

# **Event-Driven Programs**

**CS 1025 Computer Science Fundamentals I**

**Stephen M. Watt**

***University of Western Ontario***

# Command Line vs Events

- So far, all our programs have started from a “main” method which calls other methods, takes input, makes objects, prints output, etc.
- Another model of programming is to have programs that respond to “events.”
- These events could be
  - Mouse clicks
  - Key presses
  - Pressing a brake pedal
  - Lifting a telephone receiver

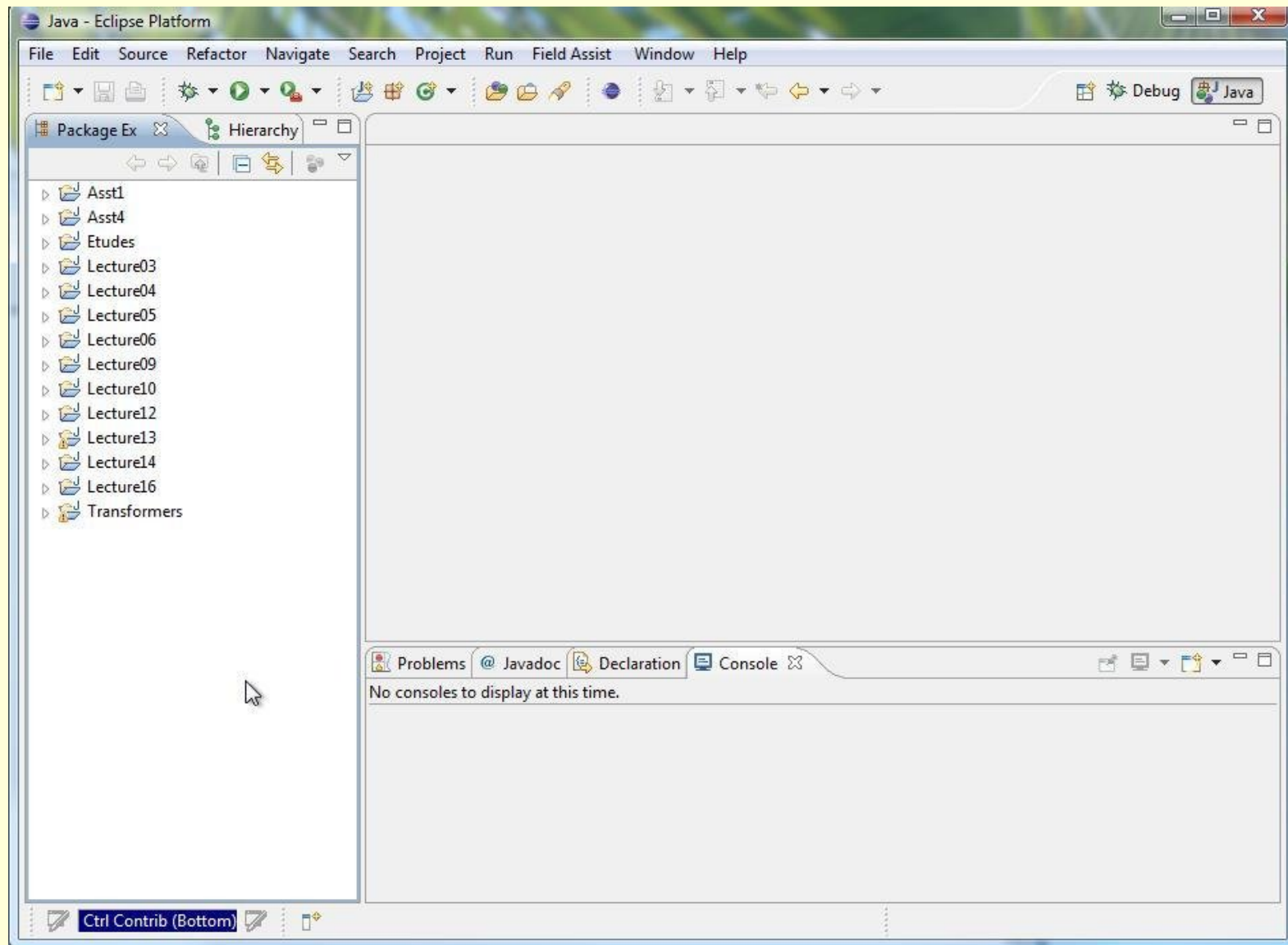
# Event-Driven Code

- Event-driven code associates particular functions or methods to be called when specified events occur.
- In Java, this is done by creating “handler” objects from which specific methods get invoked.
- These handler objects are then placed on the things that receive events, like buttons, scroll bars etc.

