

ECE 462

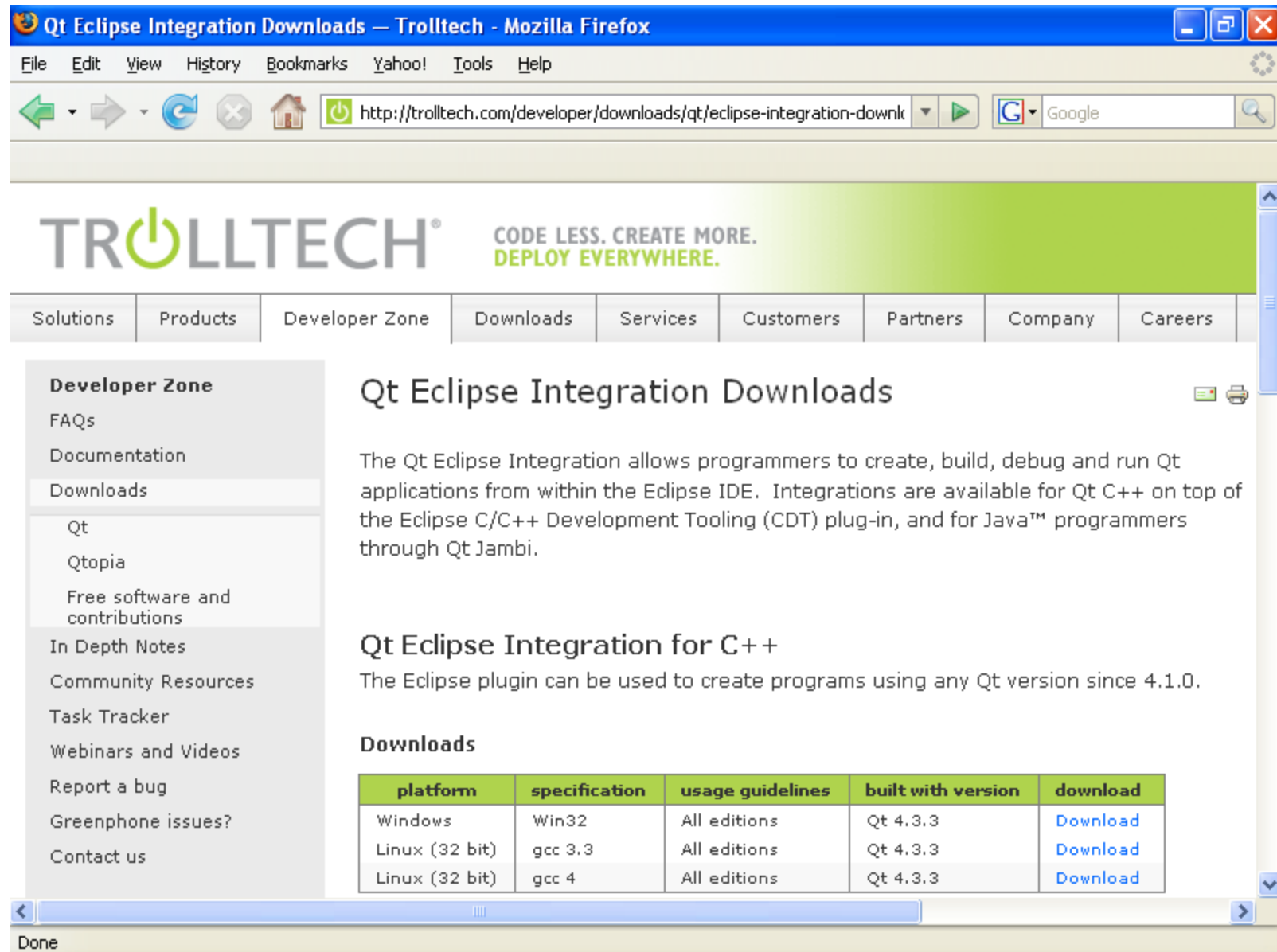
**Object-Oriented Programming
using C++ and Java**

