Project 18: Matrix Multiplication

The primary objective in this project will be to write code that will multiply the following two matrices and produce the indicated answer.

$$\begin{pmatrix} 1 & 2 & -2 & 0 \\ -3 & 4 & 7 & 2 \\ 6 & 0 & 3 & 1 \end{pmatrix} \times \begin{pmatrix} -1 & 3 \\ 0 & 9 \\ 1 & -11 \\ 4 & -5 \end{pmatrix} = \begin{pmatrix} -3 & 43 \\ 18 & -60 \\ 1 & -20 \end{pmatrix}$$

If you are not familiar with the intricacies of matrix multiplication, Appendix AA is supplied to provide a brief overview of the subject. In fact, the above example is used in that appendix.

Create a project called *MatrixStuff* that consists of two classes. These two classes will be called *Tester* and *MatrixMult* and will meet the specifications listed below.

The *Tester* class (*main* method):

1. Hard code the *int a[ ][ ]* array so as to be comprised of the 3 X 4 matrix on the left in the example above.

2. Hard code the *int b[ ][ ]* array so as to be comprised of the 4 X 2 middle matrix in the example above.

3. Call a *static* method of the *MatrixMult* class called *mult* in which we pass the *a* and *b* arrays (matrices) as arguments and receive back an integer array as the product matrix.

4. Print the product matrix.

5. The output of *main* should appear as follows:

```
-3   43

18   -60

1     -20
```

The *MatrixMult* class:

1. No constructor.

2. Create a single *static* method called *mult* that receives two *int* arrays (matrices) as parameters that are to be multiplied in the order in which they are received.

3. The *mult* method is to return an array that is the product matrix of the two parameter arrays it receives.

4. The code in the *mult* method is to determine the dimensions of the matrices that it receives and set up a "product" array (matrix) to be returned with the appropriate dimensions.

5. The code in *mult* should be general so as to adapt to any two matrices to be multiplied; however, for the sake of simplicity, you may assume that the matrices received as parameters **are** always compatible with each other for multiplication.

6. The code in the *mult* method will multiply the two incoming matrices so as to correctly produce each element of the product matrix.