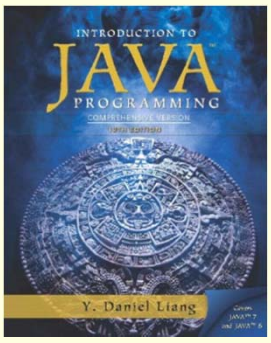
CS 259
Computer Programming Fundamentals
The `if-else` *Statement*

Instructor:
Joel Castellanos
e-mail: joel@unm.edu
Web: http://cs.unm.edu/~joel/
Office: Electrical and Computer Engineering building (ECE).
Room 233

9/9/2016

Textbook & Reading Assignment

Introduction to java Programming (10th Edition) by Y. Daniel Liang



Read by Wednesday: Aug 31
- Chapter 3: Selections
  - **Use of multi-way if-else statements

Read by Friday: Sept 2
- Chapter 4: Mathematical Functions, Characters, and Strings

Read by Monday: Sept 5
- Chapter 5: Loops

Read by Wednesday: Sept 14
- Chapter 6: Methods

2

a)

```
public class Tmp
{
  public static void main(String[] args)
  {
    System.out.println("Pick Me");
  }
}
```

b)

```
public class Tmp
  {
  public static void main(String[] args)
    {
  System.out.println("No, Me");
    }
  }
```

c)

```
public class Tmp
  {
  public static void main(String[] args)
  {
    System.out.println("Ooh, Ooh");
    }
  }
```

Quiz:

Which is the Correct Formatting in *CS-259*?

3

Java Primitive Type: **boolean**

```
public class Toy_3_2
{
  public static void main(String[] args)
  {
    boolean a = true;
    boolean b = (5*(-5)) > 0;

    System.out.println(a);
    System.out.println(b);
  }
}
```

Output:

```
true
false
```

4

Logical Expressions

```
1.  public class Hello
2.  { public static void main(String[] args)
3.    {
4.      int a = 1;   int b = 3;
5.      System.out.println( a+b +  14 );
6.      System.out.println( a+b <  14 );
7.      System.out.println( a+b >  14 );
8.      System.out.println( a+b <= a*b );
9.      System.out.println( a+b >  a*b );
10.   }
11. }
```

Output:
```
18
true
false
false
true
```

5

Logical Expressions

```
1.  public class Hello
2.  { public static void main(String[] args)
3.    {
4.      int a = 3;
5.      int b = 5;
6.      int c = 7;
7.      System.out.println( c > a && c > b );
8.      System.out.println( b > a && b > c );
9.      System.out.println( b > c && b > a );
10.     System.out.println( b > a || b > c );
11.   }
12. }
```

Output:
```
true
false
false
true
```

6

Using a boolean Variable to hold Results

```
1) public class Hello
2) { public static void main(String[] args)
3)   {
4)     int a = -3;
5)     int b =  9;
6)     int c =  5;
7)     System.out.println((a+b>c) && (a*b>c));
8)
9)     boolean r1 = a+b > c;  //true
10)    boolean r2 = a*b > c;  //false
11)    System.out.println( r1 && r2 );
12)    System.out.println( r1 || r2 );
13)  }
14)}
```

Output:
```
false
false
true
```

7

Quiz: Logical Expressions

What is the output of the **`println`** statement?

```
int a = 3;
int b = 5;
int c = 7;
System.out.println( (b < a && b < c)
  + " " +  (b < a || b < c)
  + " " +  (b > c || a > c) );
```

a) false true false
b) false false true
c) true false false
d) false true true
e) true true false

8

Quiz: Logical Expressions

What is the output of the **println** statement?

```
int a = 3;
int b = 5;
int c = 11;
System.out.println(
   (a+b > c && a*b > c)  + " " +
   (a+b > c || a*b > c)  + " " +
   (a+b < c || a*b < c));
```

a) false true false
b) false false true
c) true false false
d) false true true
e) true true false

9

The **if** Statement

The **if** statement tells your program to execute a certain section of code *only if* a particular test evaluates to **true**.

```
if (grade >= 90) System.out.println("Cash Award");
```

