Building Java Programs

Chapter 4
Lecture 4-1: Scanner; if/else

reading: 3.3 – 3.4, 4.1

Interactive Programs with Scanner

reading: 3.3 - 3.4
Interactive programs

- We have written programs that print console output.

- It is also possible to read *input* from the console.
  - The user types the input into the console.
  - The program uses the input to do something.
  - Such a program is called an *interactive program*.

Interactive programs

- Interactive programs can be challenging.
  - Computers and users think in very different ways.
  - Users tend to “misbehave”.

![Comic strip](image)

*We could design the product with a simple point-and-click interface... Or we could require the user to choose among thousands of poorly documented commands, each of which must be typed exactly right on the first try. Bear in mind, we'll never meet a customer ourselves. Make it so they have to reboot after every typo.*
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, etc...

• The **Scanner** class is found in the java.util package.

```java
import java.util.*;  // so you can use Scanner
```

• **Constructing a Scanner object to read console input:**

```java
Scanner <name> = new Scanner(System.in);
```

• Example:

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

Scanner **methods**

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

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

```java
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.
Scanner example

```java
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!");
    }
}
```

• Console (user input underlined):
How old are you? 12
53 years until retirement!

Scanner example 2

```java
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.
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.
  
  ```java
  System.out.print("What is your age? ");
  int age = console.nextInt();
  
  Output:
  What is your age? Timmy
  java.util.InputMismatchException
  at java.util.Scanner.nextInt(Unknown Source)
  at java.util.Scanner.nextInt(Unknown Source)
  ...
  ```

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.

  ```java
  public static void main(String[] args) {
      Scanner console = new Scanner(System.in);
      int sum = readSum3(console);
      double average = sum / 3.0;
      System.out.println("The average is ", average);
  }
  
  // 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;
  }
  ```
Program puzzle

• Consider changing the output to include the minimum value:

Type 3 numbers: 8 6 13
The average is 9.0
The minimum value is 6

• How would we change the previous program?

```java
public static void main(String[] args) {
    Scanner console = new Scanner(System.in);
    int sum = readSum3(console);
    double average = sum / 3.0
    System.out.println("The average is "+ average);
    // What goes here?
}

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;
}
```

Can’t return multiple values!

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

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

        System.out.print("Type 3 numbers: ");
        int num1 = console.nextInt();
        int num2 = console.nextInt();
        int num3 = console.nextInt();

        double average = (num1 + num2 + num3) / 3.0;
        System.out.println("The average is "+ average);
        System.out.println("The minimum value is "+
                        Math.min(num1, Math.min(num2, num3)));
    }
}
```
The if/else statement

reading: 4.1

Conditionals

- “If you eat your vegetables, then you can have dessert.”
- “If you do your homework, then you may go outside to play, or else you’ll be grounded for life.”
The if statement

Executes a block of statements only if a test is true

```java
if (<test>) {
    <statement>;
    ...
    <statement>;
}
```

- Example:
  ```java
double gpa = console.nextDouble();
if (gpa >= 3.0) {
    System.out.println("Good job! Here's a cookie.");
}
```

The if/else statement

Executes one block if a test is true, another if false

```java
if (<test>) {
    <statement(s)>;
} else {
    <statement(s)>;
}
```

- Example:
  ```java
double gpa = console.nextDouble();
if (gpa >= 3.0) {
    System.out.println("Good job! Here's a cookie.");
} else {
    System.out.println("No cookie for you!");
}
```
Relational expressions

- *if* statements and *for* loops both use logical tests.

```java
for (int i = 1; i <= 10; i++) { ... 
    if (i <= 10) { ... 
```

- These are Boolean expressions, seen in Ch. 5.

- Tests use *relational operators*:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equals</td>
<td>1 + 1 == 2</td>
<td>true</td>
</tr>
<tr>
<td>!=</td>
<td>does not equal</td>
<td>3.2 != 2.5</td>
<td>true</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>10 &lt; 5</td>
<td>false</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>10 &gt; 5</td>
<td>true</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
<td>126 &lt;= 100</td>
<td>false</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
<td>5.0 &gt;= 5.0</td>
<td>true</td>
</tr>
</tbody>
</table>

Misuse of *if*

- What's wrong with the following code?

```java
Scanner console = new Scanner(System.in);
System.out.print("What percentage did you earn? ");
int percent = console.nextInt();
if (percent >= 90) {
    System.out.println("You got an A!");
}
if (percent >= 80) {
    System.out.println("You got a B!");
}
if (percent >= 70) {
    System.out.println("You got a C!");
}
if (percent >= 60) {
    System.out.println("You got a D!");
}
if (percent < 60) {
    System.out.println("You got an F!");
}
...```

```java
diagram
```
Nested if/else

Chooses between outcomes using many tests

```java
if (<test>) {
    <statement(s>);
} else if (<test>) {
    <statement(s>);
} else {
    <statement(s>);
}
```

- Example:
  ```java
  if (x > 0) {
      System.out.println("Positive");
  } else if (x < 0) {
      System.out.println("Negative");
  } else {
      System.out.println("Zero");
  }
  ```

