

# Static Methods and Method Calls

# Algorithms

- **Algorithm:** A list of steps for solving a problem.
- **Example Algorithm:** `bakeSugarCookies()`
  - Mix the dry ingredients.
  - Cream the butter and sugar.
  - Beat in the eggs.
  - Stir in the dry ingredients.
  - Set the oven temperature.
  - Set the timer.
  - Place the cookies into the oven.
  - Allow the cookies to bake.
  - Spread the frosting and sprinkles onto the cookies. , etc.



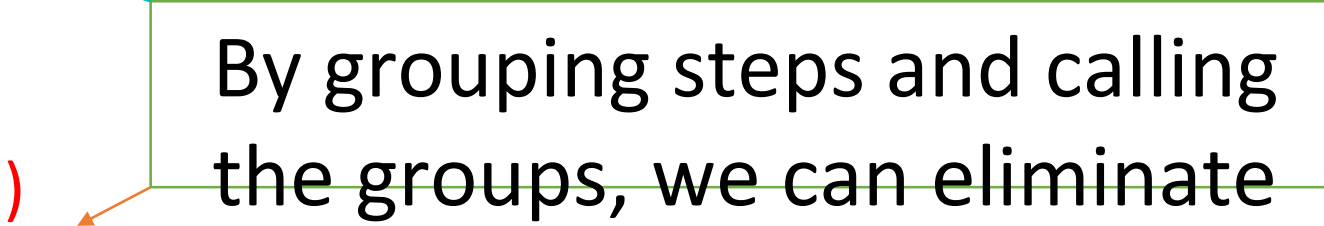
# Problems with Algorithms

- *Lack of structure:* Many tiny steps; tough to remember each step
- *Redundancy:* Consider making a double batch
  - ...
  - Stir in the dry ingredients
  - Set the oven temperature
  - Set the timer
  - Place the first batch of cookies into the oven
  - Allow the cookies to bake
  - Set the oven temperature
  - Set the timer
  - Place the second batch of cookies into the oven
  - Allow the cookies to bake
  - ....

# Removing Redundancy

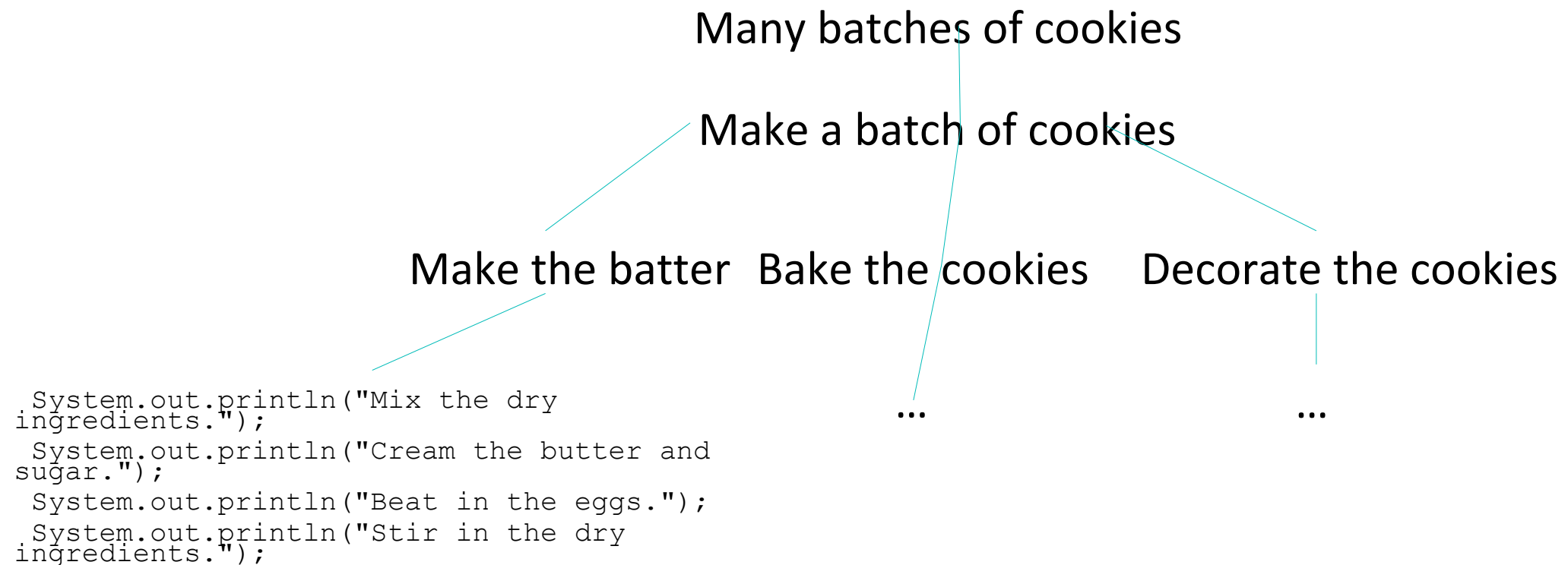
A well-structured algorithm can describe repeated tasks with less redundancy