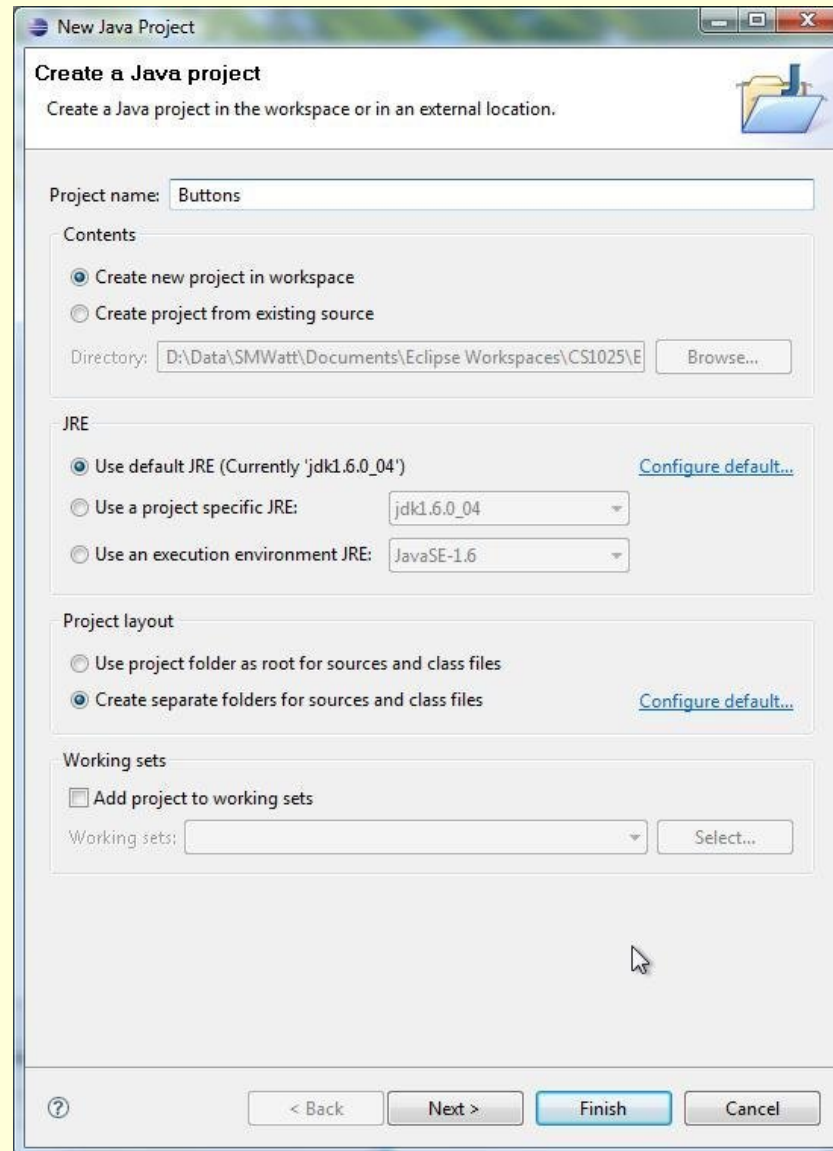
# GUIs in Java

- Graphical user interfaces are programmed in Java using
  - the abstract window toolkit (old, lower level) and
  - Swing (new, higher level)
- The next slides give an example of creating a simple Swing application.

# Start Eclipse



# Create a Project



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The dialog is titled 'New Java Project' and has a subtitle 'Create a Java project in the workspace or in an external location.' The 'Project name' field is set to 'Buttons'. Under the 'Contents' section, the radio button 'Create new project in workspace' is selected. The 'Directory' field shows the path 'D:\Data\SMWatt\Documents\Eclipse Workspaces\CS1025\E' with a 'Browse...' button next to it. In the 'JRE' section, the radio button 'Use default JRE (Currently 'jdk1.6.0\_04')' is selected, with a 'Configure default...' link to its right. The 'Use a project specific JRE' option is also visible with a dropdown menu showing 'jdk1.6.0\_04'. The 'Use an execution environment JRE' option is also visible with a dropdown menu showing 'JavaSE-1.6'. In the 'Project layout' section, the radio button 'Create separate folders for sources and class files' is selected, with a 'Configure default...' link to its right. The 'Use project folder as root for sources and class files' option is also visible. In the 'Working sets' section, the checkbox 'Add project to working sets' is unchecked. The 'Working sets' field is empty with a 'Select...' button next to it. At the bottom of the dialog, there are four buttons: '?', '< Back', 'Next >', and 'Finish'. The 'Finish' button is highlighted in blue.

**New Java Project**

Create a Java project in the workspace or in an external location.