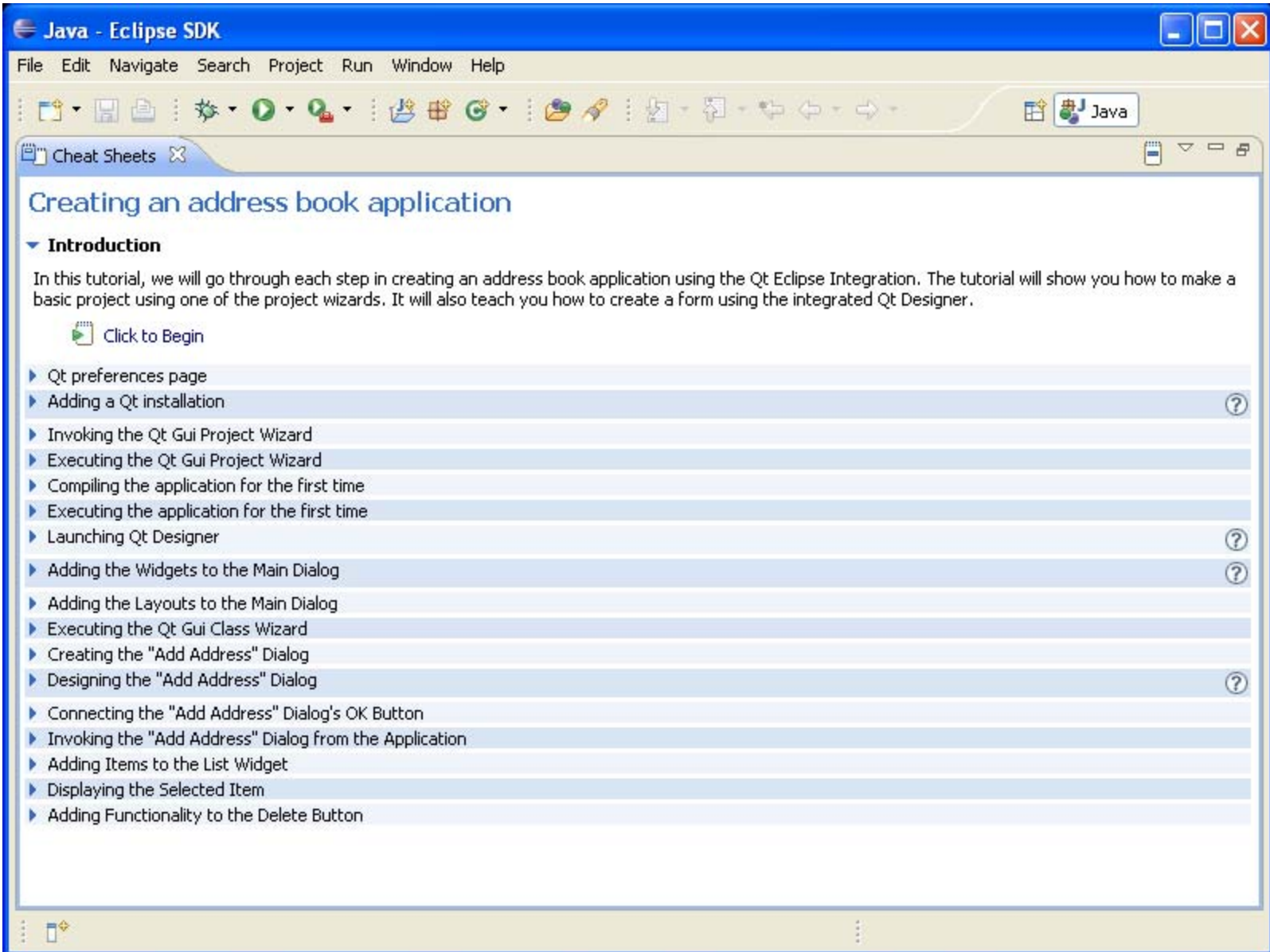
Graphical User Interface

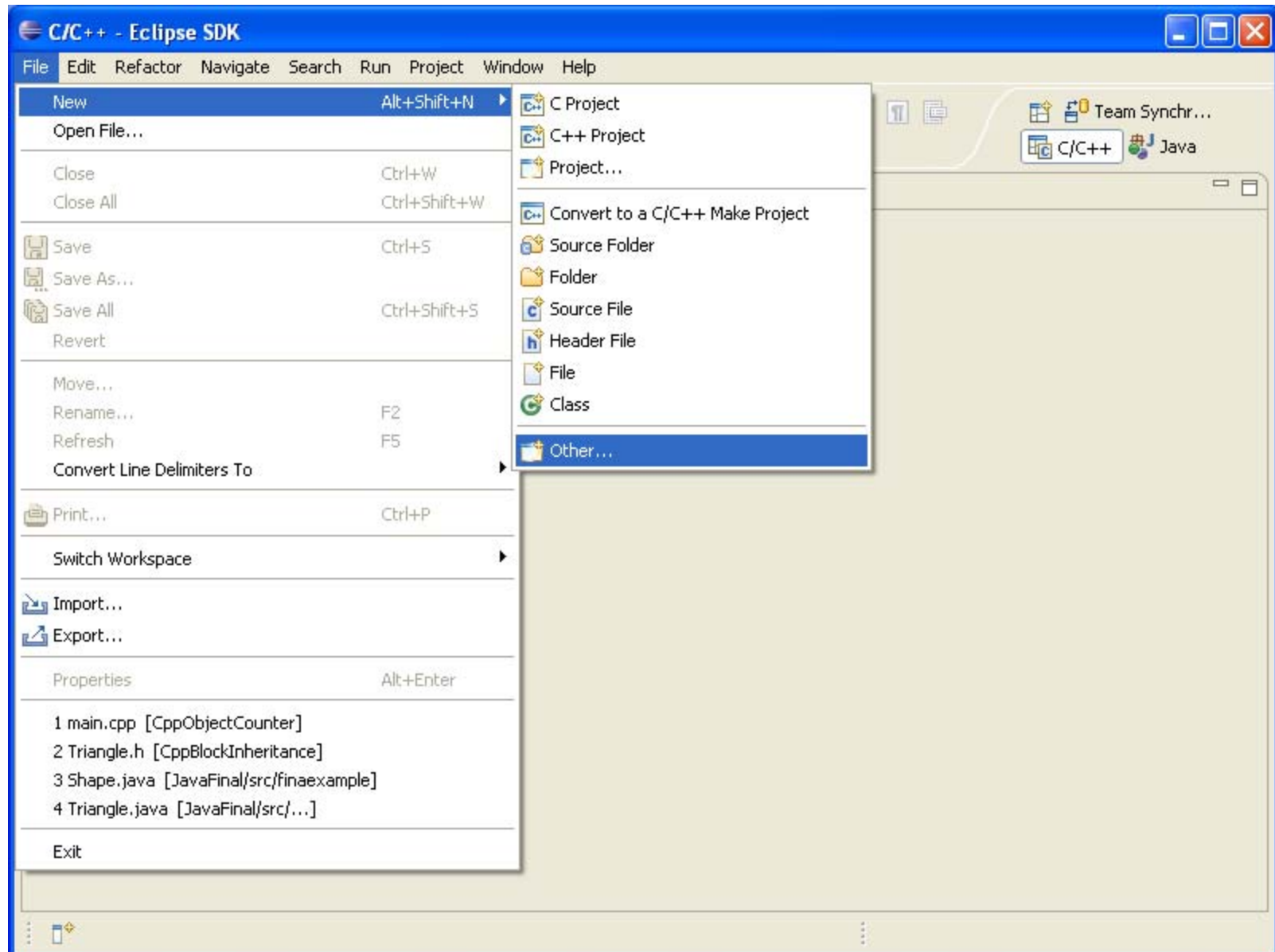
