(2) General Program Design Principles

© 2022, Jongwook Kim. All rights reserved.

Program Hygiene – a.k.a. Code Style Guide

- Use spaces judiciously and consistently
- Indent properly
- Follow a brace style
- Use comments to describe code behavior
- Do not hardcode things that are likely to change
- Follow the naming conventions – use good names
- …

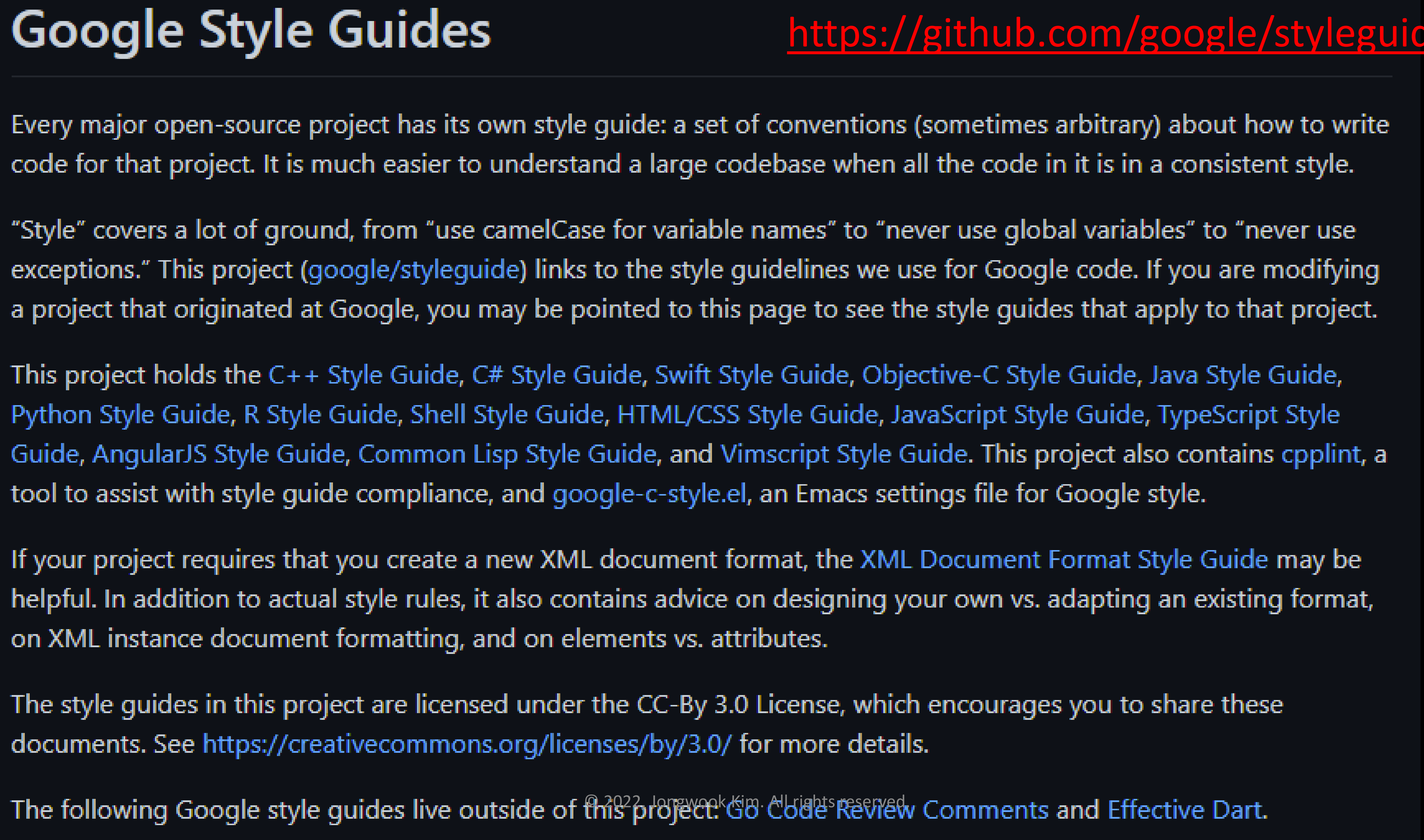
© 2022, Jongwook Kim. All rights reserved.

