



User Interaction

Deborah A. Trytten

CS 1323/4

Reminders (Aug 30)

- Multiple assignments due next week.
 - Labor Day (Monday):
 - zyBook chapters 1-4
 - TC 1 and TC 2
 - Project 1
 - Sept. 5 (Thursday):
 - zyBook chapter 5
 - Homework 1
 - Sept. 6 (Friday):
 - TC 3
 - iClicker questions

Review

- Last time:
 - Arithmetic between an int and double → promote int to double
 - Casting:
 - (double) converts int to double by adding .0
 - (int) converts double to int by truncating decimal part
 - Order of operations: (1) parentheses, (2) cast, (3) *, /, %, (4) +, -
For a given order, evaluate left to right.
 - Example: `int x = 5;`
`double y = 7.5;`
`double z = (int) (x + y) + (double) x * (int) y;`

After the code executes, what is stored in x, y, and z?

Preview

- Today and Next Class: Input and Output
 - Introducing methods
 - Examples of methods in the Math class
 - Outputting to the console with System.out
 - Reading from the keyboard with a Scanner

Information Hiding

- The practice of concealing details from the user
- Example: Driving a car
- Makes it possible to program a computer without knowing all the details

Input and Output (I/O)

- How programs communicate with users
- Relies on a multitude of details that are hidden from the user
 - Data stored on disk as 0's and 1's
 - Data read from keyboard as characters
 - Characters are converted to int/double/String
 - Details are concealed from programmer using methods supplied by Java

Calling Methods

- Methods are how Java conceals complex operations
- Example: `Math.max(3, 5)` produces 5
- **Arguments** are the data the method needs to do its job
 - What are the arguments above?
- Methods can return a value
- Returned value stored in a variable

```
int max = Math.max(3,5);
```

Using Variables for Arguments

- Variables can be arguments too
- The value of the variable is used by the method
- Example:

```
int first = 3;
```

```
int second = 5;
```

```
int largest = Math.max(first, second);
```


iClicker Question

- What is stored in `bigDogWeight` when the following statements are run?

```
int bigDogWeight = 0;
```

```
int jazzPounds = 47;
```

```
int ravenPounds = 75;
```

```
int bigDogWeight = Math.max(jazzPounds, ravenPounds);
```

a) 47

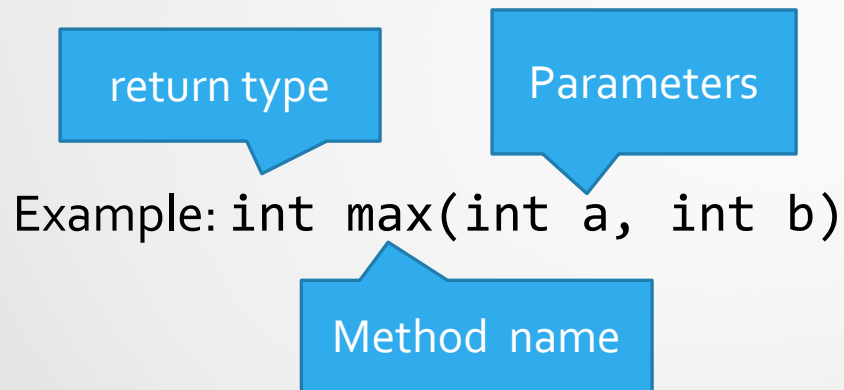
b) 75

c) 0

d) Something else

Signature

- The combination of the method's name, parameters and return type is called the signature



- **Parameters** are placeholders for the **arguments** that the programmer will supply when the method is used
 - They have to have a type to tell the programmer what is legal
 - Examine Math class in Java API

Arguments

- The value of the arguments is given to the parameters when the method is called
- Arguments do not have a type because Java already knows the type

```
Math.max(3, 5);
```

```
Math.max(3.6, 7.4);
```

- Java can promote int arguments to double
 - double to int does not make sense. Why?

iClicker Question

Suppose there is a method with the following signature:

```
int Math.round(double input)
```

If the method is called with the line

```
int roundedValue = Math.round(37.2);
```

which of the following statements is true?

- a) input is an argument
- b) 37.2 is a parameter
- c) roundedValue is a return type
- d) round is a method name
- e) None of the above

Announcements and Reminders (Sept 4)

- Upcoming assignments
 - Sept. 5 (Thursday):
 - zyBook chapter 5
 - Homework 1
 - Sept. 6 (Friday):
 - TC₃
 - Midterm 1 is a week from next Monday
- Turing's Craft issues
 - Site crashes on some computers when incorrect answer is given
 - I've contacted TC support about this

Review and Preview

- Last time:
 - Introducing methods
 - Examples of methods in the Math class
- Today:
 - Outputting to the console with System.out
 - Had to do this in Project 1 and Project 2
 - Reading from the keyboard with a Scanner
 - Unable to do this in Project 2, but we will in Project 3!