Yung-Hsiang Lu
yunглу@purdue.edu

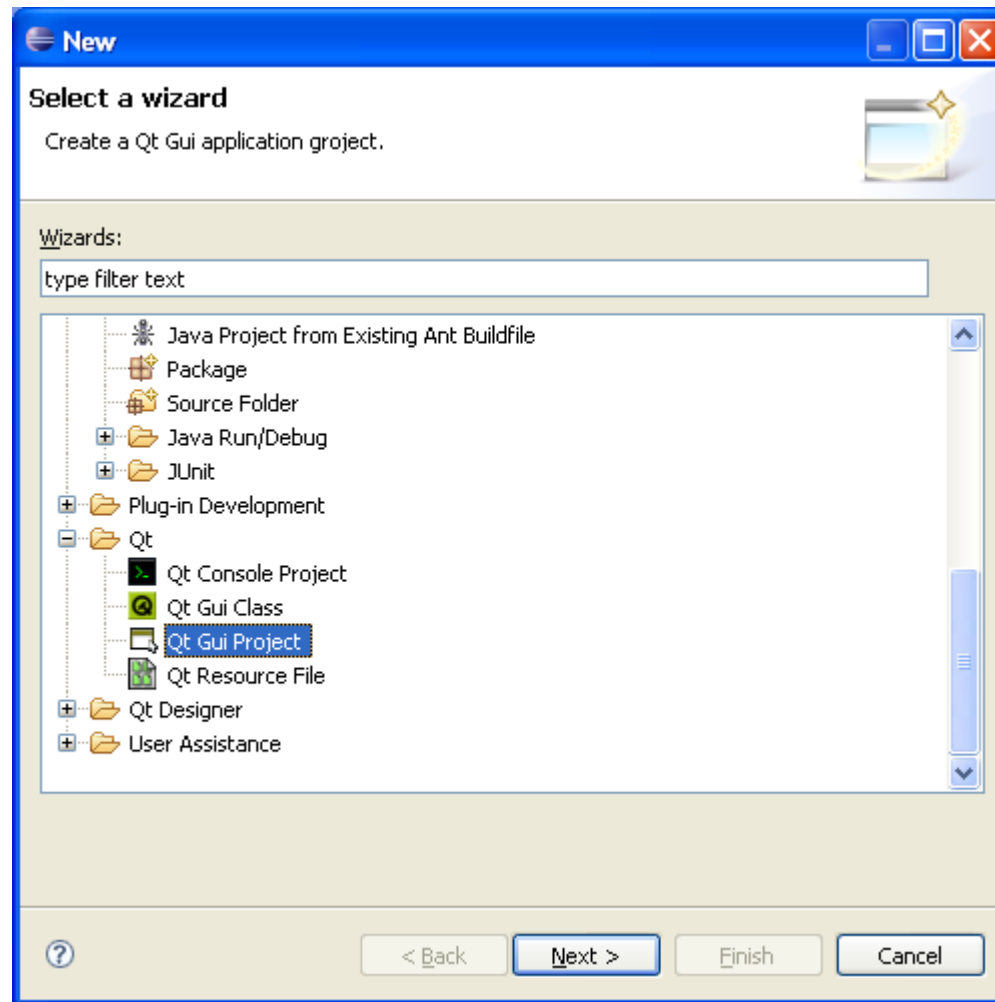
GUI using C++ / Qt