Logical expression

One statement or one block { }. Executed only if logical expression is **true**

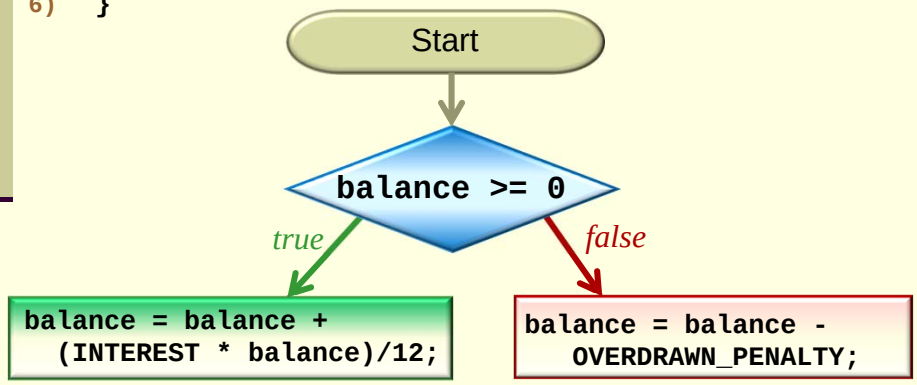
When the "then" part is more than one statement, brackets are needed.

```
if (grade >= 90)
{ System.out.println("Cash Award");
  cash = cash + 100.0;
}
```

10

`if-else` Statement Flow Chart

```
1)  if (balance >= 0)
2)  { balance = balance + (INTEREST * balance)/12;
3)  }
4)  else
5)  { balance = balance - OVERDRAWN_PENALTY;
6)  }
```



11

Quiz: Identifiers in All Caps

```
public class BankBalance
{
  public static final double OVERDRAWN_PENALTY = 8.00;
  public static final double INTEREST_RATE = 0.02;

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

In the code segment from listing 3.1 shown above, **`OVERDRAWN_PENALTY`** and **`INTEREST_RATE`** in are in all caps:

a) because they are outside **`main`**, and, therefore the code would not actually compile.

b) because they are declared **`static`**.

c) because they are declared **`final`**.

d) because they are global class variables.

e) because later they are used in different branches of an **`if`**.

12

`if` and `else` Statements

```
1) public class Toy_3_1
2) {
3)   public static void main(String[] args)
4)   {
5)     int x = 5;
6)
7)
8)     if (x == (2+3)) x = x + 10;
9)     else x = x * 2;
10)
11)
12)    System.out.println(x);
13)  }
14)}
```

Logical "*is equal to*" Operator

Assignment Operator

Output: 15

13

`if` and `else` Statements

Either statement 2 ***or*** statement 5 will execute.
It is impossible for ***both*** to execute.

```
1) if (testscore >= 90)
2) { System.out.println("Great Job!");
3) }
4) else
5) { System.out.println("Work Harder");
6) }
```

Since the “then” part is only one statement, curly brackets are not needed.

However, when the code does not fit on a single line, the CS-259 coding standard dictates that the curly brackets be used.

14

if, else if, else

```
1) public class HelloWorld
2) { public static void main(String[] args)
3)   {
4)     int x = 1;
5)
6)     if (x == 1)          ¡¡¡No semicolon!!!
7)     { System.out.println("x is 1");
8)     }
9)     else if (x == 2)     ¡¡¡No semicolon!!!
10)    { System.out.println("x is 2");
11)    }
12)    else                 ¡¡¡No semicolon!!!
13)    { System.out.println("x is special");
14)    }
15)  }
```

15

Use of else if

```
1) public class HelloWorld
2) { public static void main(String[] args)
3)   { int testscore = 76;
4)     char grade = 'F';
5)
6)     if (testscore >= 90) grade = 'A';
7)     else if (testscore >= 80) grade = 'B';
8)     else if (testscore >= 70) grade = 'C';
9)     else if (testscore >= 60) grade = 'D';
10)
11)    System.out.println("Grade = " + grade);
12)  }
13)}
```

With **testscore = 76**, the logical expressions in lines 8 and 9 would both evaluate to **true**.

However, since 8 is **true**, execution never reaches 9.

16

Find the Syntax Error

```
1) public class HelloWorld
2) { public static void main(String[] args)
3)   { int testscore = 76;
4)     char grade;
5)
6)     if (testscore >= 90) grade = 'A';
7)     else if (testscore >= 80) grade = 'B';
8)     else if (testscore >= 70) grade = 'C';
9)     else if (testscore >= 60) grade = 'D';
10)
11)    System.out.println("Grade = " + grade);
12)  }
13)}
```

Local variable grade may not have been initialized.

17

`else`: Ensuring `grade` is Initialized

```
1) public class HelloWorld
2) { public static void main(String[] args)
3)   { int testscore = 76;
4)     char grade;
5)
6)     if (testscore >= 90) grade = 'A';
7)     else if (testscore >= 80) grade = 'B';
8)     else if (testscore >= 70) grade = 'C';
9)     else if (testscore >= 60) grade = 'D';
10)    else grade = 'F';
11)    System.out.println("Grade = " + grade);
12)  }
13)}
```

The compiler recognizes there is no path to line 11 in which grade is not initialized.

18

Quiz: `if` and `else if`

