Key Point: *Good programming style and proper documentation make a program easy to read and help programmers prevent errors.*

- Programming style and documentation are as important as coding.
- Good programming style and appropriate documentation reduce the chance of errors and make programs easy to read.

1.9.1 Appropriate Comments and Comment Styles

- At the beginning of every .java file, include an author identification comment that contains your name, your student id, your e-mail address, the name of the course, the name of section in which you are enrolled, the name of the project, and the date that the project is due as shown below.

```
//-----------------------------------------------------------------------------------------------------------
//Author:       Ms. Petunia Perfect
//Student ID:   *00000001
//E-Mail:       pperfect@uco.edu
//Course:       CMSC 1513 – Beginning Programming
//CRN:          25440, Spring, 2022
//Project:      p01
//Due:          January 24, 2022
//-----------------------------------------------------------------------------------------------------------
```

- Include a summary at the beginning of the program that explains what the program does, its key features, and any unique techniques it uses.
- Comments that begin with // and end at the end of the line are called *line* comments.
- Comments that begin with /*** and end with */ are called *javadoc* comments and can be extracted into an HTML file using the JDK's **javadoc** command. Use javadoc comments for commenting on an entire class or an entire method.

1.9.2 Proper Indentation and Spacing

- Computer professionals always use indentation to format their programs.
- A program that does not have distinct and appropriate indentation is unprofessional.
- Suggestion: A single space should be added on both sides of a binary operator as shown in (a) and not in (b)

```
System.out.println(3 + 4 *4);
```
(a) Good style

```
System.out.println(3+4*4);
```
(b) Bad style

1.9.3 Block Styles

- A block is a group of statements surrounded by braces. There are two popular styles, next-line style and end-of-line style, as shown below.

```
public class Test
{
    public static void main(String[] args)
    {
        System.out.println("Next-line style");
    }
}
```

(a) Next-line style

```
public class Test{
    public static void main(String[] args)    {
        System.out.println("End-of-line style");
    }
}
```

(b) End-of-line style

- Whatever style you choose, stick to it and do not change or invent new styles.
- Our text uses the end-of-line style.