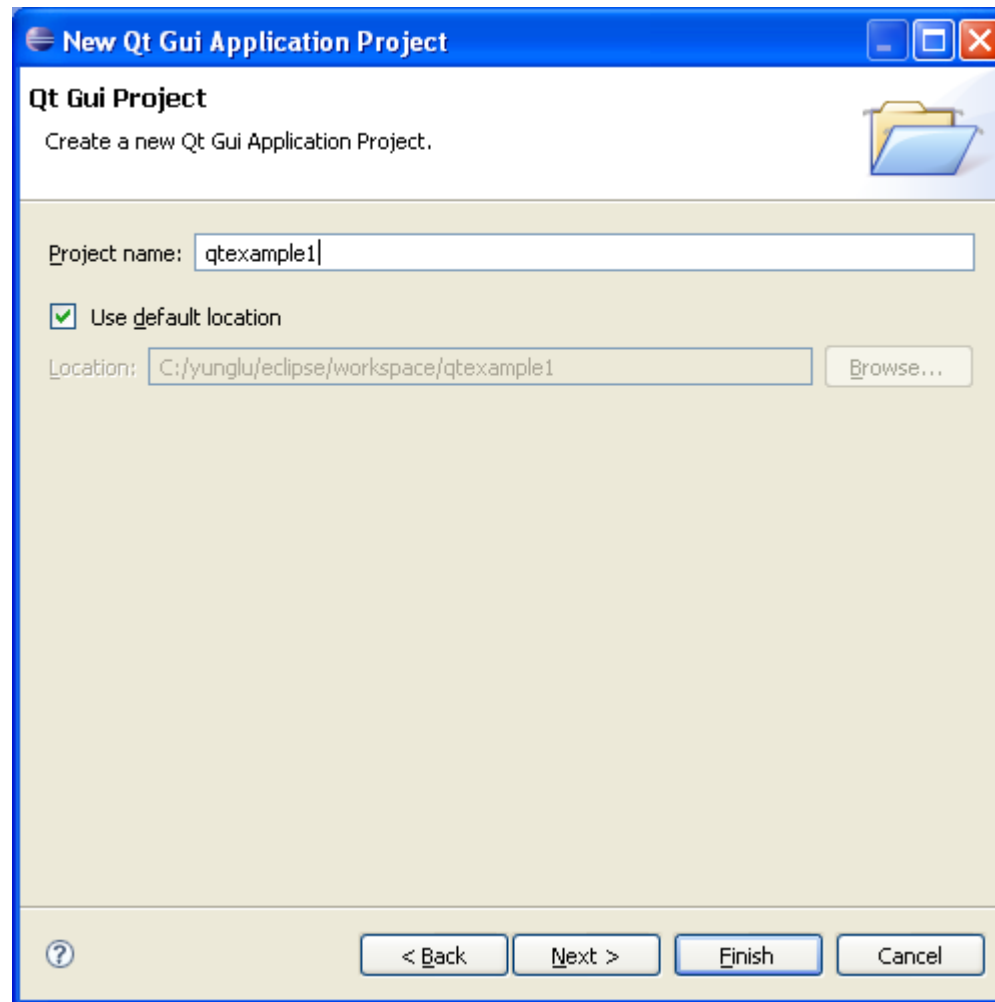
Qt + Eclipse

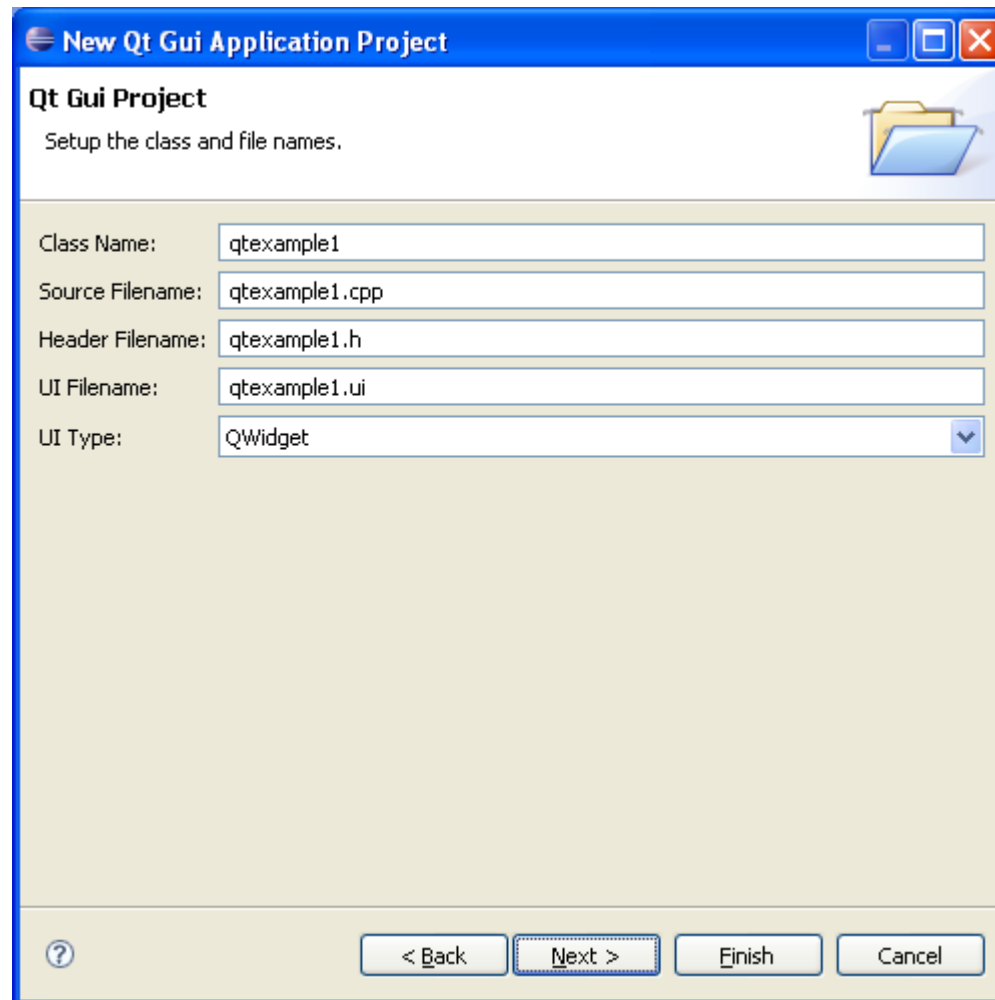


