Prerequisite

- State Properties
PROGRAMMING STYLE: THE ART OF PROGRAMMING

- Use Interfaces
- Efficiency
- Document Program
- Avoid Code Repetition
- Giving Least Privilege
- Use Named Constants
Any code in the program removed by the compiler.

Useful for documentation

```
double bmi; //computed by setWeight and setHeight
```

Also for enclosing debugging code

```
// System.out.println(newHeight);
```
WHAT TO COMMENT?

- Any code fragment needing explanation
  - Class
    - Top-level algorithm, author, date modified
  - Variable declaration
    - Purpose, where used, how its value is computed
  - Method declaration
    - params, return value, algorithm, author, date modified
  - Statement sequence
    - Explanation
- Debugging code
**COMMENT TYPES**

### Single line comments

```java
double bmi; // computed by setWeight and setHeight
```

### Single or multi-line arbitrary comments

```
/* recompute dependent properties */
bmi = weight/(height*height);
```

```
/* recompute dependent properties */
bmi = weight/(height*height);
```
REMOVING DEBUGGING CODE

/*
    System.out.println(newHeight); /*debugging statement */
*/

/*
    System.out.println(newHeight); // debugging statement
*/
What to Write in a Comment?

- `double w; // weight` - Bad variable name
- `double weight; // weight` - Redundant
- `double weight;` - Self-commenting
- `double bmi; // computed by setWeight and setHeight` - Useful comment
SELF COMMENTING VARIABLE NAMES

public static final int NUMBER_WITH_LARGE_FACTORIAL = 15;

Eclipse CTRL-SPACE will complete name for you
public static void modularReadAndPrintStrings() {
    String[] strings = readStrings(readNumStrings());
    printStrings(strings);
    printString(strings[0]); // unsafe
}

public static int readNumStrings() {
    System.out.println("Number of Strings:");
    return Console.readInt(); // reads the next line as integer
}

public static String[] readStrings(int numElements) {
    System.out.println("Please enter " + numElements + " strings");
    String[] strings = new String[numElements]; // dynamic array
    for (int elementNum = 0; elementNum < numElements; elementNum++)
        strings[elementNum] = Console.readString();
    return strings;
}
Other programmers who may have to maintain your code

Other programmers using the API provided by your code

They do not have the context of the code, so some comments may not be redundant

API comments useful in Web Documents and Tooltip Messages in Programming Environment

Follow special format so they can be extracted by web doc tools and programming environment
javadoc conventions for headers (API)

/**
 * @author Prasun Dewan
 * For more info see:
 * @link http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html#exampleresult
 */

public interface CommentedBMISpreadsheet {

/**
 * @param newValue is the new value of variable weight not only sets values of weight, but also computes new value bmi and assigns it to variable bmi
 */

public void setWeight(double newValue);

}
In well written code, difficult to find opportunities to comment

Do not comment for the sake of commenting