Methods

"Form Ever Follows Function"
Louis Henri Sullivan

Purpose of Methods

- Why do you think it is important for us to use methods?
- What things do we need to know about methods before we can use them?

Modularity

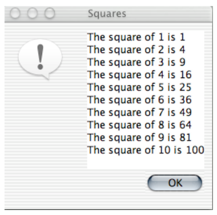
- Modularity is achieved in Java in two ways
  - Classes
  - Methods

Java API

- Math class
- One of the most useful Java classes
- **`Math.sqrt( 900.0 );`**
- **`Math.PI`**
- **`Math.E`**

Example

- Write a Java application that will display the squares of the first ten integers starting from 1.



```
The square of 1 is 1
The square of 2 is 4
The square of 3 is 9
The square of 4 is 16
The square of 5 is 25
The square of 6 is 36
The square of 7 is 49
The square of 8 is 64
The square of 9 is 81
The square of 10 is 100
```

Squares Program

```
public class SquareIntegers
{
 public static void main( String args[] )
 {
  JTextArea outputArea = new JTextArea( );

  int result;
  String output = "";

  for( int i=1; i<=10; i++)
  {
   result = square( i );
   output += "The square of " + i + " is " + result +"\n";
  }
```

Squares Program

```
 outputArea.setText( output );

  JOptionPane.showMessageDialog( null, outputArea,
  "Squares", JOptionPane.INFORMATION_MESSAGE );

 }//end main

 public static int square( int y )
 {
  return y*y;
 }//end square
}//end class
```

Calling Methods

- There are several ways to call methods
  - Just use the method name
    - `square( i )`
  - Use an objects followed by . and method name
    - `outputArea.setText( output );`
  - Use the class name to qualify the method
    - `JOptionPane.showMessageDialog( … )`
    - Only works with static methods

Properties of Methods

- Method header
  - `Return_value Method_name( argument1, argument2 )`
- Method body
  - `{ declarations, statements }`
- A method cannot be declared inside another method

Argument Promotion

- Applies to Java primitive types
- Primitive data types can be promoted to other data types
  - Similar to type casting in C++
- What problems can you see with this?

Promotion Rules

double	none
float	double
long	float or double
int	long, float or double
char	int, long, float or double
short	int, long, float or double
byte	short, int, long, float or double
boolean	none

Type Casting

- What would happen if we tried to force a promotion from a higher data type (**`double`**) to a lower data type (**`int`**)?
- For example, what would happen if we added the following statements to our squares program:
  - `result = square( 9.5 );`
  - `output += result;`

Type Casting

- The compiler throws an error:

```
SquareIntegers.java:18: square(int) in SquareIntegers
  cannot be applied to (double)
 result = square(9.5);
         ^
```

- We must force the type casting

```
result = square( (int) 9.8 );
output += "The square of 9.8 is " + result
 + "\n";
```

GUI

- A GUI presents a user-friendly mechanism for interacting with a program
- GUI’s are built from GUI components
  - Controls or widgets
- Examples of widgets:
  - `JTextArea`
  - `JTextField`
  - `JLabel`
  - `JButton`

GUI

- So far we have
  - Displayed messages in dialog boxes
    - `JOptionPane.showMessageDialog( null, “Message”, “Title”, JOptionPane.INFORMATION_MESSAGE );`
  - Receive input from the user through a dialog box
    - `JOptionPane.showInputDialog( “Enter Number” );`
  - Added a widget to a dialog box
    - `JOptionPane.showMessageDialog( null, widget, “title”, JOptionPane.INFORMATION_MESSAGE );`

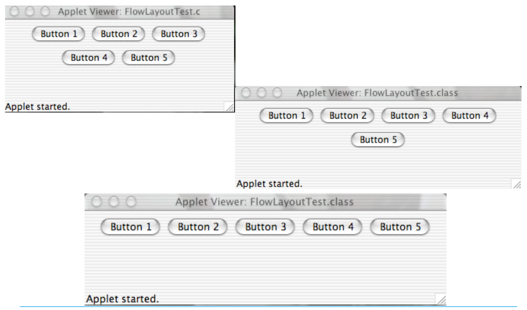
GUI

- GUI widgets can be placed in their own window
  - Not using dialog boxes
- To do this in applets, widgets need to be placed on to the onscreen display area
  - Called the content pane
- The location where the widgets get placed depends on the layout manager