1. Make the cookie batter.
  1. Mix in the dry ingredients.
  2. ...
2. Bake the cookies (first batch)
  1. Set the oven temperature.
  2. Set the timer.
  3. ...
- 2a. Bake the cookies (second batch)
3. Decorate the cookies.
  3. ...



By grouping steps and calling the groups, we can eliminate redundancy.

# Structure Diagram



Allows you to divide and conquer

# Static Methods

**Static method:** a named group of statements

**Procedural decomposition:** dividing a problem into methods

Writing a static method is like adding a new command to Java

# Using Static Methods

**Define / Declare** the method

**Call (or run)** the method

**\*Insider Tip\*** The `main` method always runs first

# Defining and Declaring a Method

*Giving your method a name so it can be executed:*

Syntax:

```
public static void name() {  
    statement;  
    statement;  
    ...  
    statement;  
}
```

Example:

```
public static void makeBatter(){  
    System.out.println("Mix the dry ingredients.");  
    System.out.println("Cream the butter/sugar.");  
    System.out.println("Beat in the eggs.");  
    System.out.println("Stir in dry ingredients.")  
}
```



# Calling Static Methods

*Executes the method's code*

**Syntax :** <name> ()

**Example:** `makeBatter ()`

**Output:**

```
Mix the dry ingredients.  
Cream the butter/sugar  
Beat in the eggs  
Stir in dry ingredients
```

**This whole block of code is called every time [ `makeBatter ()` ] is called.**

```
// This program displays a delicious recipe for baking cookies.
public static void main(String[] args)
{
    // Step 1: Make the cake batter.
    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.");

    // Step 2a: Bake cookies (first batch).
    System.out.println("Set the oven temperature.");
    System.out.println("Set the timer.");
    System.out.println("Place a batch of cookies into the oven.");
    System.out.println("Allow the cookies to bake.");

    // Step 2b: Bake cookies (second batch).
    System.out.println("Set the oven temperature.");
    System.out.println("Set the timer.");
    System.out.println("Place a batch of cookies into the oven.");
    System.out.println("Allow the cookies to bake.");

    // Step 3: Decorate the cookies.
    System.out.println("Mix ingredients for frosting.");
    System.out.println("Spread frosting and sprinkles.");
}
```

**// This program displays a delicious recipe for baking cookies.**

```
public class BakeCookies3
{
    public static void main(String[] args)
    {
        makeBatter();
        bake();           // 1st batch
        bake();           // 2nd batch
        decorate();
    }

    // Step 1: Make the cake batter.
    public static void makeBatter()
    {
        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.");
    }

    // Step 2: Bake a batch of cookies.
    public static void bake()
    {
        System.out.println("Set the oven temperature.");
        System.out.println("Set the timer.");
        System.out.println("Place batch into oven.");
        System.out.println("Allow the cookies to bake.");
    }

    // Step 3: Decorate the cookies.
    public static void decorate()
    {
        System.out.println("Mix ingredients for frosting.");
        System.out.println("Spread frosting and sprinkles.");
    }
}
```

This affords you a lot  
of new capabilities.

# Practice-It

Complete the following practice-it questions:

- Tricky
- Strange
- Confusing
- Lots-of-Errors

# Homework

- Read chapter 1 section 5 (1.5) (p. 40-46)
- Do chapter 1 exercises 11, 12, 14, and 16. (p. 56-58)