Why Worry About Program Hygiene?

- Programmers build on top of other's code all the time
  - We should not waste time deciphering what a method does
  - We should not waste time looking for a missing closing brace
  - ...

- Software maintenance takes 50–80% of the overall cost

© 2022, Jongwook Kim. All rights reserved.



Google Style Guides

https://github.com/google/styleguide

Every major open-source project has its own style guide: a set of conventions (sometimes arbitrary) about how to write code for that project. It is much easier to understand a large codebase when all the code in it is in a consistent style.

"Style" covers a lot of ground, from "use camelCase for variable names" to "never use global variables" to "never use exceptions." This project (google/styleguide) links to the style guidelines we use for Google code. If you are modifying a project that originated at Google, you may be pointed to this page to see the style guides that apply to that project.

This project holds the C++ Style Guide, C# Style Guide, Swift Style Guide, Objective-C Style Guide, Java Style Guide, Python Style Guide, R Style Guide, Shell Style Guide, HTML/CSS Style Guide, JavaScript Style Guide, TypeScript Style Guide, AngularJS Style Guide, Common Lisp Style Guide, and Vimscript Style Guide. This project also contains cpplint, a tool to assist with style guide compliance, and google-c-style.el, an Emacs settings file for Google style.

If your project requires that you create a new XML document format, the XML Document Format Style Guide may be helpful. In addition to actual style rules, it also contains advice on designing your own vs. adapting an existing format, on XML instance document formatting, and on elements vs. attributes.

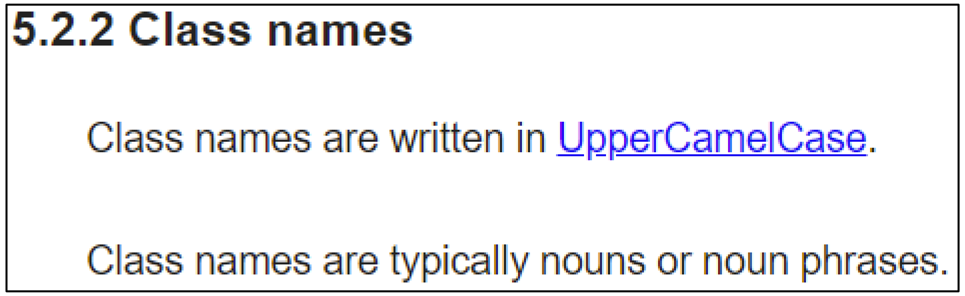
The style guides in this project are licensed under the CC-By 3.0 License, which encourages you to share these documents. See https://creativecommons.org/licenses/by/3.0/ for more details.

The following Google style guides live outside of this project: Go Code Review Comments and Effective Dart.

© 2022. Jongwook Kim. All rights reserved.

Google's Java Style Guide Examples

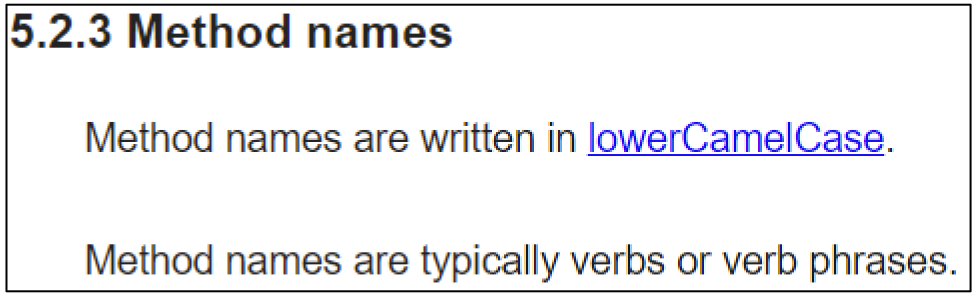
https://google.github.io/styleguide/javaguide.html



5.2.2 Class names

Class names are written in UpperCamelCase.