Project name:

**Contents**

☒ Create new project in workspace  
☐ Create project from existing source

Directory:

**JRE**

☒ Use default JRE (Currently 'jdk1.6.0\_04') [Configure default...](#)  
☐ Use a project specific JRE:   
☐ Use an execution environment JRE:

**Project layout**

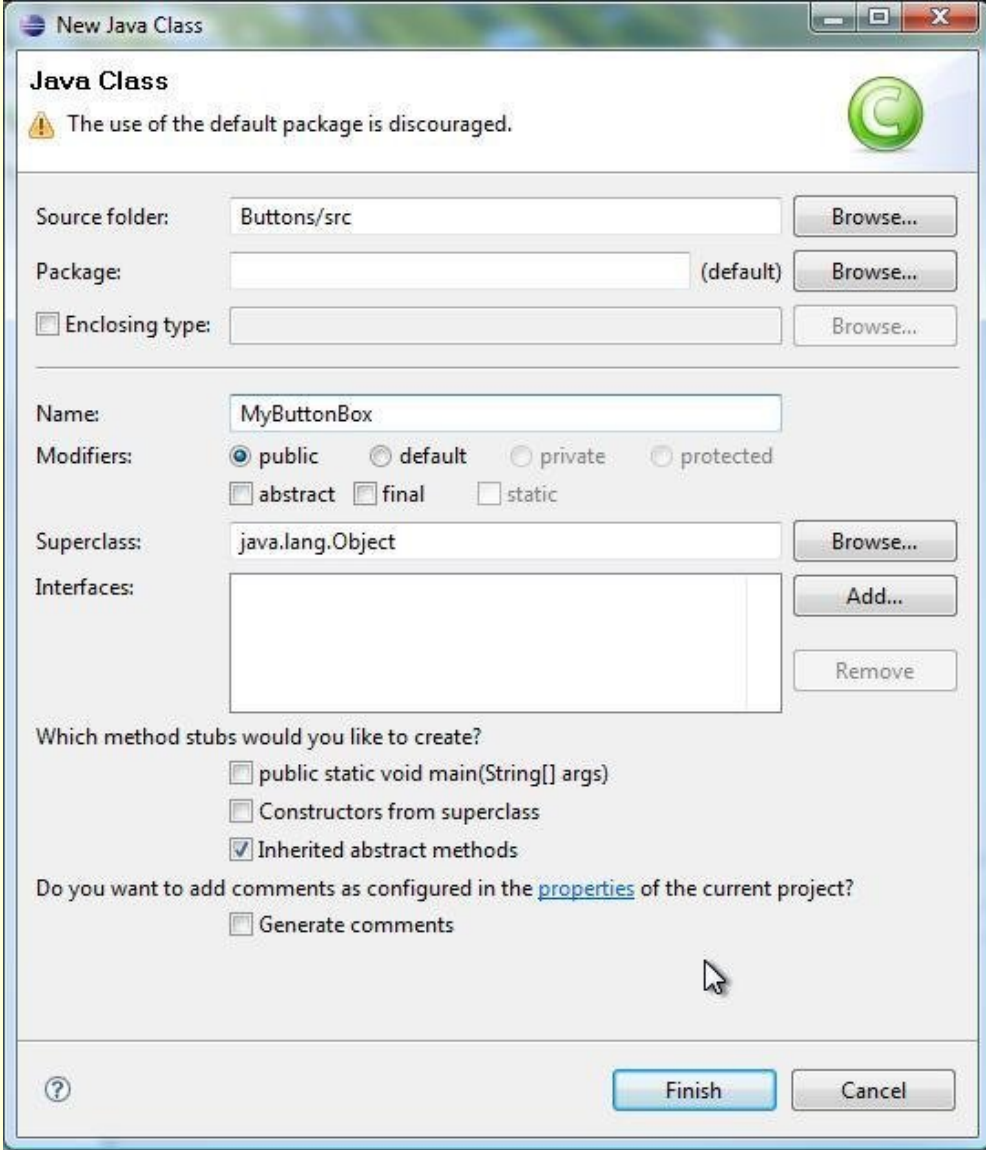
☐ Use project folder as root for sources and class files  
☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**

☐ Add project to working sets

Working sets:

# Create a Class



The image shows a 'New Java Class' dialog box from an IDE. It has a title bar with standard window controls. The main area is titled 'Java Class' and includes a warning icon and text: 'The use of the default package is discouraged.' Below this, there are three rows for specifying the location: 'Source folder' (Buttons/src), 'Package' (empty, with '(default)' text), and 'Enclosing type' (empty). Each row has a 'Browse...' button. The 'Name' field contains 'MyButtonBox'. The 'Modifiers' section has radio buttons for 'public' (selected), 'default', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'. The 'Superclass' field contains 'java.lang.Object' with a 'Browse...' button. The 'Interfaces' section has an empty list box with 'Add...' and 'Remove' buttons. A section titled 'Which method stubs would you like to create?' has checkboxes for 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods' (checked). Below this is a question about adding comments as configured in the project's properties, with a 'Generate comments' checkbox. At the bottom are 'Finish' and 'Cancel' buttons, and a help icon (?) on the left.

**New Java Class**

**Java Class**

⚠ The use of the default package is discouraged.

Source folder: Buttons/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: MyButtonBox

Modifiers: ☒ public ☐ default ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...  
Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments as configured in the [properties](#) of the current project?  
☐ Generate comments

? Finish Cancel

# The Program – note the Listener

