Selection Statements and operators

CSC 1051 – Data Structures and Algorithms I

Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/

Some slides in this presentation are adapted from the slides accompanying Java Software Solutions by Lewis & Loftus
Selection structures in Java

• Conditional statement:

```java
if (n>0)
    System.out.println("positive");
else
    System.out.println("negative");
```

• Other selection structures (Chapter 6 in text)

  – the `conditional` operator

  – the `switch` statement
The Conditional Operator Syntax

`condition ? expression1 : expression2`

- If the `condition` is true, `expression1` is evaluated; if it is false, `expression2` is evaluated.
- The value of the entire conditional operator is the value of the selected expression.
The Conditional Operator

• Similar to an `if-else` statement, except that it is an expression that computes a value

• For example:

```java
String result = (happy ? "happy" : "sad");
```

• The conditional operator requires three operands so it is sometimes called the `ternary` operator

Hands on: try this in the Person class!
The Conditional Operator

• Another example:

```java
page.drawArc (x+15, y+30, 20, 10, happy?180:0, 180);
```

Hands on: try this in the Smiley class!
The conditional operator is **not a statement**

```java
(x < 0) ? System.out.print("negative") : System.out.print("positive");
```
Quick Check

System.out.println("Your change is "+
count + "Dimes");

Rewrite this statement so that "Dime" is printed if
count equals 1.
Quick Check

Rewrite this code using the conditional operator.

```java
if (val <= 10)
    System.out.println("It is not greater than 10.");
else
    System.out.println("It is greater than 10.");
```
The **switch** Statement: multi-way branches

Recall: Logic of an if-else statement
The **switch** Statement: multi-way branches

**switch** statement logic

- **condition evaluated**
  - true → **statement1**
  - false → **statement2**

- **expression evaluated**
  - case: 1 → **statement1**
  - case: 2 → **statement2**
  - case: 3 → **statement3**

Note: this is a simplified flowchart of the logic of the switch statement
The switch Statement

• The *switch* *statement* provides a way to implement a multi-way choice (not just true/false)

• The *switch* *statement* evaluates an expression, then attempts to match the result to one of several possible cases

• Each case contains a value and a list of statements

• The flow of control transfers to the statement associated with the first case value that matches
The switch Statement - example

```java
public String toString()
{
    String result;
    switch (faceValue)
    {
        case 1:
            result = "one";
            break;
        case 2:
            result = "two";
            break;
        case 3:
            result = "three";
            break;
    }
    return result;
}
```

Hands on: try this in the Die class!
The switch Statement

• Another example – counting letters. Here letter is a char

```java
switch (letter) {
    case 'A':
        aCount++;
        break;
    case 'B':
        bCount++;
        break;
    case 'C':
        cCount++;
        break;
}
```
The switch Statement in general

• The general syntax of a `switch` statement is:

```java
switch (expression) {
  case value1:
    statement-list1
  case value2:
    statement-list2
  case value3:
    statement-list3
  case ...
}
```

If `expression` matches `value2`, control jumps to here.
So… the logic of the switch is more like this:

expression evaluated

1 2 3

Note: this is a still simplified flowchart of the logic of the switch statement
The switch Statement

- Optional `default` case (when no other case matches).
- The type of a switch expression must be integers, characters, or enumerated types.
- As of Java 7, a switch can also be used with strings.
- You cannot use a switch with floating point values.
- You cannot use ranges of values (e.g., 0<x<10).
- See `GradeReport.java`
import java.util.Scanner;

public class GradeReport
{
    public static void main (String[] args)
    {
        int grade, category;

        Scanner scan = new Scanner (System.in);

        System.out.print ("Enter a numeric grade (0 to 100): ");
        grade = scan.nextInt();

        category = grade / 10;

        System.out.print ("That grade is ");
        continue
continue

    switch (category)
    {
    case 10:
        System.out.println ("a perfect score. Well done.");
        break;
    case 9:
        System.out.println ("well above average. Excellent.");
        break;
    case 8:
        System.out.println ("above average. Nice job.");
        break;
    case 7:
        System.out.println ("average.");
        break;
    case 6:
        System.out.println ("below average. You should see the");
        System.out.println ("instructor to clarify the material "
          + "presented in class.");
        break;
    default:
        System.out.println ("not passing.");
    }
}
Sample Run

Enter a numeric grade (0 to 100): 91
That grade is well above average. Excellent.

System.out.println ("a perfect score. Well done.");
break;
case 9:
    System.out.println ("well above average. Excellent.");
break;
case 8:
    System.out.println ("above average. Nice job.");
break;
case 7:
    System.out.println ("average.");
break;
case 6:
    System.out.println ("below average. You should see the");
    System.out.println ("instructor to clarify the material "
        + "presented in class.");
    break;
default:
    System.out.println ("not passing.");
}
continue

switch (category)
{
    case 10:
        System.out.println ("a perfect score. Well done.");
        break;
    case 9:
        System.out.println ("well above average. Excellent.");
        break;
    case 8:
        System.out.println ("above average. Nice job.");
        break;
    case 7:
        System.out.println ("average.");
        break;
    case 6:
        System.out.println ("below average. You should see the");
        System.out.println ("instructor to clarify the material "
            + "presented in class.");
        break;
    default:
        System.out.println ("not passing.");
}
}
import java.util.Scanner;

public class SwitchExample
{
    // Example of using a switch statement. Counts number of digits, zeros, whitespace, and others in a line of input.

    public static void main (String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int digits = 0, zeros = 0, whitespace = 0, other = 0;

        System.out.print("Input line>");
        String message = scan.nextLine();

        int count = 0;

        continue
while (count < message.length()) {
    switch (message.charAt(count)) {
        case '0':  zeros++; break;
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':  digits++; break;
        case ' ': case '	': whitespace++; break;
        default:   other++;}
    // end switch
    count++;
} // end while
System.out.print(digits + " Digit" + (digits==1 ? "" : "s"));
System.out.print(" , of which " + zeros);
System.out.println((zeros==1 ? " is a zero " : " are zeros ");

System.out.println(whitespace + " whitespace");
System.out.println(other + " other" + (other==1 ? "" : "s"));
}
while (count < message.length()) {
    switch (message.charAt(count)) {
        case '0':  zeros++; break;
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':  digits++; break;
        case ' ': case '\t': whitespace++; break;
        default:   other++;
    } // end switch
    count++;
} // end while
System.out.print(digits + " Digit" + (digits==1 ? "" : "s"));
System.out.print(" , of which " + zeros);
System.out.println((zeros==1 ? " is a zero " : " are zeros "));

System.out.println(whitespace + " whitespace");
System.out.println(other + " other" + (other==1 ? "" : "s"));
}
Homework

• Review Sections 6.1 and 6.2
  • **Always** do all self-review exercises when you review

• Exercises – be sure to do all the hands-on exercises in this presentation. Try also the following:
  – Implement the Dog class, similar to the Person class. In addition to being happy or sad, the dog should also have another state symbolized by an integer: 1=sit; 2=sleep; 3=shake; 4=run. Use a switch statement in toString() method to indicate the state of the dog
  – Incorporate the conditional operator and the switch statement in your next project.