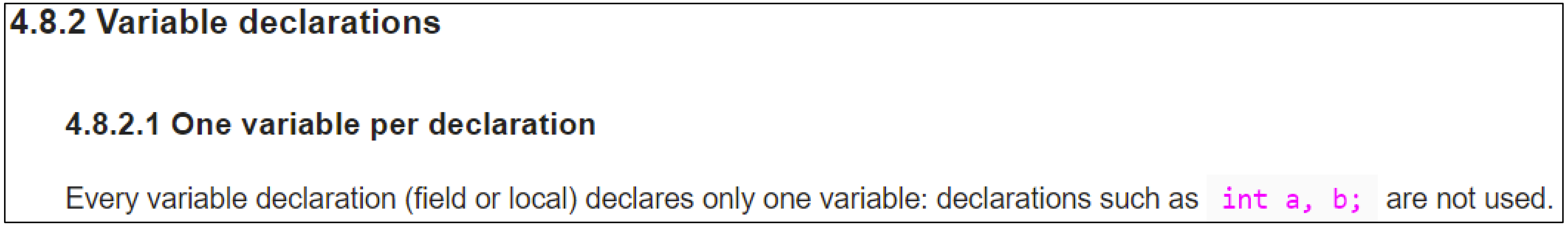
Class names are typically nouns or noun phrases.



5.2.3 Method names

Method names are written in lowerCamelCase.

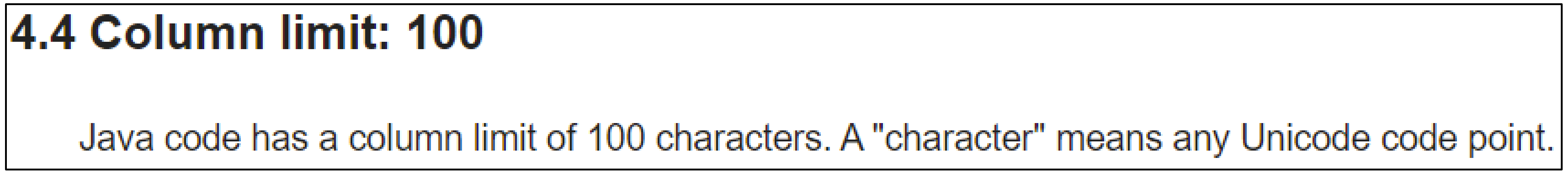
Method names are typically verbs or verb phrases.



4.8.2 Variable declarations

4.8.2.1 One variable per declaration

Every variable declaration (field or local) declares only one variable: declarations such as `int a, b;` are not used.



