COMP-248
Object Oriented Programming I

Week 3: Control Flow 3

By Emad Shihab, PhD, Fall 2015,
Parts of the slides are taken from Prof. L. Kosseim
Adapted for Section EE by S. Ghaderpanah, Fall 2015

Announcements

In this chapter, we will see:

1. The **`if`** statement
2. The **`if-else`** statement
3. Relations Operators
4. Logical operators
5. **Compound statements**
6. Nested **`if`** statements
7. The **`switch`** statement
8. The conditional operator
9. The **`while`** loop
10. The **`do-while`** loop
11. The **`for`** loop
12. Nested loops
13. break, continue & exit

5- Compound statements

```
if ( condition )
    statement;
```

```
if ( condition )
    statement1;
else
    statement2;
```

what if you wanted to execute several statements?

Several statements can be grouped together into a ***compound statement*** (or block):

```
{
    statement1;
    statement2;
    …
}
```

A **block** can be used wherever a statement is called for by the Java syntax

Example

```
int grade;

System.out.print("what is your grade?");
grade = myKeyboard.nextInt();

if (grade >= 80)
   System.out.println("congratulations!");
else
   System.out.println("you could do better");
   System.out.println("make sure you practice");
System.out.println("bye bye");
```

```
89?
79?
```

Output

In this chapter, we will see:

1. The **`if`** statement
2. The **`if-else`** statement
3. Relations Operators
4. Logical operators
5. Compound statements
6. **Nested `if` statements**
7. The **`switch`** statement
8. The conditional operator
9. The **`while`** loop
10. The **`do-while`** loop
11. The **`for`** loop
12. Nested loops
13. break, continue & exit

6- Nested if statements

an **if** is a statement… so we can put an **if** "inside" and **if** called ***nested if statements***

```
int num1, num2, num3, min = 0;
System.out.println ("Enter three integers:
   ");
num1 = keyboard.nextInt();
num2 = keyboard.nextInt();
num3 = keyboard.nextInt();
if (num1 < num2)
    if (num1 < num3)
       min = num1;
    else
       min = num3;
else
    if (num2 < num3)
       min = num2;
    else
       min = num3;
System.out.println ("Minimum value: " +
   min);
```

```
Enter three integers:
70
98
122
Minimum value: 70
```

Output

Just checking …

Given the following code segment, what is stored in **a** at the end of this sequence if **a** is initialized to 0?

```
if (a >= 10)
if (a < 20)
a = a + 2;
else
a = a + 1;
```

A. 0

B. 1

C. 2

D. 3

E. Syntax error

Dangling else

An `else` clause is matched to the last unmatched `if`
(no matter what the indentation implies)

```
if (condition1)
   if (condition2)
      statement1;
else
      statement2;
```

```
if (condition1)
{
   if (condition2)
      statement1;
}
else
   statement2;
```

Example: Leap year

Problem:

leap years occur in years exactly divisible by four, except the years ending in 00 are leap years only if they are divisible by 400.

Examples

1700, 1800, 1900, 2100, and 2200 are not leap years

1600, 2000, and 2400 are leap years.

Algorithm:

```
if year is a multiple of 400 --> leap
otherwise
    if year is a multiple of 100 --> not leap
    otherwise
        if year is a multiple of 4 --> leap
        otherwise --> not leap
```

Code

```
if (year % 400 == 0)
{
        System.out.println("Leap");
}
else
{
        if (year % 100 == 0)
        {
                System.out.println("Not leap");
        }
        else
        {
                if (year % 4 ==0)
                {
                        System.out.println("Leap");
                }
                else
                {
                        System.out.println("Not leap");
                }
        }
}
```

if year is a multiple of 400 --> leap
otherwise
    if year is a multiple of 100 --> not leap
    otherwise
        if year is a multiple of 4 --> leap
        otherwise --> not leap

In this chapter, we will see:

1. The **`if`** statement
2. The **`if-else`** statement
3. Relations Operators
4. Logical operators
5. Compound statements
6. Nested **`if`** statements
7. **The `switch` statement**
8. The conditional operator
9. The **`while`** loop
10. The **`do-while`** loop
11. The **`for`** loop
12. Nested loops
13. break, continue & exit

7- The switch statement

Remember: Java's conditional statements are
- the statement
- the statement
- the **`switch`** statement

The **`switch`**:
- replaces a series of **`if-else-if-else-if-else…`**
- like a multiple-choice question
- that tests the **equality** of an expression
- the expression must evaluate to a **`char`**, **`int`**, **`short`**, **`byte`** or **`String`**
- In case of **`String`** the .equals is tested (not ==)

The switch statement

syntax:

switch, case, break, default are reserved words

break and **default** case are optional

```
switch ( expression )
{
    case value1 :
        statement-list1
        break;
    case value2 :
    statement-list2
        break;
    case value3 :
    statement-list3
        break;
    case  ...
    default:
      default-statement-list
}
```

If ***expression*** matches ***value2***, control jumps to here

More on the switch

`break`

- Often used as the last statement in each case
- **`break`** causes control to transfer to the end of the **`switch`**
- If a **`break`** is not used, the flow of control will continue into the next case

`default` *case*

- A **`switch`** can have an optional **`default`** *case*
- If the **`default`** case is present, control will transfer to it **if no other case value matches**
- the default case can be positioned **anywhere** in the switch, but usually it is placed at the end

Logic of the switch

1. the expression is evaluated
2. its value is compared to the various cases
3. if an equality is found, the corresponding statements are executed until a **break** or until the end of the switch
4. if no equality is found, the default statements are executed if a default case is there.