```
1) int testscore = 88;
2) char grade = 'F';
3)
4) if (testscore >= 60) grade = 'D';
5) else if (testscore >= 70) grade = 'C';
6) else if (testscore >= 80) grade = 'B';
7) else if (testscore >= 90) grade = 'A';
8)
9) System.out.println(grade);
```

What would be the output of the above Java code?

a) B b) D c) F d) FB e) FDCB

19

Control Flow and Program State

```
1) public class HelloWorld
2) { public static void main(String[] args)
3)   { int x = 5;
4)     int a = 0;
5)     if (x < 10) a=1;
6)     if (x < 6)  a=2;
7)     if (x < 1)  a=3;
8)     System.out.println(a);
9)   }
10)}
```

start of line	x	a
line 3		
line 4	5	
line 5	5	0
line 6	5	1
line 7	5	2
line 8	5	2

Table of ***program state*** at the ***start*** of each line in the order of execution.

Output:
2

20

Control Flow and Program State

```
1) public static void main(String[] args)
2) { int x = 5;
3)   int a = 0;
4)   if (x < 10)
5)   { a=1;
6)   }
7)   if (x < 6)
8)   { a=2;
9)   }
10)  if (x < 1)
11)  { a=3;
12)  }
13)  System.out.println(a);
14)}
```

start of line	x	a
line 2		
line 3	5	
line 4	5	0
line 5	5	0
line 7	5	1
line 8	5	1
line 10	5	2
line 13	5	2

Output:
2

21

Control Flow: `if` & `else if`

```
1) public static void main(String[] args)
2) { int x = 5;
3)   int a = 0;
4)   if (x < 10)
5)   { a=1;
6)   }
7)   else if (x < 6)
8)   { a=2;
9)   }
10)  else if (x < 1)
11)  { a=3;
12)  }
13)  System.out.println(a);
14)}
```

start of line	x	a
2)		
3)	5	
4)	5	0
5)	5	0
13)	5	1

Output:
1

22

Quiz: `if`, `else if`, `else`

```
1)  public static void main(String[] args)
2)  {
3)    int x = 1;
4)
5)    if (x == 1)
6)    { System.out.println("x is 1");
7)    }
8)    else if (x == 2)
9)    { System.out.println("x is 2");
10)   }
11)   else x = 3;
12)   { System.out.println("wild: x=" + x);
13)   }
14) }
```

Look carefully
This code does not do what it was probably intended to do.

The output is:
a) x is 1
wild: x=1
b) x is 1
c) x is 2
d) wild: x=1
e) wild: x=3

23

`if`, `else if`, `else`

```
1)  public static void main(String[] args)
2)  {
3)    int x = 4;
4)
5)    if (x == 1)
6)    { System.out.println("x is 1");
7)    }
8)    else if (x == 2)
9)    { System.out.println("x is 2");
10)   }
11)   else x = 3;
12)   { System.out.println("x is " + x);
13)   }
14) }
```

Only Change

Output: `x is 3`

24

Example: `if`, `else if`, `else`

```
1)   int x=3, y=7;
2)   if (x*x < x+y)
3)   { System.out.print("B");
4)   }
5)   if (x > 0)
6)   { System.out.print("E");
7)   }
8)   else if (y > 0)
9)   { System.out.print("A");
10)  }
11)  else
12)  { System.out.print("T");
13)  }
14)  if (x*y > 0)
15)  { System.out.print("S");
16)  }
```

Output:

BES

25

Quiz: `if & else if`

```
1)  public static void main(String[] args)
2)  {
3)    int x = 50;
4)    if (x > 20)
5)    { System.out.print("A");
6)    }
7)    else if (x > 30)
8)    { System.out.print("B");
9)    }
10)   else if (x > 40)
11)   { System.out.print("C");
12)   }
13)   System.out.println("D");
14) }
```

The output is:
a) ABCD b) ABD c) CD d) C e) AD

26

Quiz: `if`, `else if`

```
1)  int a = 3, b = 4, c = 5;
2)
3)  if (a+b > c) System.out.print("A");
4)
5)  else if (a*b > c) System.out.print("B");
6)
7)  if (a*a > c) System.out.print("C");
8)
9)  else if (b*b > c) System.out.print("D");
10)
11) else if (b*b >= c*a) System.out.print("E");
12)
13) System.out.println("F");
```

The output of this Java code segment is:
a) AF b) ACF c) ABCF d) ACDF e) ACDEF

27

Logical Operators

`==`	Equals
`!=`	Not Equal
`<`	Less than
`>`	Grater than
`<=`	Less than or Equal to
`>=`	Greater than or Equal to
`||`	Logical OR
`&&`	Logical AND
`!`	Logical NOT

The bitwise operators are not covered in CS-259.