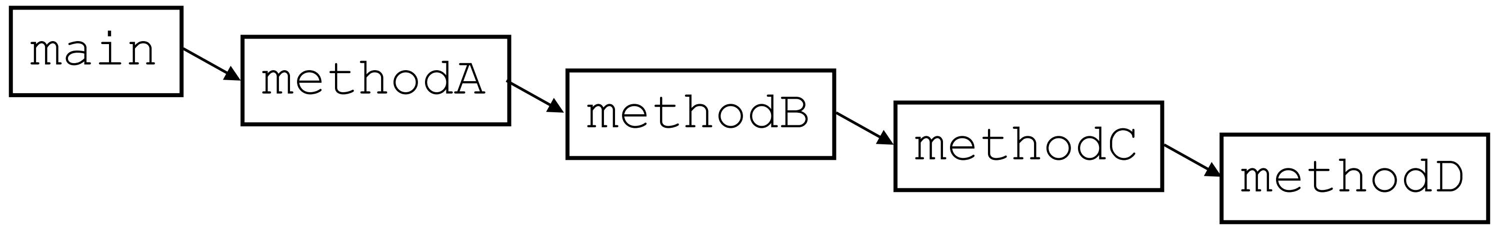
4.4 Column limit: 100

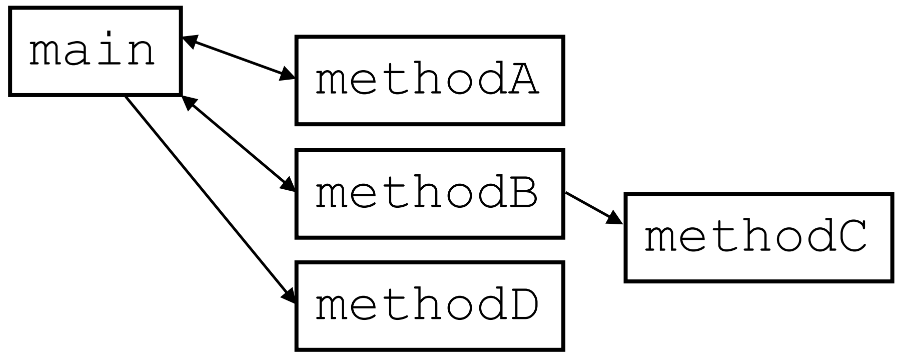
Java code has a column limit of 100 characters. A "character" means any Unicode code point.

© 2022, Jongwook Kim. All rights reserved.

General Program Design Principles

- Variables should be declared/used at the lowest level possible
- Minimize coupling and dependencies between methods





- The main method should read as a concise summary of the tasks performed by the program
- Eliminate redundant code
- ...

© 2022, Jongwook Kim. All rights reserved.

Code Redundancy

© 2022, Jongwook Kim. All rights reserved.

Example #1

© 2022, Jongwook Kim. All rights reserved.

What is bad about the following code?

```
// Calculate total owed, assuming 8% tax / 15% tip
System.out.print("Subtotal: ");
System.out.println(38 + 40 + 30);

System.out.print("Tax: ");
System.out.println((38 + 40 + 30) * 0.08);

System.out.print("Tip: ");
System.out.println((38 + 40 + 30) * 0.15);

System.out.print("Total: ");
System.out.println(38 + 40 + 30 +
                    (38 + 40 + 30) * 0.15 +
                         (38 + 40 + 30) * 0.08);
```

© 2022, Jongwook Kim. All rights reserved.

Code Redundancy

```
// Calculate total owed, assuming 8% tax / 15% tip
System.out.print("Subtotal: ");
System.out.println(38 + 40 + 30);

System.out.print("Tax: ");
System.out.println((38 + 40 + 30) * 0.08);

System.out.print("Tip: ");
System.out.println((38 + 40 + 30) * 0.15);

System.out.print("Total: ");
System.out.println(38 + 40 + 30 +
                     (38 + 40 + 30) * 0.15 +
                          (38 + 40 + 30) * 0.08);
```

© 2022, Jongwook Kim. All rights reserved.

Eliminating Redundant Code (1)

```
// Calculate total owed, assuming 8% tax / 15% tip
int subtotal = 38 + 40 + 30;
System.out.print("Subtotal: ");
System.out.println(subtotal);

System.out.print("Tax: ");
System.out.println(subtotal * 0.08);

System.out.print("Tip: ");
System.out.println(subtotal * 0.15);

System.out.print("Total: ");
System.out.println(subtotal + subtotal * 0.15 + subtotal * 0.08);
```

© 2022, Jongwook Kim. All rights reserved.

Eliminating Redundant Code (2)

```
// Calculate total owed, assuming 8% tax / 15% tip
int subtotal = 38 + 40 + 30;
double tax = subtotal * 0.08;
double tip = subtotal * 0.15;
System.out.print("Subtotal: ");
System.out.println(subtotal);

System.out.print("Tax: ");
System.out.println(tax);

System.out.print("Tip: ");
System.out.println(tip);

System.out.print("Total: ");
System.out.println(subtotal + tip + tax);
```

© 2022, Jongwook Kim. All rights reserved.