Example 1: SwitchDemo.java

```
System.out.println("Enter number of ice cream flavors:");
int numberOfFlavors = keyboard.nextInt();
switch (numberOfFlavors)
{
    case 32:
       System.out.println("Nice selection."); // break;
    case 1:
       System.out.println("I bet it's vanilla.");
       break;
    case 2:
    case 3:
    case 4:
       System.out.println(numberOfFlavors + " flavors");
       System.out.println("is acceptable.");
       break;
    default:
       System.out.println("I didn't plan for");
       System.out.println(numberOfFlavors + " flavors.");
}
```

Example 2

```
int grade, category;
System.out.print("Enter a grade (0 to 100):");
grade = keyboard.nextInt();
category = grade / 10;
switch (category)  {
   case 10:
   case 9:
      System.out.println ("excellent.");
      break;
   case 8:
      System.out.println ("nice job.");
       break;
   case 7:
      System.out.println ("average.");
      break;
   case 6:
      System.out.println ("below average.");
       break;
   default:
      System.out.println ("problem.");
   }
}
```

If user enters 100?
excellent

If user enters 94?
excellent

If user enters 57?
problem

Output

What about…

```
int grade, category;
System.out.print("Enter a grade (0 to 100):");
grade = keyboard.nextInt();
category = grade / 10;
switch (category)
{
   case 10:
   case 9:
      System.out.println ("excellent");
   case 8:
      System.out.println ("nice job");
      break;
   case 7:
      System.out.println ("average");
      break;
   case 6:
      System.out.println ("below average");
      break;
}
```

If user enters 100?

A. excellent

B. excellent
nice job

C. excellent
nice job
average

D. excellent
nice job
average
below average

E. None of the
above choices

What about…

```
int grade, category;
System.out.print("Enter a grade (0 to
   100):");
grade = keyboard.nextInt();
category = grade / 10;
switch (category)
{
   case 10:
   case 9:
      System.out.println ("excellent");
   case 8:
      System.out.println ("nice job");
      break;
   case 7:
      System.out.println ("average");
      break;
   case 6:
      System.out.println ("below average");
      break;
}
```

If user enters **94**?

A. excellent

B. excellent
nice job

C. excellent
nice job
average

D. excellent
nice job
average
below average

E. None of the
above choices

What about…

```
int grade, category;
System.out.print("Enter a grade (0 to
   100):");
grade = keyboard.nextInt();
category = grade / 10;
switch (category)
{
   case 10:
   case 9:
      System.out.println ("excellent");
   case 8:
      System.out.println ("nice job");
      break;
   case 7:
      System.out.println ("average");
      break;
   case 6:
      System.out.println ("below average");
      break;
}
```

If user enters **57**?

A. excellent

B. excellent
   nice job

C. excellent
   nice job
   average

D. excellent
   nice job
   average
   below average

E. None of the above choices

Homework

transform the previous switch into a **`if-else`** statement

In this chapter, we will see:

1. The **`if`** statement
2. The **`if-else`** statement
3. Relations Operators
4. Logical operators
5. Compound statements
6. Nested **`if`** statements
7. The **`switch`** statement
8. **The conditional operator**
9. The **`while`** loop
10. The **`do-while`** loop
11. The **`for`** loop
12. Nested loops
13. break, continue & exit

8- The conditional operator

shortcut to an `if` in some cases

ternary operator (needs 3 operands)

syntax : ***`condition`*** `?` ***`expression1`*** `:` ***`expression2`***

semantics:

- if the ***`condition`*** is true, ***`expression1`*** is evaluated;
- if it is false, ***`expression2`*** is evaluated
- the result of the chosen expression is the result of the entire conditional operator

Example

```
larger = ((num1 > num2) ? num1 : num2);
System.out.println(larger);
```

```
If num1 is 10 and num2 is 20 ?
20

If num1 is 20 and num2 is 10 ?
20
```

Output

Example

```
System.out.println("Change is " + count +((count==1) ?
       " Dime": " Dimes"));
```

If **count** is 1?

If **count** is not 1?