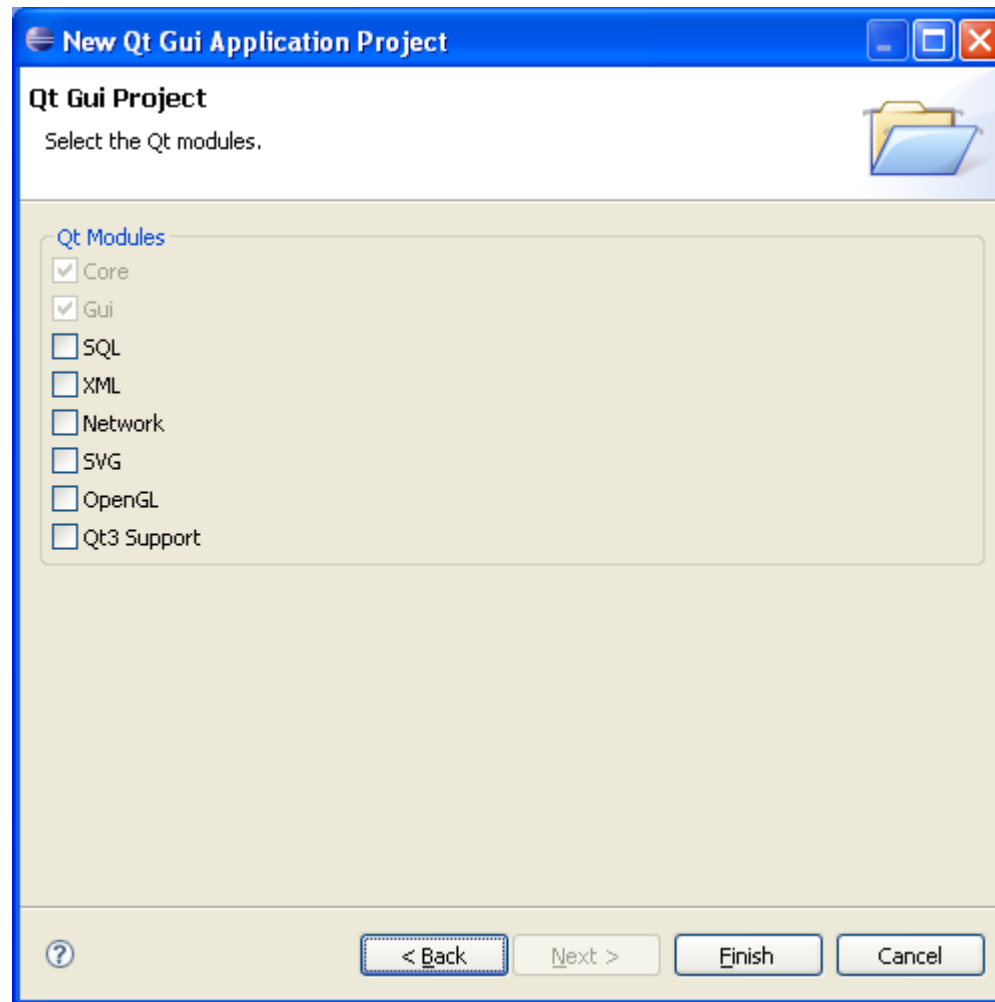


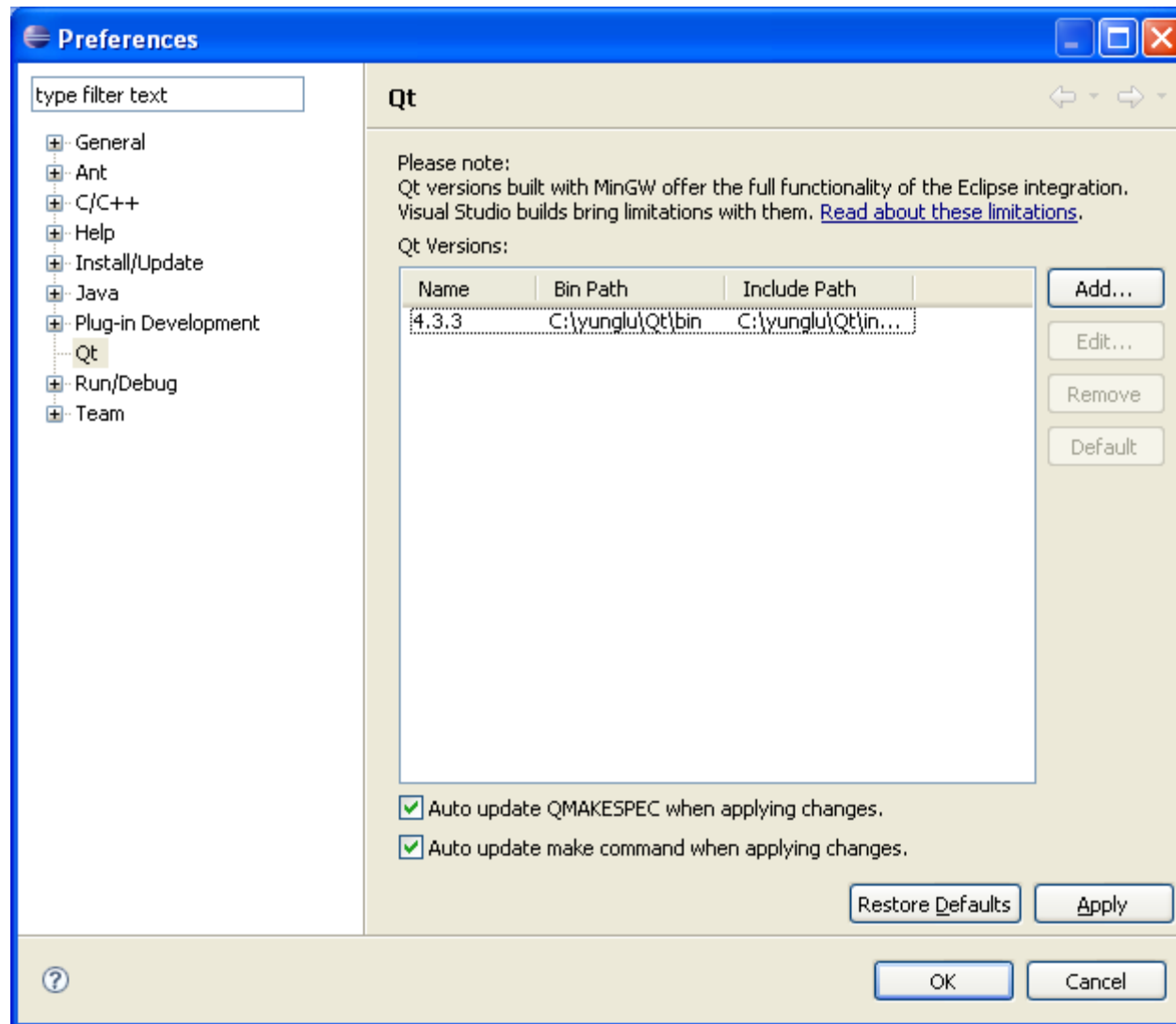


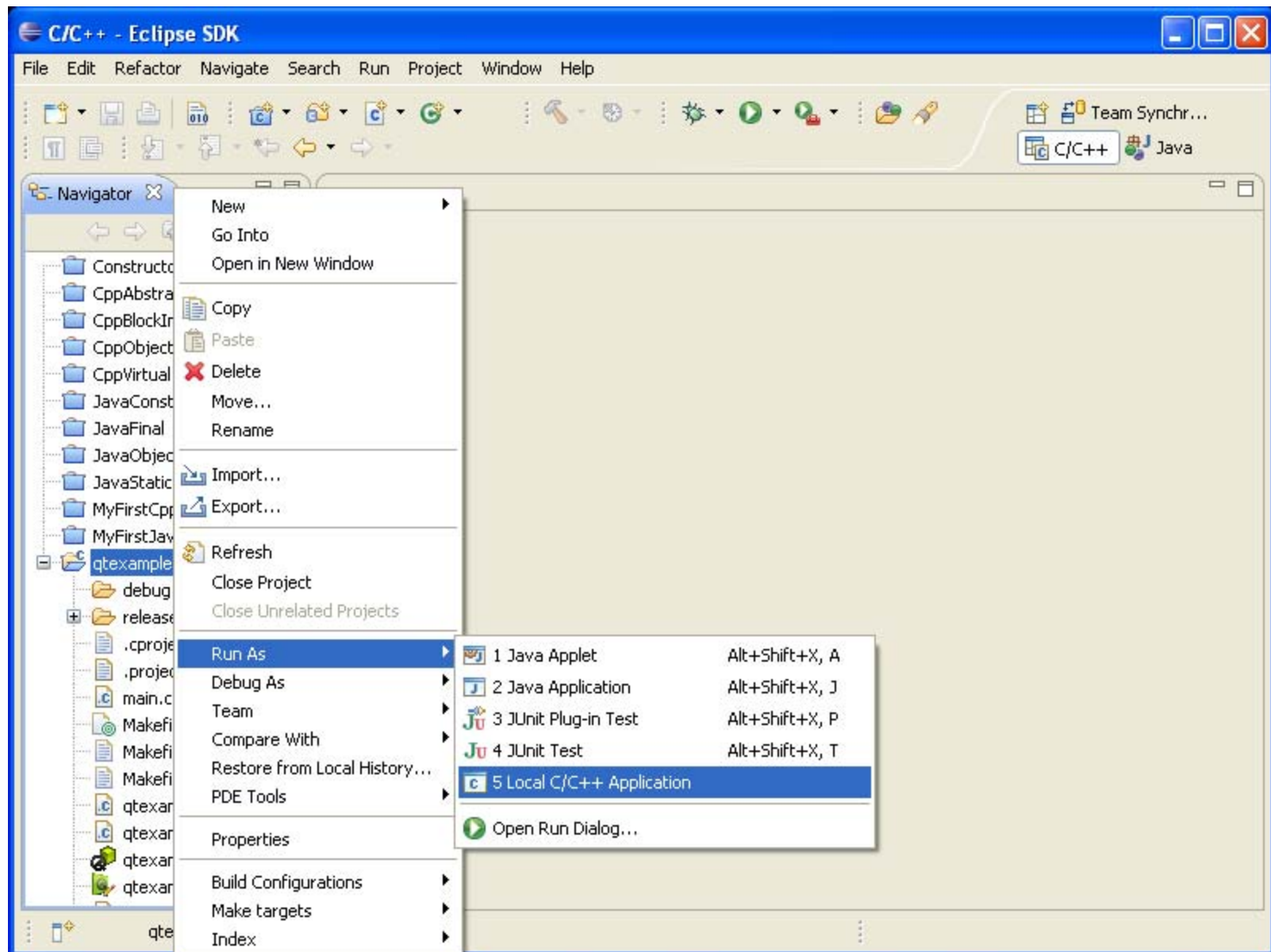