Example #2

© 2022, Jongwook Kim. All rights reserved.

What is bad about the following code?

```
public static void main(String[] args) {
    System.out.println("Mix the dry ingredients.");
    System.out.println("Cream the butter and sugar.");
    System.out.println("Beat in the eggs.");
    System.out.println("Stir in the dry ingredients.");
    System.out.println("Set the oven temperature.");
    System.out.println("Set the timer for 10 minutes.");
    System.out.println("Place the cookies into the oven.");
    System.out.println("Allow the cookies to bake.");
    System.out.println("Spread frosting and sprinkles onto the
    cookies.");
}
```

© 2022, Jongwook Kim. All rights reserved.

Unstructured Programming

```
public static void main(String[] args) {
    System.out.println("Mix the dry ingredients.");
    System.out.println("Cream the butter and sugar.");
    System.out.println("Beat in the eggs.");
    System.out.println("Stir in the dry ingredients.");
    System.out.println("Set the oven temperature.");
    System.out.println("Set the timer for 10 minutes.");
    System.out.println("Place the cookies into the oven.");
    System.out.println("Allow the cookies to bake.");
    System.out.println("Spread frosting and sprinkles onto the
    cookies.");
}
```

make batter → bake cookies → frost cookies

© 2022, Jongwook Kim. All rights reserved.

Use methods to structure your program

```
make_batter();
bake_cookies();
frost_cookies();
```

© 2022, Jongwook Kim. All rights reserved.

Consider making a double batch

```
make_batter();
bake_cookeis();
bake_cookeis();
frost_cookies();
```

Consider making 100 batches

```
make_batter();
bake_cookeis();
bake_cookeis();
...
...
...
bake_cookeis();
frost_cookies();
```

© 2022, Jongwook Kim. All rights reserved.

Eliminating Redundant Code

```
make_batter();
for(int i = 0; i < n; i++) {
    bake_cookies();
}
frost_cookies();
```

Example (3)

© 2022, Jongwook Kim. All rights reserved.

What is bad about the following code?

- Write a program to print these figures

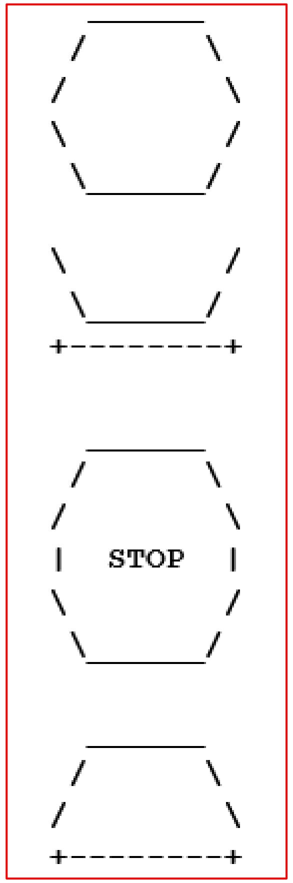
```
  _______
 /       \
/         \
\         /
 \_______/

\         /
 \_______/
+---------+

  _______
 /       \
/         \
|  STOP   |
\         /
 \_______/

  _______
 /       \
/         \
+---------+
```

```java
public class Figure {
    public static void main(String[] args) {
        System.out.println("  _______");
        System.out.println(" /       \\");
        System.out.println("/         \\");
        System.out.println("\\         /");
        System.out.println(" \\_______/");
        System.out.println();
        System.out.println("\\         /");
        System.out.println(" \\_______/");
        System.out.println("+---------+");
        System.out.println();
        System.out.println("  _______");
        System.out.println(" /       \\");
        System.out.println("/         \\");
        System.out.println("|  STOP   |");
        System.out.println("\\         /");
        System.out.println(" \\_______/");
        System.out.println();
        System.out.println("  _______");
        System.out.println(" /       \\");
        System.out.println("/         \\");
        System.out.println("+---------+");
    }
}
```

© 2022, Jongwook Kim. All rights reserved.

