

hi

## Building Java Programs

Chapter 4  
Lecture 4-1: Scanner;  
returns revisited;  
Cumulative algorithms

Copyright 2010 by Pearson Education

## Interactive Programs with Scanner

**reading: 3.3 – 3.4**

Copyright 2010 by Pearson Education

### Interactive programs

**interactive program:** Reads input from the console.

- While the program runs, it asks the user to type input.
- The input typed by the user is stored in variables in the code.
- Can be tricky; users are unpredictable and misbehave.
- But interactive programs have more interesting behavior.

Copyright 2010 by Pearson Education 3

### Scanner

- **Scanner:** An object that can read input from many sources.
  - Communicates with `System.in`
  - Can also read from files (Ch. 6), web sites, databases, ...
- The `Scanner` class is found in the `java.util` package.
 

```
import java.util.*; // so you can use Scanner
```
- Constructing a `Scanner` object to read console input:
 

```
Scanner name = new Scanner(System.in);
```
- Example:
 

```
Scanner console = new Scanner(System.in);
```

Copyright 2010 by Pearson Education 4

### Scanner methods

Method	Description
<code>nextInt()</code>	reads an <code>int</code> from the user and returns it
<code>nextDouble()</code>	reads a <code>double</code> from the user
<code>next()</code>	reads a one-word <code>String</code> from the user
<code>nextLine()</code>	reads a one-line <code>String</code> from the user

- Each method waits until the user presses Enter.
- The value typed by the user is returned.

```
System.out.print("How old are you? "); // prompt
int age = console.nextInt();
System.out.println("You typed " + age);
```

- **prompt:** A message telling the user what input to type.

Copyright 2010 by Pearson Education 5

### Scanner example

```
import java.util.*; // so that I can use Scanner
public class UserInputExample {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.print("How old are you? ");
        int age = console.nextInt();
        int years = 65 - age;
        System.out.println(years + " years until retirement!");
    }
}
```

How old are you? 29  
36 years until retirement!

- Console (user input underlined):

Copyright 2010 by Pearson Education 6

bye

1

hi

### Scanner example 2

```
import java.util.*; // so that I can use Scanner
public class ScannerMultiply {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.print("Please type two numbers: ");
        int num1 = console.nextInt();
        int num2 = console.nextInt();
        int product = num1 * num2;
        System.out.println("The product is " + product);
    }
}
```

- Output (user input underlined):  
Please type two numbers: 8 6  
The product is 48
- The Scanner can read multiple values from one line.

Copyright 2010 by Pearson Education 7

### Input tokens

- token:** A unit of user input, as read by the Scanner.
  - Tokens are separated by *whitespace* (spaces, tabs, new lines).
  - How many tokens appear on the following line of input?  
23 John Smith 42.0 "Hello world" \$2.50 " 19"
- When a token is not the type you ask for, it crashes.
 

```
System.out.print("What is your age? ");
int age = console.nextInt();
```

Output:  
What is your age? Timmy  
java.util.InputMismatchException  
at java.util.Scanner.next(Unknown Source)  
at java.util.Scanner.nextInt(Unknown Source)  
...

Copyright 2010 by Pearson Education 8

### Scanner BMI question

A person's body mass index (BMI) is defined to be:

$$BMI = \frac{weight}{height^2} \times 703$$

Write a program that produces the following output:

This program reads in data for two people and computes their body mass index (BMI) and weight status.

```
Enter next person's information:
height (in inches)? 62.5
weight (in pounds)? 130.5

Enter next person's information:
height (in inches)? 58.5
weight (in pounds)? 90

Person #1 body mass index = 23.485824
Person #2 body mass index = 18.487836949375414
Difference = 4.997987050624587
```

Copyright 2010 by Pearson Education 9

### Scanners as parameters

- If many methods need to read input, declare a Scanner in main and pass it to the other methods as a parameter.

```
public static void main(String[] args) {
    Scanner console = new Scanner(System.in);
    int sum = readSum3(console);
    System.out.println("The sum is " + sum);
}

// Prompts for 3 numbers and returns their sum.
public static int readSum3(Scanner console) {
    System.out.print("Type 3 numbers: ");
    int num1 = console.nextInt();
    int num2 = console.nextInt();
    int num3 = console.nextInt();
    return num1 + num2 + num3;
}
```

Copyright 2010 by Pearson Education 10

### Return

- return:** To send out a value as the result of a method.
  - The opposite of a parameter:
    - Parameters send information *in* from the caller to the method.
    - Return values send information *out* from a method to its caller.
    - A call to the method can be used as part of an expression.

