"We're flooding people with information. We need to feed it through a processor. A human must turn information into intelligence or knowledge. We've tended to forget that no computer will ever ask a new question."

— Rear Admiral Grace Murray Hopper, Ph.D.
Redundant recipes

Recipe for baking **20** cookies:
- Mix the following ingredients in a bowl:
  - 4 cups flour
  - 1 cup butter
  - 1 cup sugar
  - 2 eggs
  - **40** oz. chocolate chips ...
- Place on sheet and Bake for about **10** minutes.

Recipe for baking **40** cookies:
- Mix the following ingredients in a bowl:
  - 8 cups flour
  - 2 cups butter
  - 2 cups sugar
  - 4 eggs
  - **80** oz. chocolate chips ...
- Place on sheet and Bake for about **10** minutes.
Parameterized recipe

Recipe for baking 20 cookies:
- Mix the following ingredients in a bowl:
  - 4 cups flour
  - 1 cup sugar
  - 2 eggs
  - ...

Recipe for baking N cookies:
- Mix the following ingredients in a bowl:
  - \( \frac{N}{5} \) cups flour
  - \( \frac{N}{20} \) cups butter
  - \( \frac{N}{20} \) cups sugar
  - \( \frac{N}{10} \) eggs
  - \( 2N \) oz. chocolate chips ...
- Place on sheet and Bake for about 10 minutes.

parameter: A value that distinguishes similar tasks.
Parameterization

- **Parameter**: A value passed to a method by its caller.

Example:

```java
public static void main(String[] args) {
    sayPassword(3);
}

public static void sayPassword(int code) {
    System.out.println("The password is: " + code);
}
```
Passing a parameter

Calling a method and specifying values for its parameters

\[ \text{<name> (<expression>)} ; \]

Example:

```java
public static void main(String[] args) {
    sayPassword(42);
    sayPassword(12345);
}
```

Output:

The password is 42
The password is 12345
Parameters and loops

A parameter can guide the number of repetitions of a loop.

```java
public static void main(String[] args) {
  chant(3);
}

public static void chant(int times) {
  for (int i = 1; i <= times; i++) {
    System.out.println("Just a salad...";color="yellow");
  }
}
```

Output:

```
Just a salad...
Just a salad...
Just a salad...
```
How parameters are passed

- In a game of basketball
  - Guard passes ball to center
  - Center does her job (slam dunk)
How parameters are passed

Pretend instead that in a game of basketball:
- Guard clones the ball
- Guard passes that duplicate ball to the center
- Center does her job (slam dunk)

In Java:
- Caller clones the parameter
- Caller passes that duplicate parameter to the method
- Method does its job
How parameters are passed

- When the method is called:
  - The value is stored into the parameter variable.
  - The method's code executes using that value.

```java
public static void main(String[] args) {
    chant(3);
    chant(7);
}

public static void chant(int times) {
    for (int i = 1; i <= times; i++) {
        System.out.println("Just a salad...");
    }
}
```
Common errors

- If a method accepts a parameter, it is illegal to call it without passing any value for that parameter.
  ```java
  chant();  // ERROR: parameter value required
  ```

- The value passed to a method must be of the correct type.
  ```java
  chant(3.7);  // ERROR: must be of type int
  ```
public static void main(String[] args) {
    printNumber(4, 9);
    printNumber(17, 6);
    printNumber(8, 0);
    printNumber(0, 8);
}

public static void printNumber(int number, int count) {
    for (int i = 1; i <= count; i++) {
        System.out.print(number);
    }
    System.out.println();
}

Output:
444444444
171717171717
000000000
Value semantics

- **value semantics**: When primitive variables (int, double) are passed as parameters, their values are copied.
  - Modifying the parameter will not affect the variable passed in.

```java
public static void strange(int x) {
    x = x + 1;
    System.out.println("1. x = " + x);
}

public static void main(String[] args) {
    int x = 23;
    strange(x);
    System.out.println("2. x = " + x);
    ...
}
```

Output:

1. x = 24
2. x = 23
public class ParameterMystery {
    public static void main(String[] args) {
        int x = 9;
        int y = 2;
        int z = 5;

        mystery(z, y, x);
        mystery(y, x, z);
    }

    public static void mystery(int x, int z, int y) {
        System.out.println(z + " " + (y - x));
    }
}

A. 5 -7  B. 9 -3  C. 2 4  D. 9 -3  E. None of
5 -7  5 7  9 3  5 12  A - D
What is output by the following code?

```java
int x = 2;
int y = 5;
mystery2(x, y);
System.out.println(x + " " + y);

public static void mystery2(int x, int y) {
    System.out.println(x + " " + y);
    x *= y + 3;
    y--;
    x++;
    System.out.println(x + " " + y);
}
```

A. 2 5  
B. 2 5  
C. 17 4  
D. 2 5  
E. None of

17 4  
17 4  
2 5  
2 5  
17 4  
17 4  
17 4
Recall: Strings

- **string**: A sequence of text characters.

  ```java
  String <name> = "<text>";
  String <name> = <expression resulting in String>;
  ```

- Examples:

  ```java
  String name = "Marla Singer";
  int x = 3;
  int y = 5;
  String point = "(" + x + ", " + y + ")";
  ```
public class StringParameters {
    public static void main(String[] args) {
        sayHello("Marty");
        String teacher = "Bictolia";
        sayHello(teacher);
    }

    public static void sayHello(String name) {
        System.out.println("Welcome, " + name);
    }
}

Output:
Welcome, Marty
Welcome, Bictolia