Eliminating Redundant Code

```
public class Figures {
    public static void main(String[] args) {
        egg();
        teaCup();
        stopSign();
        hat();
    }

    public static void eggTop() {
        System.out.println("  ______");
        System.out.println(" /      \\");
        System.out.println("/        \\");
    }

    public static void eggBottom() {
        System.out.println("\\        /");
        System.out.println(" \\______/");
    }

    public static void egg() {
        eggTop();
        eggBottom();
        System.out.println();
    }

    public static void teaCup() {
        eggBottom();
        line();
        System.out.println();
    }

    public static void stopSign() {
        eggTop();
        System.out.println("|  STOP  |");
        eggBottom();
        System.out.println();
    }

    public static void hat() {
        eggTop();
        line();
    }

    public static void line() {
        System.out.println("+--------+");
    }
}
```



```
  ______
 /      \
/        \
\        /
 \______/

\        /
 \______/
+--------+

  ______
 /      \
/        \
|  STOP  |
\        /
 \______/

  ______
 /      \
/        \
+--------+
```

© 2022, Jongwook Kim. All rights reserved.

Exercise

© 2022, Jongwook Kim. All rights reserved.

Exercise

- Write a program to print the following figure

```
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

© 2022, Jongwook Kim. All rights reserved.

Skeleton Solution

- Write a program to print the following figure

```
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

```
public class Mirror {
    public static void main(String[] args) {
        line();
        topHalf();
        bottomHalf();
        line();
    }

    public static void topHalf() {
        ...
    }

    public static void bottomHalf() {
        ...
    }

    public static void line() {
        ...
    }
}
```

© 2022, Jongwook Kim. All rights reserved.

topHalf

- Compute spaces and dots from line number

```
012345678901234567
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

```
public static void topHalf() {
    for (int line = 1; line <= 4; line++) {


    }
}
```

line	spaces	-2 * line + 8
1	6	6
2	4	4
3	2	2
4	0	0

line	dots	4 * line - 4
1	0	0
2	4	4
3	8	8
4	12	12

© 2022, Jongwook Kim. All rights reserved.

topHalf Solution

```
public static void topHalf() {
    for (int line = 1; line <= 4; line++) {
        System.out.print("|");

        for (int space = 1; space <= (line * -2 + 8); space++) {
            System.out.print(" ");
        }

        System.out.print("<>");

        for (int dot = 1; dot <= (line * 4 - 4); dot++) {
            System.out.print(".");
        }

        System.out.print("<>");

        for (int space = 1; space <= (line * -2 + 8); space++) {
            System.out.print(" ");
        }

        System.out.println("|");
    }
}
```

© 2022, Jongwook Kim. All rights reserved.

Scaling the mirror

- Modify the Mirror code to be resizable using a class constant
- The current mirror (left) is at size 4; the right is at size 3

```
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

```
#============#
|    <><>    |
|  <>....<>  |
|<>........<>|
|<>........<>|
|  <>....<>  |
|    <><>    |
#============#
```

© 2022, Jongwook Kim. All rights reserved.

Using a Class Constant

- Scale the figure by changing a class constant value

```
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

```java
public class Mirror {
    public static final int SIZE = 4;

    public static void main(String[] args) {
        line();
        topHalf();
        bottomHalf();
        line();
    }

    public static void topHalf() {
        for (int line = 1; line <= 4; line++) {
            ...
        }
    }

    public static void bottomHalf() {
        for (int line = 1; line <= 4; line++) {
            ...
        }
    }
}
```

© 2022, Jongwook Kim. All rights reserved.

topHalf

- The left mirror is at size 4; the right is at size 3

```
012345678901234567
#================#
|      <><>      |
|    <>....<>    |
|  <>........<>  |
|<>............<>|
|<>............<>|
|  <>........<>  |
|    <>....<>    |
|      <><>      |
#================#
```

```
public static void topHalf() {
  for (int line = 1; line <= 4; line++) {
    System.out.print("|");

    for (int space = 1; space <= (line * -2 + 8); space++) {
      System.out.print(" ");
    }

    System.out.print("<>");

    for (int dot = 1; dot <= (line * 4 - 4); dot++) {
      System.out.print(".");
    }

    System.out.print("<>");

    for (int space = 1; space <= (line * -2 + 8); space++) {
      System.out.print(" ");
    }

    System.out.println("|");
  }
}
```

© 2022, Jongwook Kim. All rights reserved.

