
Introduction to Methods

CS0007

Introduction

- We defined one special method so far
- We made use of already implemented methods in Java API



Introduction

- We defined one special method so far
- We made use of already implemented methods in Java API



Introduction

- We defined one special method so far

```
public static void main(String[] args){  
    statement;  
    statement;  
}
```

- We made use of already implemented methods in Java API

```
Math.min(int x,int y) Math.max(int x, int y)
```

Introduction

- Basic idea - a named group of statements
 - It provides a well-defined functionality
 - After defining a method, you can think of it as a black box, which may or may not take input arguments and performs the desired function
 - A method can also return a value to the caller
 - An input argument is the information passed from outside to the method, which effects the behaviour, the output of the method or both
-

Outline

- How to define new methods other than main
 - How to call a method and get the return value
 - Why do we want to use methods
-

Defining methods

- A method definition consists of two general parts a header and a body
- The method body is a collection of statements that are performed when the method is executed (It is enclosed inside a set of curly braces)
- The method header consists of method modifiers, return type, method name, and parameter variables (in parantheses)

see example 1

Calling Methods

- The arguments provided in the method call are copied into the parameter variables, which behave like initialized local variables in the method
- Execution is transferred to the beginning of the called method
- When the method returns, the caller continues from the next statement after the method call
- If the method returned a value, that value is passed back to the caller

see example 1

Important issues

- Arguments are passed by value
- An argument you pass to the method should have the same type as the corresponding parameter variable (or should be castable)
- While passing multiple arguments to a method, the order of the arguments is important

see example 1

Advantages of using methods

- Reusable code
 - If you need to do the same thing, or almost the same thing, many times, write a method to do it, then call the method each time you have to do that task.
- Divide and Conquer
 - Solving a big problem by breaking it down into little problems and by writing a method for each little problem. It helps to create conceptual units.

see example 2

Advantages of using methods

■ To parameterize code

- In addition to making reusable code that is the same in all cases, you will often want to use parameters that change the way the method works.

■ References

- Java Programming Notes
 - (<http://www.csd.abdn.ac.uk/~etadjoud/teaching/CS5536/information/notes-java/index.html>)
 - Deitel: Java How to program
 - Course book (Starting out with Java 5)
-