Nested if/else/if

- If it ends with else, exactly one path must be taken.
- If it ends with if, the code might not execute any path.

```java
if (<test>) {
    <statement(s>);
} else if (<test>) {
    <statement(s>);
} else if (<test>) {
    <statement(s>);
}
```

- Example:
  ```java
  if (place == 1) {
      System.out.println("Gold medal!");
  } else if (place == 2) {
      System.out.println("Silver medal!");
  } else if (place == 3) {
      System.out.println("Bronze medal.");
  }
  ```
Summary: if structures

- exactly 1 path (mutually exclusive)
  
  ```
  if (<test>) {
    <statement(s>);
  } else if (<test>) {
    <statement(s>);
  } else {
    <statement(s>);
  }
  ```

- 0 or 1 path (mutually exclusive)
  
  ```
  if (<test>) {
    <statement(s>);
  } else if (<test>) {
    <statement(s>);
  } else if (<test>) {
    <statement(s>);
  }
  ```

- 0, 1, or many paths (independent tests; not exclusive)
  
  ```
  if (<test>) {
    <statement(s>);
  }
  if (<test>) {
    <statement(s>);
  }
  if (<test>) {
    <statement(s>);
  }
  ```

Which nested if/else?

- (1) if/if/if
- (2) nested if/else
- (3) nested if/else/if

- Whether a user is lower, middle, or upper-class based on income.
  - (2) nested if / else if / else

- Whether you made the dean's list (GPA ≥ 3.8) or honor roll (3.5-3.8).
  - (3) nested if / else if

- Whether a number is divisible by 2, 3, and/or 5.
  - (1) sequential if / if / if

- Computing a grade of A, B, C, D, or F based on a percentage.
  - (2) nested if / else if / else if / else if / else
Nested if/else question

Formula for body mass index (BMI):

\[
BMI = \frac{\text{weight}}{\text{height}^2} \times 703
\]

<table>
<thead>
<tr>
<th>BMI</th>
<th>Weight class</th>
</tr>
</thead>
<tbody>
<tr>
<td>below 18.5</td>
<td>underweight</td>
</tr>
<tr>
<td>18.5 - 24.9</td>
<td>normal</td>
</tr>
<tr>
<td>25.0 - 29.9</td>
<td>overweight</td>
</tr>
<tr>
<td>30.0 and up</td>
<td>obese</td>
</tr>
</tbody>
</table>

- Write a program that produces output like the following:

  This program reads data for two people and computes their body mass index (BMI).
  Enter next person's information:
  height (in inches)? 70.0
  weight (in pounds)? 194.25
  Enter next person's information:
  height (in inches)? 62.5
  weight (in pounds)? 130.5
  Person 1 BMI = 27.868928571428572
  overweight
  Person 2 BMI = 23.485824
  normal
  Difference = 4.3831045714285715

// This program computes two people's body mass index (BMI) and compares them. The code uses Scanner for input, and parameters/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 = person(console);
        double bmi2 = person(console);
        // report overall results
        report(1, bmi1);
        report(2, bmi2);
        System.out.println("Difference = " + Math.abs(bmi1 - bmi2));
    }
    // prints a welcome message explaining the program
    public static void introduction() {
        System.out.println("This program reads data for two people and");
        System.out.println("computes their body mass index (BMI). ");
    }
    // function to calculate BMI
    public static double person(Scanner console) {
        System.out.print("Enter height (in inches)? ");
        double height = console.nextDouble();
        System.out.print("Enter weight (in pounds)? ");
        double weight = console.nextDouble();
        return weight / (height * height) * 703;
    }
    // function to report BMI
    public static void report(int person, double bmi) {
        if (bmi < 18.5) {
            System.out.println(person + " BMI = " + bmi + " underweight");
        } else if (bmi >= 18.5 && bmi <= 24.9) {
            System.out.println(person + " BMI = " + bmi + " normal");
        } else if (bmi >= 25.0 && bmi <= 29.9) {
            System.out.println(person + " BMI = " + bmi + " overweight");
        } else {
            System.out.println(person + " BMI = " + bmi + " obese");
        }
    }
    ...
Nested if/else, cont'd.

// reads information for one person, computes their BMI, and returns it
public static double person(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 bodyMass = bmi(height, weight);
    return bodyMass;
}

// Computes/returns a person's BMI based on their height and weight.
public static double bmi(double height, double weight) {
    return (weight * 703 / height / height);
}

// Outputs information about a person's BMI and weight status.
public static void report(int number, double bmi) {
    System.out.println("Person " + number + " BMI = " + bmi);
    if (bmi < 18.5) {
        System.out.println("underweight");
    } else if (bmi < 25) {
        System.out.println("normal");
    } else if (bmi < 30) {
        System.out.println("overweight");
    } else {
        System.out.println("obese");
    }
}

The if/else hammer

• Just because you learned a new construct does not mean that every new problem has to be solved using that construct!

    int z;
    if (x > y) {
        z = x;
    } else {
        z = y;
    }

double d = a;
    if (b < d) {
        d = b;
    } else if (c < d) {
        d = c;
    }

double d = Math.min(a, Math.min(b, c));