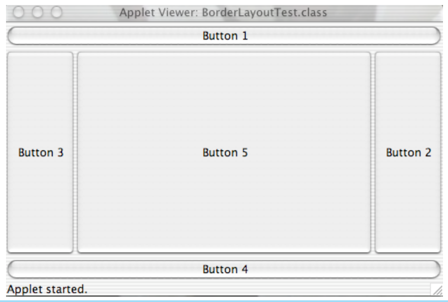
Layout Managers

- Programmer can specify where the widget is going to appear on the window using the X and Y coordinates
- Use layout manager:
  - FlowLayout
    - Widgets added sequentially, left to right
  - BorderLayout
    - Widgets are added to NORTH, SOUTH, EAST, WEST or CENTER
  - GridLayout
    - Arranges components into rows and columns

FlowLayout

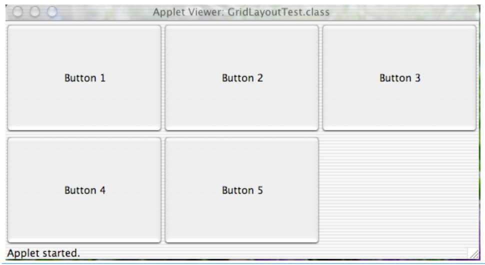


BorderLayout



GridLayout

- 2 rows, 3 columns



Constants

- Constants are declared in Java using the keyword **`final`**
- **`final int WON = 0, LOST = 1, CONTINUE = 2;`**

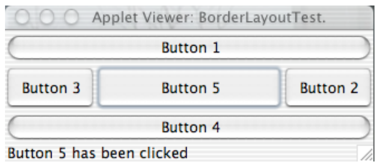
Event Handling

- To handle user interaction with the GUI, Java programs must contain the necessary event handling code
- Event handling is performed by
  - Implementing the interface `ActionListener`
    - **`public class BorderLayoutTest extends JApplet implements ActionListener { }`**
  - Adding the method `ActionPerformed`
    - **`public void actionPerformed( ActionEvent actionEvent ) { }`**

Event Handling

- We need to also import the necessary classes
  - **`import java.awt.event.*;`**
- Add an action listener to each widget that accepts an event from the user
  - **`button1.addActionListener( this );`**

Event Handling Program



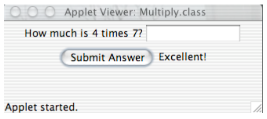
Button 5 has been clicked

Multiplication Program

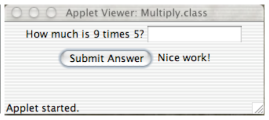
- Write a program that will assist children in learning multiplication of single digit numbers. Your program should display a text label asking the children the question, a text field for the user to enter the result, a button to submit the answer and a label displaying if the answer was correct or incorrect
- If the answer was incorrect, the user should be asked the same question again, otherwise the numbers change

Multiplication Program

- The message displayed after the user has submitted an answer should randomly choose from a list of different messages, so that the user doesn't get bored



How much is 4 times 7?
Excellent!
Applet started.



How much is 9 times 5?
Nice work!
Applet started.

Fields

- Fields are variables that have been declared inside a class but outside of any method
- These fields are accessible to all methods
- Called *instance variables* in other programming languages

Scope of Declarations

- The scope of a parameter declaration is the body of the method in which the declaration appears
- The scope of a local variable declaration is from the point at which the declaration appears in the block to the end of that block
- The scope of a local variable declaration that appears in the initialisation of a **for** statement's header is the body of the **for** statement

Scope of Declarations

- The scope of a method or field of a class is the entire body of the class
- If a local variable or parameter in a method has the same name as a field, the field is "hidden" until the block terminates executing
- This is called shadowing

What is the Output?

Method Overloading

- Several methods of the same name can be declared in the same class as long as they have different parameters
- This is called method overloading
- Example:
  - `Public int square( int x ) { }`
  - `Public double square( double x ) { }`

Summary

- Today we covered:
  - Random numbers
  - GUI components
  - GUI layout managers
  - Event handling
  - Scope of declaration
  - Method overloading
- Reading:
  - Today we covered Chapter 6 up till 6.12
  - Remainder of chapter 6 covers recursion (not required)
  - So far we have completed chapters 1 through 7
  - Next time we will start on object based programming (chapter 8)