Output

In this chapter, we will see:

1. The **`if`** statement
2. The **`if-else`** statement
3. Relations Operators
4. Logical operators
5. Compound statements
6. Nested **`if`** statements
7. The **`switch`** statement
8. The conditional operator
9. **The `while` loop**
10. The **`do-while`** loop
11. The **`for`** loop
12. Nested loops
13. break, continue & exit

Repetition statements (loops)

allow us to execute a statement several times

like conditional statements, they are controlled by boolean expressions

Java has 3 kinds of loops:

- the **`while`** loop
- the **`do-while`** loop
- the **`for`** loop

9- The while loop

syntax:

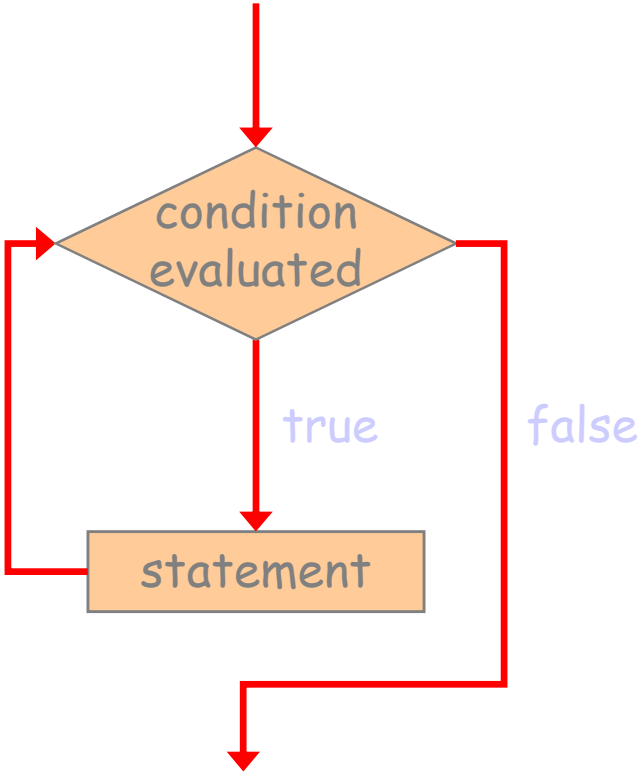
while is a reserved word

```
while ( condition )
    statement;
```

If the *condition* is true, the *statement* is executed. Then the *condition* is evaluated again.

The *statement* is executed repeatedly until the *condition* becomes false.

Logic of a while loop



The while loop

note that if the condition of a **while** statement is false initially, the statement is never executed

so, the body of a **while** loop will execute zero or more times

```
final int LIMIT = 5;
int count = 1;

while (count <= LIMIT)
{
   System.out.println(count);
   count = count + 1;
}
System.out.println("Done");
```

LIMIT	count

Trace

1
2
3
4
5
Done

Output

Example 1

```
int remainingStars = 5;

while (remainingStars > 0)
{
    System.out.println("*");
    remainingStars--;
}
```

```
int remainingStars = 5;

while (remainingStars > 0)

    System.out.println("*");
    remainingStars--;
```

remainingStars

Trace

remainingStars

Trace

```
*
*
*
*
*
```

Output

```
*
*
*
*
*
……
```

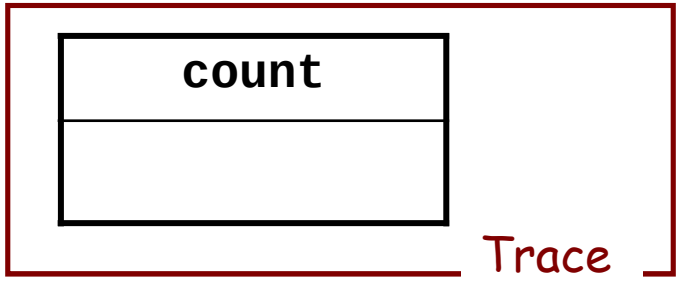
Output

Example 2

```
public class Forever
{
  public static void main(String[] args)
  {
     int count = 1;

     while (count <= 25)
     {
        System.out.println(count);
        count = count - 1;
     }

     System.out.println("Done");
  }
}
```



What will the following output?

```
index = 1;
while (index != 10)
{
    System.out.print("Hello");
    index = index + 2;

}
```

A. There will be no output, since index is not equal to 10

B. HelloHelloHelloHello

C. HelloHelloHelloHelloHello

D. HelloHelloHelloHelloHelloHello

E. None of the above are correct choices

Just checking...

The termination condition for the while loop

```
while (loopCount < 9)
{
    System.out.print(loopCount);
    loopCount++;
}
```

is loopCount > 9.

A. true

B. false

C. Syntax error

Next topic:

1. The **`if`** statement
2. The **`if-else`** statement
3. Relations Operators
4. Logical operators
5. Compound statements
6. Nested **`if`** statements
7. The **`switch`** statement
8. The conditional operator
9. The **`while`** loop
10. **The `do-while` loop**
11. The `for` loop
12. Nested loops
13. break, continue & exit