The following exercise is intended to introduce the student to the programming environment and the power of the interactive debugger. The information contained herein will be used throughout the course.

**Starting Microsoft Visual J++**

Under Microsoft Visual Studio 6.0, select Microsoft Visual J++ 6.0. The program will automatically open the New Project window. Select *Console Application* and enter your initials for the name.

**Creating a New Class**

Select *Project, Add Item*, which will open up the adjacent window. Create a new class named “ProgInfo.java”.

Select Open.
Editing Your First Class

In your Project Explorer window, double click on the class you just created. This will bring that class up into the central editing window. Enter the following code for the “ProgInfo.java” class:

```java
public class ProgInfo
{
    ProgInfo()
    {
        System.out.print( "This program computes the factorial of the number 
" );
        System.out.print( "passed to it via the command line at a DOS prompt. \n" );
        System.out.print( "The syntax used on the command line is: 
" );
        System.out.print( "C:\\PROG1 <integer> 
" );
        System.out.print( "This will compute the factorial of the integer. 
" );
        System.out.print( "Example: 
" );
        System.out.print( "C:\\PROG1 4 
" );
        System.out.print( "The program will compute and display the factorial 
" );
        System.out.print( "of 6 which is 720. 
" );
    }
}
```

Don't forget to save your work.

Creating Your Main Class

Right click on “Class1.java” in the Project Explorer window and rename it to “Prog1.java”. Then double click on “Prog1.java” to bring it into the central
import ProgInfo;

public class Prog1
{
public ProgInfo intro;

public static void main(String[] args)
{
int count = 1, entered = 1, result = 1;
ProgInfo intro = new ProgInfo(); //Display program info from ProgInfo class

if (args.length == 1) // Check for command line values
    count = entered = Integer.parseInt(args[0],10);
else
{
    System.out.print( "***** Error *****  

" );
    System.out.print( "Please Enter:      

" );
    System.out.print( "C:\2811P1 <integer> 
" );
    System.exit(0);
}

while (count > 1) // Do the factorial of count.
{
result *= count;
count--;
}

System.out.println("The factorial of "+entered+" is "+result);
}

"Build" the Program

Go to the Build menu and select Build (Ctrl-Shift-B).

The system will then compile and link your program. It will take a moment. Wait to see if any errors appear in the task window. (If you have no errors, remove a semicolon at the end of a line and try rebuilding to force an error.)

When you have a compiler error, double click on the line in the bottom window that indicates the error. The line in your source code on which the compiler detected an error will be indicated by an arrow to the left of the line.

Run the Program

Once you've fixed the compiler errors, you are ready to run. (Warnings should be heeded, but they generally do not stop a program from linking however, some warnings can be ignored.)

Go to the Debug menu and select Start (F5). Your program should run (with an error) by first opening a window and displaying the execution error. This is because the program has been designed to run from the command line by entering the program name and an
argument. For example: To generate factorial of six (6!), you would type the following at the command line:

C:\JView Prog1 6

Since your program did not have a way of obtaining the 6 (since we did not run from the command line) we need to tell the Visual Studio interactive environment to use 6 as our argument. (Later you will test the program at the command line.)

### Edit Settings Used When Executing the Program

Go to the `Project` menu, and select `(project name) Properties`, which will open the adjacent window. Select the `Launch` tab, and then the `Custom` radio button. For `Program:`, enter `JView.exe`. For `Arguments:`, enter `/p /cp:p "<JAVAPACKAGES>" Prog1 6`. This will cause your program to be executed using the `JView.exe` program as an interpreter and with the single argument of 6. Select OK.

### Debugging

Put the cursor on the line that says “count = entered = Integer.parseInt(args[0],10);” select `Insert/Remove Breakpoint` from the `Build` menu. Note the dot placed to the left of the line selected. Run the program again.

Note that the program runs up to and stops just before executing the line that you are on. It is important that you realize that the breakpoint occurs just before the line is executed.

Several new toolbars should have been opened. Explore each toolbar by allowing your cursor to rest on top of each one for a second. This should bring up a description of the function of each button. Click on the button called “Locals”, which will open a window at the bottom of your screen to display all local variables in for your program and their current values. Click the toolbar button that indicates “Step Over” several times while watching the text and variable windows.

At this point, you should experiment with the debug capabilities of the environment. Try clicking on any variable while the program is stopped in the middle of a debug run. A small window will appear and give you the value of the variable.
Maximum Limitation

Run the program at the command line using different numbers for the argument. Find out where the accuracy of the program is exceeded (answer will go negative).

Running From the Command Line

If you can open a DOS window and find your program, you can run it from the prompt directly by entering “JView PROG1 6”. You may need to first change the current directory to the location of the “JView.exe” file on your computer.