CIS 255 • Java Programming
Laboratory Assignment 1: TextPad and the JDK

In this laboratory assignment you will learn how to enter Java code into a TextPad window, how to compile your code, and how to execute the resulting program. Remember: you may use any Java environment when coding on your own computer, but it may be helpful to know how to use TextPad in conjunction with the JDK in the classroom.

In the instructions for each laboratory assignment, a text box labeled with the first symbol is a warning; with the second symbol, a hint; and with the third, a note.

Step 1: Configure Your Computer

If you want to be able to compile and run your Java programs on your own computer, you'll need to install Sun's Java Development Kit (JDK). Additionally, you'll probably want to use a text editor that highlights Java syntax; TextPad is one such program. The accompanying document **Java Configuration for Windows** (**CIS 255 Configuration Instructions.pdf**) describes the process of installing both the JDK and TextPad.

Step 2: Launch TextPad and Use Java Highlighting

Once TextPad is installed, a shortcut to TextPad should appear in alphabetical order in the All Programs list in the Start Menu. Launching TextPad opens a blank window in which you may enter text.

When software is installed on the laboratory computers, the shortcuts may not appear in the All Programs list automatically. There should be a folder on the desktop named CIS that includes shortcuts to software commonly used in CIS courses.

Although TextPad is not free software, the dialog that pops up when you launch the program dismisses itself after a few seconds, so there are no immediate consequences to using the program without paying for it. Still, it might be nice to show your appreciation for the program by purchasing a license for your own computer that will get rid of that annoying dialog.

Because TextPad supports syntax highlighting for several different languages, it does not highlight your text in any particular language until you save the document with a particular file extension. Select the command **Save As** from the **File** menu. In the dialog that appears, navigate to the location where you want your code to be saved (preferably on a USB flash drive), enter the file name **Payroll.java**, and select **Java (*.java)** in the **Save as type:** combo box.

The file name you use must correspond to a particular element in your code (including case) for the program to compile successfully, as mentioned in chapter 2.

Step 3: Enter Your Code

Enter the code from Code Listing 1-1 in chapter one in your textbook. The code also appears below:

```
public class Payroll
{
    public static void main(String[] args)
    {
        int hours = 40;
        double grossPay, payRate = 25.0;

        grossPay = hours * payRate;
        System.out.println("Your gross pay is $" + grossPay);
    }
}
```

Be careful to enter the code exactly as it appears in the textbook and / or in the figure above.

TextPad (and many other coding environments) highlight Java syntax in various ways. In TextPad, keywords appear in blue, brackets and parentheses appear in red, and text within quotation marks appears in green. If the highlighting on your screen does not match what is shown above, you may have misspelled a keyword, or you may have forgotten the closing character of a character pair.

Step 4: Compile Your Code

As mentioned in the lecture, the process of compiling the program converts Java source code (in a .java file) into byte code (in a .class file) that runs in the Java Virtual Machine (JVM). To determine if there are any syntax errors in your source code, and to generate the byte code once your source code is free of syntax errors, select the item **Compile Java** from the **External Tools** submenu of the **Tools** menu.

If the Java-related items in the previous figure do not appear in your External Tools submenu, there may be a configuration issue with your computer. The TextPad web site includes a guide to solving problems with the integration of the JDK tools into the External Tools submenu. If you are working on a laboratory computer and cannot access the JDK tools, ask your instructor or a laboratory technician for assistance.

The results of the build process will appear in the **Tool Output** area below your code. If the Tool Output area indicates that there are syntax errors in your code, you may double-click the line that describes the syntax error to see the line that contains the error (or, in some cases, the line after the error). In the following example, I omitted the closing parenthesis before the semicolon on line 9:

```
F:\CIS 255 Code\Payroll.java:9: ')' expected
                System.out.println("Your gross pay is $" + grossPay;
                                                                    ^
```

The process of removing syntax errors from a program may take some time. You may want to correct a single error in the code and run the Compile Java command again to see how many syntax errors remain in the list. Once you receive a message in the Tool Output area that the tool completed successfully, you're ready to test your program.

Step 5: Execute the Byte Code

There are two additional commands in the External Tools submenu related to executing Java programs. For the majority of the semester, you'll use the second command, **Run Java Application**, but at the end of the semester, you'll also use the command **Run Java Applet**.

A Java application is a standalone program that can be executed locally on a computer. A Java applet is designed to be embedded in a web page. The syntax of a Java applet will vary somewhat from the syntax of a Java application. Initially you'll be writing Java applications.

Once you select Run Java Application, a console window appears. The program displays the result of the calculation with the labeling specified in the output statement:

```
Your gross pay is $1000.0
Press any key to continue...
```

For now this program is not particularly interesting: it displays a set result based on values hard-coded into the source code. As you progress into chapter 2, you'll discover how to allow users to specify their own values for variables as they run your programs.

Step 6: Submit Your Code

You are not required to attach your compiled byte code (the file that ends in `.class`) for any assignment. Only your source code file (the file that ends in `.java`) is required.

Follow the instructions on the course Web site to attach and submit your source code.

Make sure you complete the submission process! If you attach your file without submitting the assignment, your file will not be received!