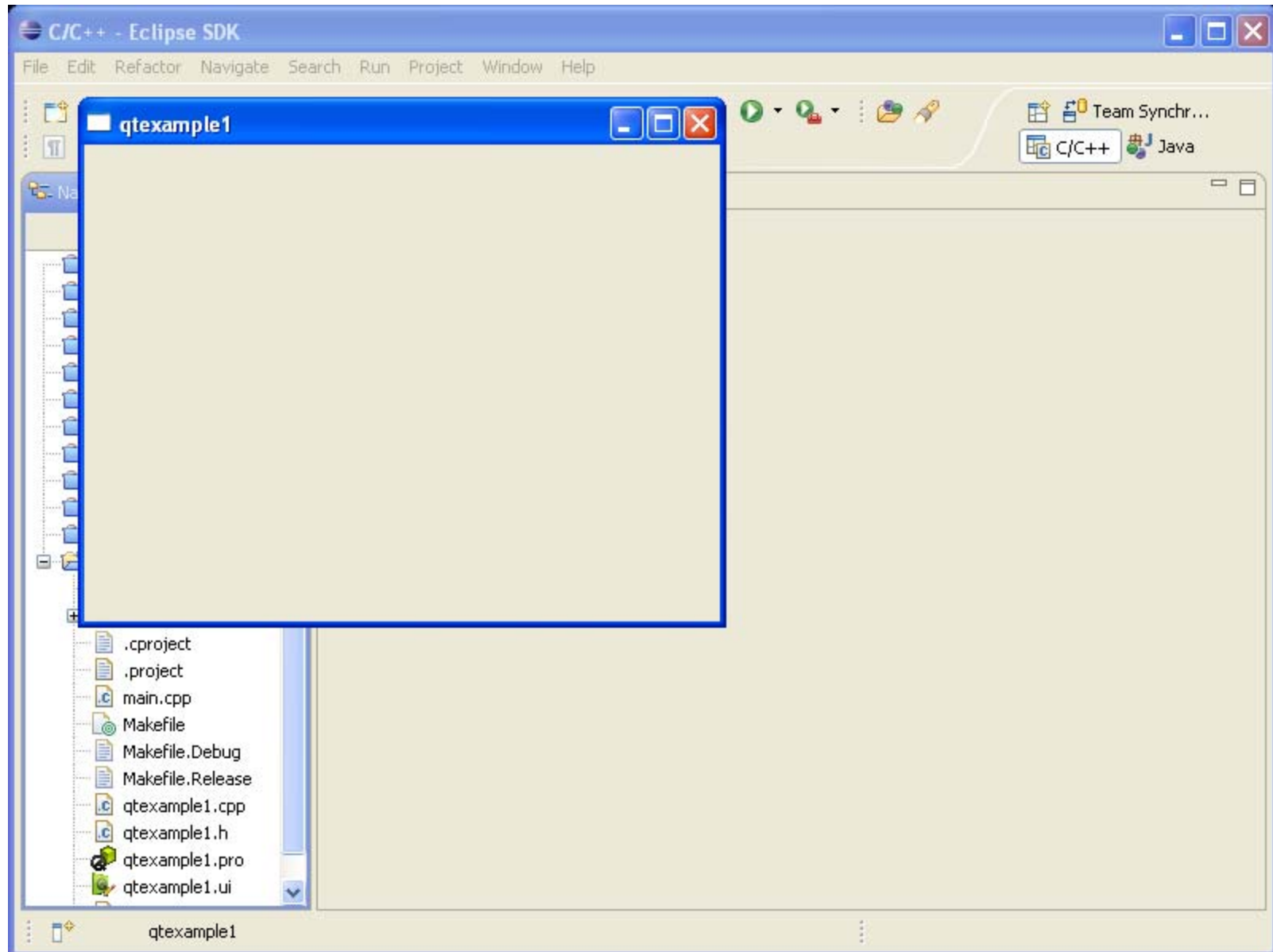




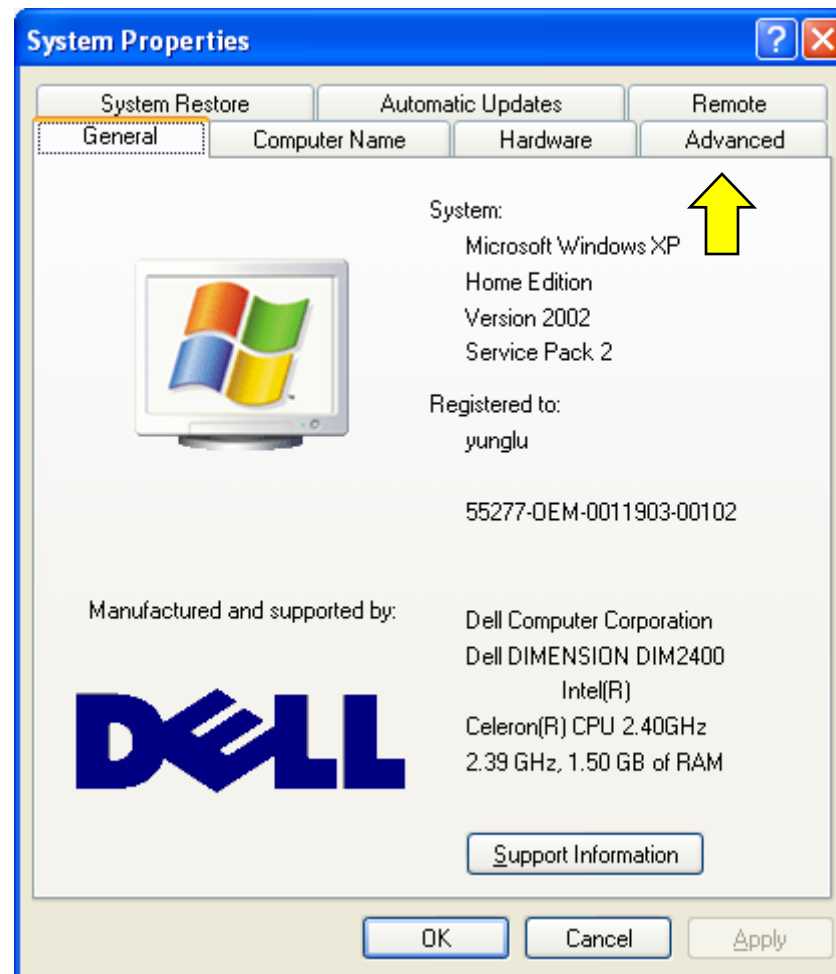