One More Example

- Go to the Math class in the API and find a method that finds the absolute value of an integer
 - Math: $|3| = 3$, $|-3| = 3$
- Use this method to find the absolute value of -5, saving the result in the variable below
 - `int positive;`

Your Turn: Think, Pair, Share

- A signature for a method in the Math class is:
`double pow(double a, double b)`
- `pow` calculates a^b
- Write code that calculates 7^3 using `pow` and stores the result in an integer variable named `power`
 - Carefully note the data types in the method signature

iClicker Question

- A signature for a method in the Math class is
`double sqrt(double number)`
- Which of the following is illegal in Java:
 - Answer a: `Math.sqrt(36.0);`
 - Answer b: `Math.sqrt(36);`
 - Answer c: `Math.sqrt(double 36.0); // not a cast`
 - Answer d: `Math.sqrt((double) 36.0); // cast`
 - Answer e: All of those are legal

What's before the dot?

- Java has to know where to find the instructions that are concealed by the method
- Two possibilities:
 - Class name
 - Example: Math
 - Object name
 - Example: System.out
- Memorize for now

Input and Output

- System.in
 - Standard input
 - In Eclipse this comes from the keyboard via the Console
- System.out
 - Standard output
 - In eclipse this goes to Console
 - You used this in Project 1 and 2

Output

- `int count = 7;`
- `System.out.println("I am a number: " + count);`
 - Output: I am a number: 7<newline>
 - + is String concatenation
 - Same precedence as addition of numerical data
 - Puts a newline character at end ('\n')
- Primitive data is automatically converted to a String
- `System.out.print("I am a number: " + count);`
 - Output: I am a number: 7
 - Similar, but no newline at end

iClicker Question

- What will the following code output?

```
int a = 3;
```

```
int b = 9;
```

```
System.out.println("A+B is " + a + b);
```

Answer a: A+B is 12

Answer b: A+B is12

Answer c: A+B is 3 9

Answer d: A+B is 39

Construct a Scanner

- Scanner is the class we use for input
 - We need an object from this class
- Find package to import from Java API
- Only use one Scanner in each program
 - In main program

Scanner Object

- `import java.util.Scanner;`
 - Import at top of source file (before class declaration)
- `Scanner keyboard;`
 - Declare a variable to store the Scanner
- `keyboard = new Scanner(System.in);`
 - Connect the variable above to the keyboard via the Console
- `keyboard.close();`
 - Close Scanner at end of main
 - Not actually required, but it is nice

Scanner Methods

- `nextLine(): String`
 - Return characters read from keyboard until next newline entered
- `next(): String`
 - Return characters read from keyboard until new whitespace entered (space, tab, newline)
- `nextInt(): int`
 - Return the next int read from keyboard
- `nextDouble(): double`
 - Return the next double read from the keyboard
- These methods need no arguments. Why?

Announcements and Reminders (Sept 6)

- Upcoming assignments
 - TC 3 postponed until Sept. 8 (Sunday)
 - Homework 2 is posted
 - Please download and input answers with Adobe Reader, rather than using your web browser
 - Homework 1 solution will be posted soon
 - Midterm 1 is a week from next Monday
- Welcome party!
 - 2-4 PM in Devon atrium
 - Free food and a free T-shirt!

Write Code: Scanner Example

- Write code to read the following input from the keyboard into variables that store the name, age, and weight
- Example input:
Raven 11 79.8
- Should we use `next`, `nextInt`, `nextDouble`, or `nextLine` to read each part of the input?

iClicker Question

Suppose you want to read input of the form

`firstName lastName age`

into the variables

`String name; int age;`

Example input: Fred Muggs 18

Example output: name is assigned "Fred Muggs", age is assigned 18

Assuming the Scanner keyboard has been constructed, which two lines of code accomplish this?

Answer a: `name = keyboard.nextLine();`

`age = keyboard.nextInt();`

Answer b: `name = keyboard.next() + " " + keyboard.next();`

`age = keyboard.nextInt();`

Answer c: `name = keyboard.next();`

`age = keyboard.nextInt();`

Answer d: `name = keyboard.next() + keyboard.next();`

`age = keyboard.nextInt();`

Example Program

- Write a program that reads in two integers from the user, performs the remainder operation (%) with them, and reports the result to the user
 - You could use this program to practice %!
 - Create this on your own, or follow along with me
 - My code will go to Canvas at the end of class
1. Create a project
 2. Create a class
 3. Create a main program
 4. Put commands in main program

Exceptions: Java's Temper Tantrum

- If the Scanner class is given something unanticipated, it throws an Exception
 - Show example
- Many different types of exceptions
 - InputMismatchException
 - Data couldn't be converted to desired type
 - IllegalStateException
 - Scanner already closed
- Perfect user assumption: We assume that our users are perfect at all times
- Show exceptions being thrown using program