Module One - Assignment Submission

The Assignment

The assignment for Module One was very limited in its request. It provided a picture of the word JAVA spelled out in ASCII art, and asks the user to write a program which will display the pictured output.

Design

In order to create the output, it is clear that the user needs to in some way print out the ascii art in the given format to the standard output. There are no requirements for the underlying data structres to be used, or how the program should operate outside of the final requirement of displaying the text. Normally, I would want to do something a bit more interesting to save the manual labor of printing each letter out in a static program, for example a program which is able to take in a word from the user and then display the letters in a similar format as to what is given to us, but for the scope of this project I figured that just printing out the word line by line should be sufficient.

As such, my first task was to determine the spacing requriements of the word JAVA. My implementation was in an ubuntu bash shell instead of in command prompt, so I don't have the exact same font availabe as that which is on DOS. As such, I had to estimate the spacing using a ruler.

Though definitely all coming from my own assumptions, I thought I would need at least 7 newline only prints in order to isolate the word JAVA as it is on the screen, and then four prints which will print out the letters with the following spacings:

1. J - Four characters tall, with 5 characters across
2. A - Four characters tall, with 7 characters across
3. V - Four characters tall, with 7 characters across

To do this, I would write out the ascii art for JAVA in a consistently spaced text editor, and then wrap the lines in their own print to out statments in order to display the text

After this, I would print a few more newlines to get the correct spacing from the bottom of the screen.

Implementation

```
/**
 * Assignment One
 *
 * This program prints to the standard output an ascii-art of
 * the word java, as shown below:
 *
 *         J    A    V     V    A
 *         J   A A    V   V    A A
 *     J   J  AAAAA    V V    AAAAA
 *      J J  A     A    V    A     A
 *
 * @author: Duncan Parke
```

```
 */

public class ModuleOne
{
    /**
     * Program Entry Point
     */
    public static void main( String [] args)
    {
        // Display JAVA in ascii art
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println("    J    A    V     V    A   ");
        System.out.println("    J   A A    V   V    A A  ");
        System.out.println("J   J  AAAAA    V V    AAAAA ");
        System.out.println(" J J  A     A    V    A     A");
        System.out.println(); // Newline
        System.out.println(); // Newline
        System.out.println(); // Newline
    } // end main()

} // end ModuleOne
```

Output From Execution

```
    J    A    V     V    A
    J   A A    V   V    A A
J   J  AAAAA    V V    AAAAA
 J J  A     A    V    A     A
```