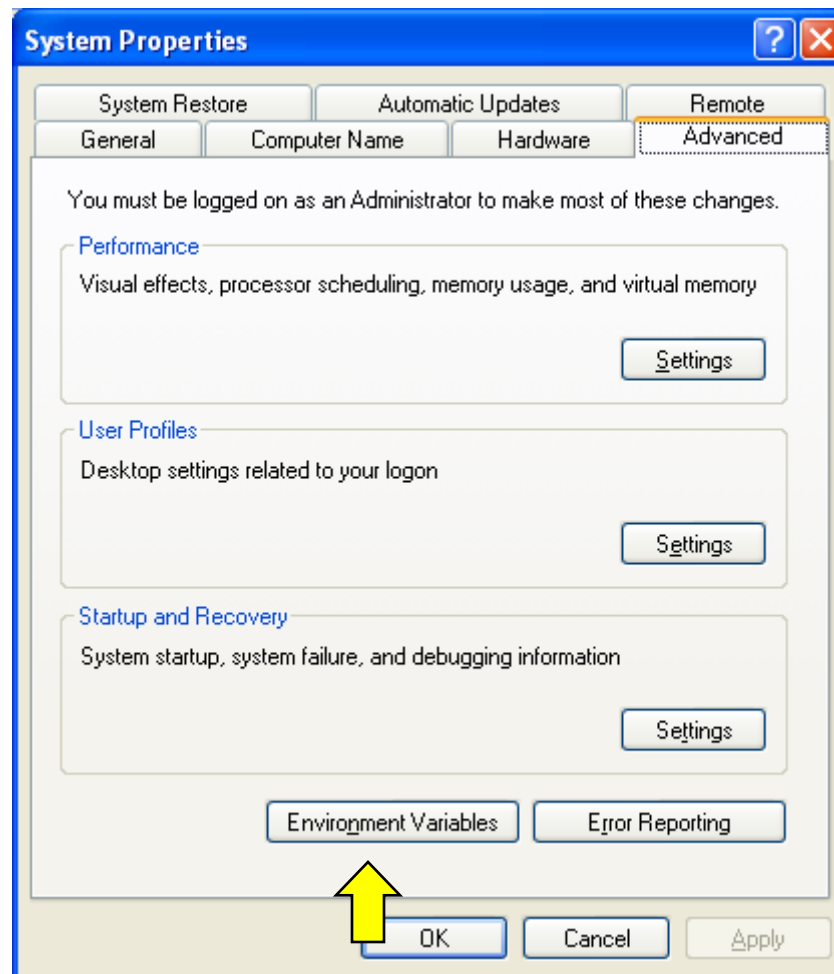


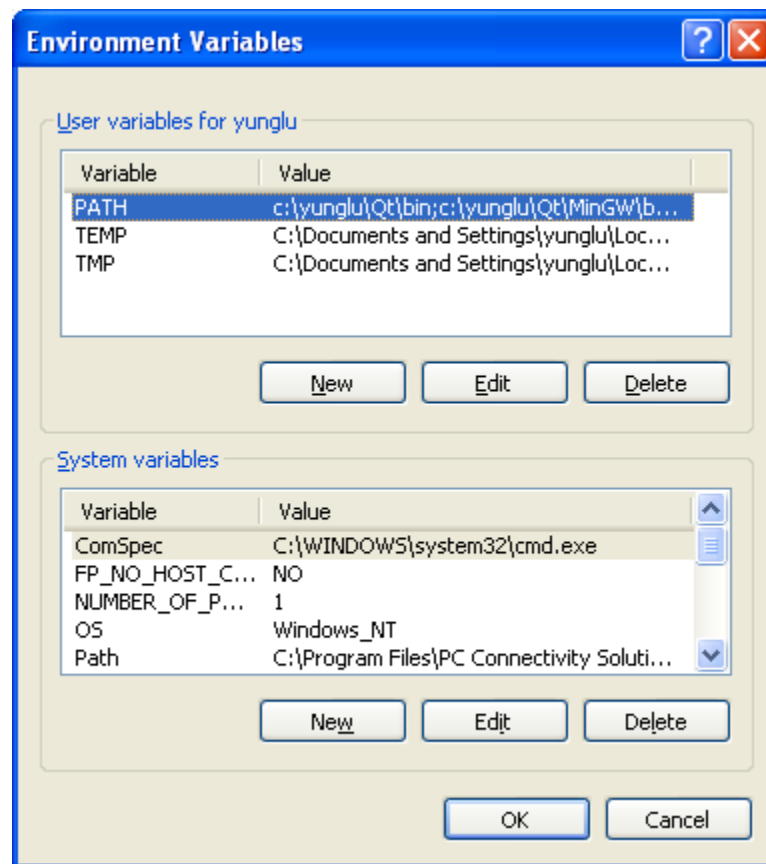




