

CS 152

Computer Programming Fundamentals Methods

Brooke Chenoweth

University of New Mexico

Fall 2026

Methods

- Instructions chunked together and given a name.
- Also may be referred to as subroutine, procedure, function
- A subroutine inside a class is a *method*
- In Java, all subroutines must be inside a class, so they are all methods.

Compact Java File vs Regular Class

- Most of our code so far has been inside *compact Java files*
 - Less boilerplate, good for beginners
 - Entry point for a program
 - Hard to use this code from another file
- For future projects, we'll start explicitly defining classes to hold our variables and methods.
 - Class name must match filename.
 - Class is *public*
 - All methods and variables are inside the class.

Method Definition

```
modifiers return-type methodName( paramList ) {  
    statements for method body  
}
```

```
public static void printNum(int n) {  
    IO.println(n);  
}
```

```
public static int addNums(int n, int m) {  
    return n+m;  
}
```

Method modifiers

- Access modifiers (public, private)
 - Methods used by other code should be public
 - Helper methods only used within the class may be private
- Static or not (At this point, we are only using static methods)
 - A static method belongs to the class and doesn't need an object to use it.
 - We'll write non-static methods when we make our own data types later.

Parameters

- Allows information to be passed into a method.
- Each parameter has a name and a type.
- May have empty parameter list.
- When calling a method, the values passed as *arguments* are assigned to the parameters.

Return type

- A method may return a value (or not)
- May only return value of the specified *return type*
- Method that does not return a value has return type void
- Use a return statement to return a value

```
return value;
```

Methods can call other methods

```
public static void main(String[] args) {  
  
    prettyPrintNum(42);  
  
    int foo = 37;  
    prettyPrintNum(foo);  
  
    prettyPrintNum(3 + foo);  
  
    int result = addAndPrint(5,4);  
    IO.println("the result was " + result);  
  
    IO.println("the next result was " + addAndPrint(2,3));  
}  
  
public static int addAndPrint(int a, int b) {  
    int sum = a + b;  
    prettyPrintNum(sum);  
    return sum;  
}  
  
public static void prettyPrintNum(int num) {  
    IO.println("My number is " + num);  
}
```