Designing Methods & Classes

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/
Last time:

Lab 7: Creating methods

• Syntax for declaring a method –
  – method header
  – method body

• Method invocation

• Parameters
Next

• More about methods
• Designing classes
import javax.swing.JApplet;
import java.awt.*;

public class Snowman extends JApplet
{

    //-----------------------------------------------------------------
    //  Draws a winter scene, including some snowmen.
    //-----------------------------------------------------------------
    public void paint (Graphics page)
    {
        << paint method-body >>
    }

    //-----------------------------------------------------------------
    //  Draws a snowman at MID, TOP.
    //-----------------------------------------------------------------
    public void paintSnowman(Graphics page, int MID, int TOP)
    {
        << paintSnowman method-body >>
    }
}
Method Header

• A method declaration begins with a *method header*

```
char calc (int num1, int num2, String message)
```

- **Method Name**: `calc`
- **Return Type**: `char`
- **Parameter List**: `int num1, int num2, String message`

The parameter list specifies the type and name of each parameter.

The name of a parameter in the method declaration is called a *formal parameter*.
Method Body

• The method header is followed by the *method body*

```java
char calc (int num1, int num2, String message) {
    int sum = num1 + num2;
    char result = message.charAt (sum);

    return result;
}
```

The return expression must be consistent with the return type

*sum and result are local data*

They are created each time the method is called, and are destroyed when it finishes executing.
Method Control Flow

- If the called method is in the same class, only the method name is needed
Method Control Flow

• The called method is often part of another class or object
Parameters

• When a method is called, the *actual parameters* in the invocation are copied into the *formal parameters* in the method header.

```java
char calc (int num1, int num2, String message)
{
    int sum = num1 + num2;
    char result = message.charAt (sum);
    return result;
}
```

```java
ch = obj.calc (25, count, "Hello");
```
Examples:

• Write a method that has one int parameter num, and prints “Happy Birthday” num times

• Write a method with two double parameters a and b that computes and returns the sum of squares: \( a^2 + b^2 \) of its two int parameters

• Write a method that has one int parameter num, and returns the String “Happy Birthday” num times

• Write a method that has one char parameter (assume it is a letter) and returns that char shifted forward 3 letters of the alphabet (with wrap-around). So \( a \rightarrow d, b \rightarrow e, \ldots, x \rightarrow a, y \rightarrow b, z \rightarrow c. \)
Homework

• Chapter 4, Section 4.4
• EX4.5- EX4.9

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

CSC 1051 M.A. Papalaskari, Villanova University