```
01234567890123
#============#
|    <><>    |
|  <>....<>  |
|<>........<>|
|<>........<>|
|  <>....<>  |
|    <><>    |
#============#
```

Final Solution

```
public static void topHalf() {
    for (int line = 1; line <= SIZE; line++) {
        System.out.print("|");

        for (int space = 1; space <= (line * -2 + (2*SIZE)); space++) {
            System.out.print(" ");
        }

        System.out.print("<>");

        for (int dot = 1; dot <= (line * 4 - 4); dot++) {
            System.out.print(".");
        }

        System.out.print("<>");

        for (int space = 1; space <= (line * -2 + (2*SIZE)); space++) {
            System.out.print(" ");
        }

        System.out.println("|");
    }
}
```

© 2022, Jongwook Kim. All rights reserved.

Another Example

SIZE = 4

```
       /**\
      //**\\
     ///**\\\
    ////**\\\\
   /////**\\\\\
  //////**\\\\\\
 ///////**\\\\\\\
+=*=*=*=*=*=*=*=*+
|.../\....../\...|
|../\/\..../\/\..|
|./\/\/\../\/\/\.|
|/\/\/\/\/\/\/\/\|
|\/\/\/\/\/\/\/\/|
|.\/\/\/..\/\/\/.|
|..\/\/....\/\/..|
|...\/......\/...|
+=*=*=*=*=*=*=*=*+
|\/\/\/\/\/\/\/\/|
|.\/\/\/..\/\/\/.|
|..\/\/....\/\/..|
|...\/......\/...|
|.../\....../\...|
|../\/\..../\/\..|
|./\/\/\../\/\/\.|
|/\/\/\/\/\/\/\/\|
+=*=*=*=*=*=*=*=*+
       /**\
      //**\\
     ///**\\\
    ////**\\\\
   /////**\\\\\
  //////**\\\\\\
 ///////**\\\\\\\
```

SIZE = 5

```
         /**\
        //**\\
       ///**\\\
      ////**\\\\
     /////**\\\\\
    //////**\\\\\\
   ///////**\\\\\\\
  ////////**\\\\\\\\
 /////////**\\\\\\\\\
+=*=*=*=*=*=*=*=*=*=*+
|..../\......../\....|
|.../\/\....../\/\...|
|../\/\/\..../\/\/\..|
|./\/\/\/\../\/\/\/\.|
|/\/\/\/\/\/\/\/\/\/\|
|\/\/\/\/\/\/\/\/\/\/|
|.\/\/\/\/..\/\/\/\/.|
|..\/\/\/....\/\/\/..|
|...\/\/......\/\/...|
|....\/........\/....|
+=*=*=*=*=*=*=*=*=*=*+
|\/\/\/\/\/\/\/\/\/\/|
|.\/\/\/\/..\/\/\/\/.|
|..\/\/\/....\/\/\/..|
|...\/\/......\/\/...|
|....\/........\/....|
|..../\......../\....|
|.../\/\....../\/\...|
|../\/\/\..../\/\/\..|
|./\/\/\/\../\/\/\/\.|
|/\/\/\/\/\/\/\/\/\/\|
+=*=*=*=*=*=*=*=*=*=*+
         /**\
        //**\\
       ///**\\\
      ////**\\\\
     /////**\\\\\
    //////**\\\\\\
   ///////**\\\\\\\
  ////////**\\\\\\\\
 /////////**\\\\\\\\\
```

© 2022, Jongwook Kim. All rights reserved.