Program Style and Documentation Specifications

Required Documentation
Every program should begin with a comment block that contains information about you and the program you have written. A comment block is a multi-line comment, usually set off with the liberal use of the "*" character to make it stand out. The template below illustrates a comment block, and shows the information that should appear in a comment block:

,strlen

Description of the program, including:
A brief statement or statements describing what the program does; for example:
This program computes a car's gas mileage

A list of the inputs and outputs (if any)
Example:
Inputs: miles driven, gas consumed (in gallons)
Outputs: gas mileage, in miles per gallon

In addition to the comment block, your documentation should include comments placed throughout your code that explain various sections. At a minimum, you should include:

- a brief description of each variable or object at declaration; example:
  ```
  float miles; // distance driven -- from user
  float gallons; // amount of fuel consumed -- input
  float mpg; // miles per gallon -- calculated
  int counter; // loop counter
  JFrame myWindow // main display window
  ```

- a brief description of each method (besides main); example:
  ```
  public int Square (int num);
  // returns the square of its argument
  ```

- brief explanation of the purpose of all major control structures adjacent to their code; example:
  ```
  for (counter=0; counter < MAX; ++counter) // fill number array
  {
    (body of loop appears here)
  }
  ```

- clarifying comments adjacent to any code you feel might require further explanation; example:
  ```
  g.setColor(Color.RED); // set color for balloon
  g.fillOval(20, 20, 40, 50); // draw balloon
  ```
General Notes on Java programming style

1. Java is a case sensitive language. This means that a word written in ALL CAPS and the same word written in lowercase or mixed case characters is, as far as the compiler is concerned, an entirely different word. Make sure that all Java language keywords (names of data types, control structures, etc.) are written as lowercase words; the compiler won't recognize them otherwise.

2. Follow Java API naming conventions:
   - In general, make defined constants all uppercase, and variable, object and method names mixed case starting with a lowercase letter, class names mixed case starting with an uppercase character.
   - Use descriptive names for all program components you declare; the name of each variable, method, constant, class, object etc. should reflect its purpose. This will help make your code self-documenting.
   - If you need to use more than one word in a name, use capital letters to set off the beginning of new words; example: primeFactor
   - Class names start with capital letters; all other identifiers (except constants) should start with lowercase letters.

3. Indentation and white space: use these to clarify the structure of your program. Some specifics:
   - Use a new indentation level for each pair of begin/end braces you use. For example:
     ```java
     class myClass {
       public static void main(String [] args) {
         // this is the second level
         {
           // this is the third level
         } // ends innermost block
       } // ends main
     } // ends myClass
     ```
     Use blank lines between discrete program sections (for example, between declarations and regular program statements, between control structures, and between functions.

4. If a statement is too long to fit on a single line, break it at a logical point. Do not break a line in the middle of a String literal or identifier; it is ok to break a line at an operator or between method arguments. Some examples:

   ```java
   JOptionPane.setInputDialog (inputWindow,
      "Please enter your full name in the form: First Last:";
   
   String longString = "I just want to celebrate" +
     " another day of living...";
   ```