```
 |  bitwise OR
 &  bitwise AND
 ^  bitwise exclusive OR
 ~  bitwise NOT
```

28

Truth Tables for AND, OR and NOT

```
1) public class TruthTables
2) {
3)   public static void main(String[] args)
4)   {
5)     System.out.println( true && true );
6)     System.out.println( true && false );
7)     System.out.println( false && true);
8)     System.out.println( false && false );
9)
10)    System.out.println( true || true );
11)    System.out.println( true || false );
12)    System.out.println( false || true );
13)    System.out.println( false || false );
14)
15)    System.out.println(!true);
16)    System.out.println(!false);
17)  }
18)}
```

```
true
false
false
false
```

```
true
true
true
false
```

```
false
true
```

29

The Logical AND Operator: &&

true when ***both parts*** are true

```
1) if ((pressure >= min) && (pressure <= max))
2) { System.out.println("Pressure is OK");
3) }
4) else
5) { System.out.println(
6)     "Warning: Pressure is out of range.");
7) }
```

The order of operations, in ***Java***, makes these equivalent:

```
if ((pressure >= min) && (pressure <= max))
if ( pressure >= min  &&  pressure <= max )
```

30

The Logical AND Operator: ||

true when ***both parts*** are true
true when ***either part*** is true

```
1) if ((pressure < min ) || (pressure > max ))
2) { System.out.println(
3)    "Warning: Pressure is out of range.");
4) }
5) else
6) { System.out.println("Pressure is OK");
7) }
```

31

What Happens on Line 4?

```
1)  public class TruthTable
2)  { public static void main(String[] args)
3)    {
4)      System.out.println(xor(true,  true));
5)      System.out.println(xor(true,  false));
6)      System.out.println(xor(false, true));
7)      System.out.println(xor(false, false));
8)    }
9)
10)   public static boolean xor(boolean a, boolean b)
11)   {
12)     return ( a || b) && !(a && b);
13)   }
14) }
```

false

```
( T || T) && !(T && T)
(    T  ) && !(  T   )
     T    && F
          F
```

Truth Table for Exclusive OR

```
T T = F
T F = T
F T = T
F F = F
```

32

Trace of What Happens on Line 5

```
1)  public class TruthTable
2)  { public static void main(String[] args)
3)    {
4)      System.out.println(xor(true,  true));
5)      System.out.println(xor(true,  false));
6)      System.out.println(xor(false, true));
7)      System.out.println(xor(false, false));
8)    }
9)
10)   public static boolean xor(boolean a, boolean b)
11)   {
12)     return ( a || b) && !(a && b);
13)   }
14) }
```

```
true
```

```
( T || F) && !(T && F)
(    T  ) && !(  F   )
     T    && T
          T
```

Truth Table for Exclusive OR

```
T T = F
T F = T
F T = T
F F = F
```

33

Which of the `if` statements have identical truth tables?

```
1) public class TruthTable
2) { public static void main(String[] args)
3)   {
4)      foo(true,true);
5)      foo(true,false);
6)      foo(false,true);
7)      foo(false,false);
8)   }
9)
10)  public static void foo(boolean a, boolean b)
11)  {
12)    if (   a ||  b) System.out.print("CAT ");
13)    if (  !a && !b) System.out.print("ANT ");
14)    if (!( a &&  b)) System.out.print("BAT ");
15)    if (!(!a && !b)) System.out.print("DOG ");
16)    System.out.println();
17)  }
18)}
```

```
CAT DOG
CAT BAT DOG
CAT BAT DOG
ANT BAT
```

34

“Short circuit” Evaluation

What is meant by “this expression is safe”.

