateMsg(double cTemp){
        DecimalFormat df = new DecimalFormat();
        df.setMinimumFractionDigits(2);
        df.setMaximumFractionDigits(2);
        String begin = " Celsius= ";
        String end = " Fahrenheit.";
        String fStr = df.format(cTemp);
        String cStr = df.format(celsToFahrs(cTemp));
        while (fStr.length()<7) fStr = " " + fStr;
        while (cStr.length()<7) cStr = " " + cStr;

        return fStr + begin + cStr + end;
    }


    /*************************************************************
    * Method:         celsToFahrs
    * Programmer      Mir Farooq Ali
    * Arguments:      a Celsius temperature as a double
    * Returns:        the corresponding Celsius temperature as a double
    * Last Modified:  September 11, 2003
    *
    * Purpose: This method does arithmetic temperature
    *      conversion from Celsius to Fahrenheit scales.
    *************************************************************/

    static double celsToFahrs(double cels){
        return (cels*9.0)/5.0 + 32;
    }
}
```