Copyright 2010 by Pearson Education 11

### Returning a value

```
public static type name(parameters) {
    statements;
    ...
    return expression;
}
```

- Example:
 

```
// Returns the slope of the line between the given points.
public static double slope(int x1, int y1, int x2, int y2) {
    double dy = y2 - y1;
    double dx = x2 - x1;
    return dy / dx;
}
```

  - slope(1, 3, 5, 11) returns 2.0

Copyright 2010 by Pearson Education 12

bye

hi

### Common error: Not storing

- Many students incorrectly think that a return statement sends a variable's name back to the calling method.

```
public static void main(String[] args) {
    slope(0, 0, 6, 3);
    System.out.println("The slope is " + result); // ERROR:
                                                // result not defined
}

public static double slope(int x1, int x2, int y1, int y2) {
    double dy = y2 - y1;
    double dx = x2 - x1;
    double result = dy / dx;
    return result;
}
```

Copyright 2010 by Pearson Education 13

### Fixing the common error

- Instead, returning sends the variable's value back.
  - The returned value must be stored into a variable or used in an expression to be useful to the caller.

```
public static void main(String[] args) {
    double s = slope(0, 0, 6, 3);
    System.out.println("The slope is " + s);
}

public static double slope(int x1, int x2, int y1, int y2) {
    double dy = y2 - y1;
    double dx = x2 - x1;
    double result = dy / dx;
    return result;
}
```

Copyright 2010 by Pearson Education 14

### Scanner BMI solution

```
// This program computes two people's body mass index (BMI)
// and compares them. The code uses parameters and returns.

import java.util.*; // so that I can use Scanner

public class BMI {
    public static void main(String[] args) {
        introduction();
        Scanner console = new Scanner(System.in);

        double bmi1 = processPerson(console);
        double bmi2 = processPerson(console);

        // report overall results
        System.out.println("Person #1 body mass index = " + bmi1);
        System.out.println("Person #2 body mass index = " + bmi2);
        double difference = Math.abs(bmi1 - bmi2);
        System.out.println("Difference = " + difference);
    }

    // prints a welcome message explaining the program
    public static void introduction() {
        System.out.println("This program reads in data for two people");
        System.out.println("And computes their body mass index (BMI)");
        System.out.println("And weight status.");
        System.out.println();
    }
}
```

Copyright 2010 by Pearson Education 15

### Scanner BMI solution, cont.

```
...

// reads information for one person, computes their BMI, and returns it
public static double processPerson(Scanner console) {
    System.out.println("Enter next person's information:");
    System.out.print("height (in inches)? ");
    double height = console.nextDouble();

    System.out.print("weight (in pounds)? ");
    double weight = console.nextDouble();
    System.out.println();

    double bmi = getBMI(height, weight);
    return bmi;
}

// Computes a person's body mass index based on their height and weight
// and returns the BMI as its result.
public static double getBMI(double height, double weight) {
    double bmi = weight / (height * height) * 703;
    return bmi;
}
```

Copyright 2010 by Pearson Education 16

### Cumulative algorithms

reading: 4.2

Copyright 2010 by Pearson Education

### Adding many numbers

- How would you find the sum of all integers from 1-1000?

```
int sum = 1 + 2 + 3 + 4 + ... ;
System.out.println("The sum is " + sum);
```
- What if we want the sum from 1 - 1,000,000? Or the sum up to any maximum?
- We could write a method that accepts the max value as a parameter and prints the sum.
  - How can we generalize code like the above?

Copyright 2010 by Pearson Education 18

bye

3

hi

### A failed attempt

- An incorrect solution for summing 1-1000:

```
for (int i = 1; i <= 1000; i++) {
    int sum = 0;
    sum = sum + i;
}
// sum is undefined here
System.out.println("The sum is " + sum);
```
- sum's scope is in the for loop, so the code does not compile.
- cumulative sum:** A variable that keeps a sum in progress and is updated repeatedly until summing is finished.
  - The sum in the above code is an attempt at a cumulative sum.

Copyright 2010 by Pearson Education 19

### Fixed cumulative sum loop

- A corrected version of the sum loop code:

```
int sum = 0;
for (int i = 1; i <= 1000; i++) {
    sum = sum + i;
}
System.out.println("The sum is " + sum);
```
- Key idea:**
  - Cumulative sum variables must be declared *outside* the loops that update them, so that they will exist after the loop.

Copyright 2010 by Pearson Education 20

### Scanner and cumulative sum

- We can do a cumulative sum of user input:

```
Scanner console = new Scanner(System.in);
int sum = 0;
for (int i = 1; i <= 100; i++) {
    System.out.print("Type a number: ");
    sum = sum + console.nextInt();
}
System.out.println("The sum is " + sum);
```

Copyright 2010 by Pearson Education 21

### Cumulative Sum Exercise

- Write a program that computes the average of any given number of doubles.
- Example output:

```
How many numbers would you like to average? 4
Enter number 1: 3
Enter number 2: 2
Enter number 3: 6.6
Enter number 4: 77
The average is: 22.15
```

Copyright 2010 by Pearson Education 22

### Types int and double

- Printing double values can be ugly:

```
double result = 1.0 / 3.0;
System.out.println(result); // 0.3333333333333333
```
- Can we print it with only 2 digits after the decimal?
- Rounding the number doesn't help:

```
double result = 1.0 / 3.0;
System.out.println(Math.round(result)); // 0
```

Copyright 2010 by Pearson Education 23

### Rounding real numbers

- To round to  $N$  places:
  - multiply by  $10^N$
  - round
  - divide by  $10^N$
- Example:

```
double result = 1.0 / 3.0; // 0.3333333333333333
result = result * 100; // 33.3333333333
result = Math.round(result); // 33.0
result = result / 100; // 0.33
System.out.println(result);
```

Copyright 2010 by Pearson Education 24

bye

4