```
if (x != 0.0 && 1.0/x > x + y)
```

exp1 exp2

The expression: **`(exp1 && exp2)`**

can only be **`true`** if ***both*** **`exp1`** ***and*** **`exp2`** are **`true`**.

Thus, if **`exp1`** is **`false`**, Java does not evaluate **`exp2`**.

Thus, if **`x`** = **`0`**, then **`1.0/x`** is not evaluated.

35

Order of Evaluation

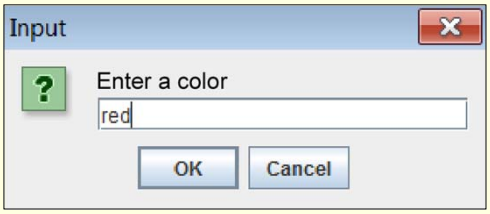
```
double x =  2.0;
double y = -2.0;
if (x != 0.0 && 1.0/x > x + y)
```

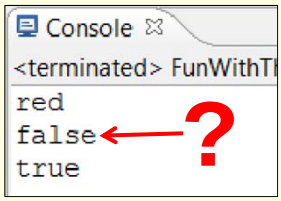
1) `x != 0.0`
2) `1.0/x`
3) `x + y`
4) `0.5 > 0.0`
5) `true && true`
6) `true`

36

Comparing Strings

```
1)  String myColor = "red"
2)  System.out.println(myColor);
3)  System.out.println(myColor == "red");
4)  System.out.println(myColor.equals("red"));
```





37

Comparing Addresses verses Strings

```
String myColor;
//Some code that puts data in myColor.
```

`myColor == "red"`

`true` if and only if the ***memory location*** of **`myColor`** is the same as the ***memory location*** of "**`red`**"

`myColor.equals("red")`

`true` if and only if the ***data*** in **`myColor`** is "**`red`**" .

38

Possible Null Pointer Exception

```
1)  String str = JOptionPane.showInputDialog(null,
2)    "Enter red");
3)  if (str.equals(""))
4)  { System.out.println("Clicked OK with no data");
5)  }
6)  if (str == null)
7)  { System.out.println("Clicked Cancel");
8)  }
```

A Null Pointer Exception occurs when code attempts to access a ***member of*** an object using an object reference that is not pointing anywhere.

In line 3, **`equals`** is a method that is a ***member of*** a String object.

However, if **`str`** is **`null`**, then, even though str was ***declared as*** being a ***reference to*** a String object, it doesn't, in fact, ***point to*** anything.

39

Does This Fix the Problem?

```
1)  String str = JOptionPane.showInputDialog(null,
2)    "Enter red");
3)  if (str == null)
4)  { System.out.println("Clicked Cancel");
5)  }
6)  if (str.equals(""))
7)  { System.out.println("Clicked OK with no data");
8)  }
```

40

Checking Input: All Golden

```
1)  String str = JOptionPane.showInputDialog(
2)     null, "Enter red");
3)  if (str == null)
4)  { System.out.println("Clicked Cancel");
5)  }
6)  else if (str.equals(""))
7)  { System.out.println("OK with no data");
8)  }
```

```
1)  String str = JOptionPane.showInputDialog(
2)     null, "Enter red");
3)  if (str == null)
4)  { System.out.println("Clicked Cancel");
5)    System.exit(0);
6)  }
7)  if (str.equals(""))
8)  { System.out.println("OK with no data");
9)  }
```

41

How can this be coded in Java?

- Let X be a course with 3 grades: a midterm, a final exam and a final project. Each counts as 1/3 of the course grade.
  Example: t_1=90, t_2=70, p=80 → *grade* = 80.0
- However, if a student does better on the final than on the midterm, then the final is counted with twice the weight as the midterm.
  Example: t_1=0, t_2=90, p=90 → *grade* = 70.0
- A student who gets a 95% or more on the final project can drop his or her final exam score and count the final project as 2/3 and the midterm as 1/3 of the course grade.
  Example: t_1=90, t_2=0, p=90 → *grade* = 90.0

42

Example Calculation: Count Final 2x Midterm

Case 1: $grade = (t_1 + t_2 + p)/3$

Paper & Pencil ***before*** code!

Given:

$grade = (w)t_1 + (2w)t_2 + (1/3)p$

$w + 2w + 1/3 = 1$

Case 2

Counting t_2 with twice the weight as t_1

$w + 2w = 2/3$

$3w = 2/3$

$w = 2/9$

$grade = (2/9)t_1 + (4/9)t_2 + (1/3)p$

Java Code for Conditional Grade

```
1)  Scanner in = new Scanner(System.in);
2)  double t1 = in.nextDouble();
3)  double t2 = in.nextDouble();
4)  double p  = in.nextDouble();
5)
6)  double grade = (t1 + t2 + p) / 3.0;
7)
8)  if (t2 > t1)
9)  { grade = t1*(2.0/9.0) + t2*(4.0/9.0) + p/3.0;
10) }
11)
12) if (p >= 95)
13) { double tmp = (t1/3.0) + (p*2.0/3.0);
14)   if ( tmp > grade) grade = tmp;
15) }
16) System.out.println("Grade="+grade);
```

Could or Should the order of the two `if` statements be switched?

Could or Should this be changed to `else if`?