## CIS 110: Introduction to Computer Programming

Lecture 2
Decomposition and Static Methods
(§ 1.4)

#### Outline

- · Structure and redundancy in algorithms
- · Static methods
- · Procedural decomposition

0/45/204

011 CIS 110 (11fa) - University of Pennsylva

#### Reminders for the week

- Intro CIS 110/final exam surveys
- Sign up for and use Piazza!
- · Lecture and lab content
- My office hours (GRW 260)
  - MW: one hr immediately after each lecture
  - Th: 5:30 6:30
  - Or email me for an appointment!

9/16/2011

IS 110 (11fa) - University of Pennsylvani

#### In Review: Why are We Here?

- Learning about algorithmic thinking via computer programming!
  - 1. Precision
  - 2. Decomposition
  - 3. Abstraction

Our focus for most of the semester!

1/16/2011 CIS 110 (11fa) - University of Pennsylvania

### Structure and Redundancy in Algorithms

9/16/2011

CIS 110 (11fa) - University of Pennsylvania

#### **Our Running Example**

# public class PBJ { public static void main(String[] args) { System.out.println("Take out the ingredients and utensils."); System.out.println("Put ingredients and utensils on the table."); System.out.println("Put ingredients and utensils on the table."); System.out.println("Scop out some peanut butter."); System.out.println("Scop out some peanut butter."); System.out.println("Spread it on a piece of bread."); System.out.println("Mash knife in the sink."); System.out.println("Mash knife in the sink."); System.out.println("Coop out some jelly."); System.out.println("Spread it on a piece of bread."); System.out.println("Wash knife in the sink."); System.out.println("Wash knife in the sink."); System.out.println("Put the two pieces of bread together."); System.out.println("Put the two pieces of bread together."); System.out.println("Put the bread into your mouth and chew."); } } 9/8/2011 CS 110 (11fg) - University of Pennyhania

```
Problem #2: We're Repeating
Ourselves (Poorly)

public class PBJ {
    public static void main(String[] args) {
        System.out.println("Take out the ingredients and utensils.");
        System.out.println("Take out the ingredients and utensils on the table.");
        System.out.println("Remove the cap from the peanut butter.");
        System.out.println("Scoop out some peanut butter.");
        System.out.println("Scoop out some peanut butter.");
        System.out.println("Wash kinfe in the sink.");
        System.out.println("Wash kinfe in the sink.");
        System.out.println("Remove the cap from the jelly.");
        System.out.println("Remove the cap from the jelly.");
        System.out.println("Scoop out some jelly.");
        System.out.println("Spread it on a piece of bread.");
        System.out.println("Wash knife in the sink.");
        System.out.println("Wash knife in the sink.");
        System.out.println("Put the vopicess of bread together.");
        System.out.println("Put the bread into your mouth and chew.");
    }
}

9/16/2011 CS 110 (116) - University of Persmyhannia 2
```

#### Structure and Redundancy

- Programs should reflect the structure of the problem at hand
  - ➤ Better understanding
- Programs should not contain redundancy
   Better maintainability

What tool can we use to solve these problems?

9/16/201

CIS 110 (11fa) - University of Pennsylvania

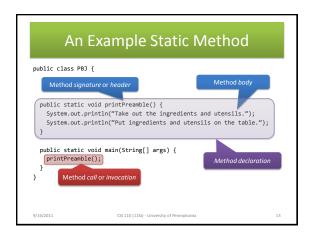
# Static Methods 9/16/2011 CS 110 (116) - University of Pennsylvania 10

### First: a Note About Learning Programming Languages

- 1. There are lots of layers to a language
- 2. As you peel away the layers, you'll cry
- 3. But it's not the layer that causes you to cry, it's the *cutting itself* that does it!
  - It gets better the sharper your programming language knife becomes!



#### 



#### Static Method Calls

· You use methods by invoking or calling them.

```
[Inside some method]
...
<name of method>();
```

- · Calling a method results in
  - 1. Executing the body of that method.
  - 2. Resuming execution right after you made the call.
- System.out.println(...) is just another method call!

9/16/2011 CIS 110 (11fa) - University of Pennsylvania

#### Where Can We Call Methods?

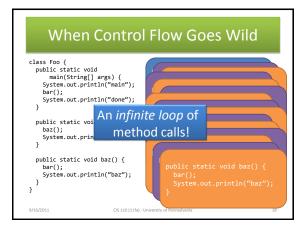
From main...
 public static void main(String[] args) {
 foo();
 }

But main is just another method, so really...
public static void myMethod() {
 someOtherMethod();
}

• We can call any method from any other method!

/16/2011 CIS 110 (11fa) - University of Pennsylvania

```
Control flow
class Foo {
 public static void
    main(String[] args) {
System.out.println("main");
    System.out.println("done");
  public static void bar() {
    baz();
    System.out.println("bar");
                                      Output:
 public static void baz() {
                                      main
    System.out.println("baz")
                                      baz
                                      bar
                                      done
                        CIS 110 (11fa) - Un
```



#### **Runtime and Logic Errors**

- · Remember: compilation is only step #2!
- · Your program may compile, but it might (probably!) still has errors to fix:
  - Runtime errors, e.g., infinite method call chains.
  - Logic errors, e.g., incorrect output.
- Lesson: compilation isn't the end! Always test and check your programs before you're done!

CIS 110 (11fa) - University of Pennsylvania

#### Using Static Methods to Capture Structure // Prints instructions to make a PBJ sandwich public class PBJ { // Prints the PBJ preamble to the screen public static void printPreamble() { /\* .. \*/ } // Prints the jelly step to the screen public static void printJellyStep() { /\* = \*/ } // Prints the eating step to the screen public static void printEatStep() { /\* .. \*/ } public static void main(String[] args) { printPreamble(); printPeanutButterStep(); printJellyStep(); printEatStep();

CIS 110 (11fa) - University of Pennsylvania

#### Using Static Methods to Reduce Redundancy // Prints instructions to make a PBJ sandwich // Prints the clean up step to the scree public static void printCleanupStep() { // Prints the peanut butter step to the screen public static void printPeanutButterStep() { /\* \_ \*/ printCleanupStep(); // Prints the jelly step to the screen public static void printJellyStep() { printCleanupStep(); public static void main(String[] args) { /\* ... \*/ }

## **Procedural Decomposition**

#### Decompose, Decompose

- Our focus thus far: procedural decomposition
  - "Procedures" are (non-object oriented) methods
- Two development strategies arise:
- Top-down development
  - Start with empty main, write skeletons for methods you believe you need, fill them in.
- Iterative refinement
  - Write a (relatively) complete program in main, factor out existing functionality into methods.
- Both approaches focus on
  - Keeping your program in a compliable state.
  - Constantly checking and testing your program.
- Not mutually exclusive, neither one better than the other.

CIS 110 (11fa) - University of Pennsylvania

#### Example #1: MarathonTraining

- Week 1
  - Monday: Rest
- Tuesday: 4 miles
- Wednesday: Rest
- Thursday: 1-hour run
- Friday: Rest
- Saturday: 4 miles
- Sunday: 6 miles
- Week 2
  - Monday: Rest
  - Tuesday: 4 miles
  - Wednesday: Rest
- Thursday: 1-hour run

- - Friday: Rest
  - Saturday: 4 miles
  - Sunday: 7 miles
- Week 3
  - Monday: Rest
  - Tuesday: 4 miles
  - Wednesday: Rest
    - Thursday: 6 miles
  - Friday: Rest
  - Saturday: Rest
  - Sunday: 8 miles