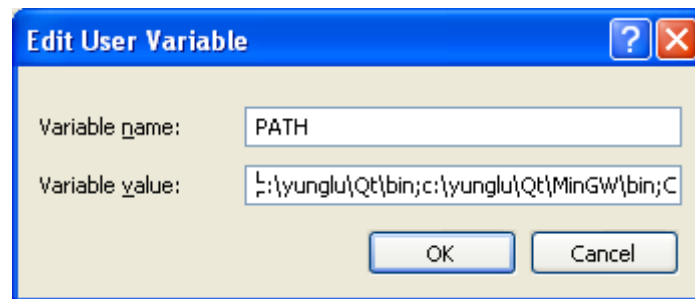
Path Setting (if there are problems)

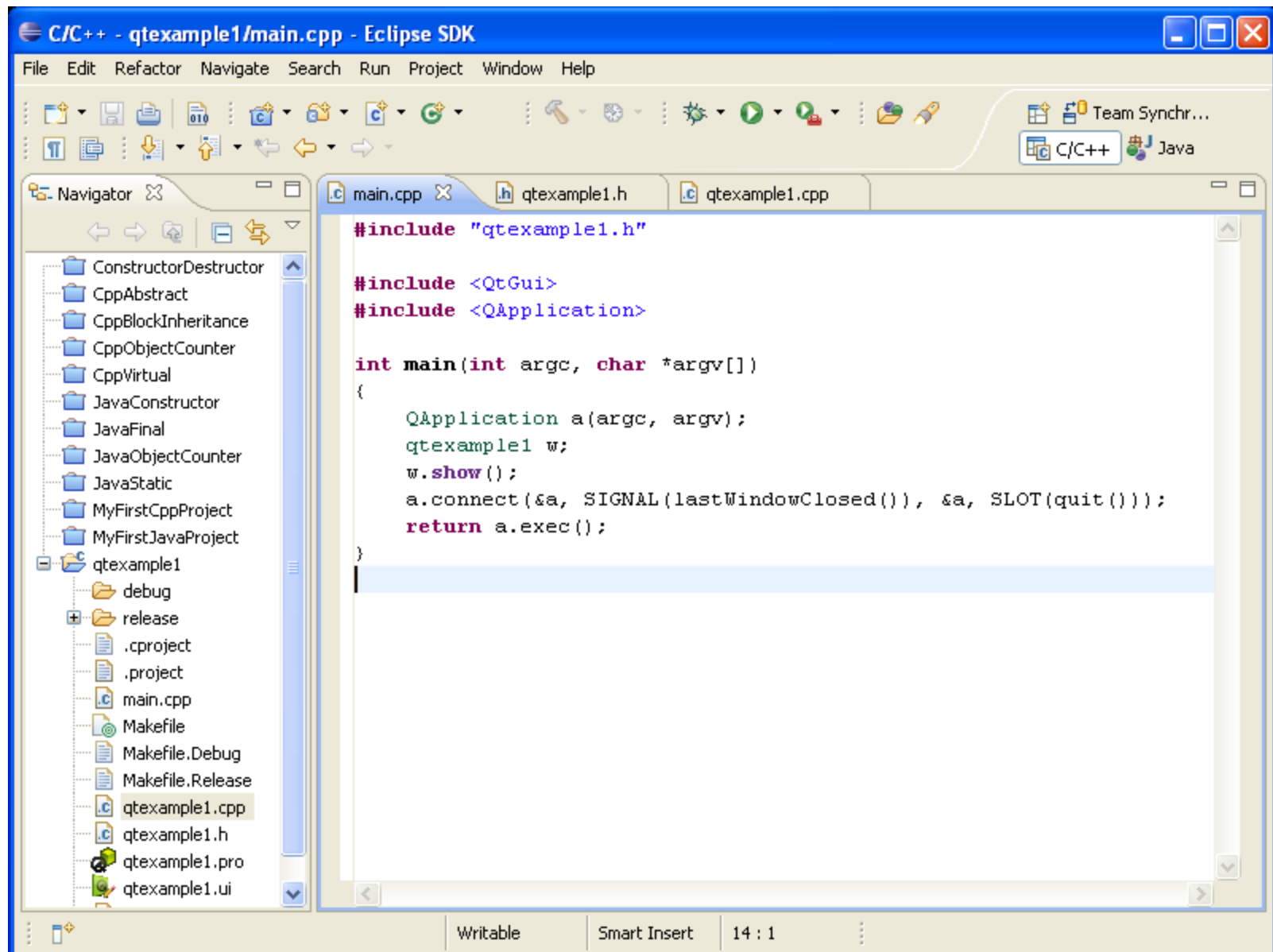


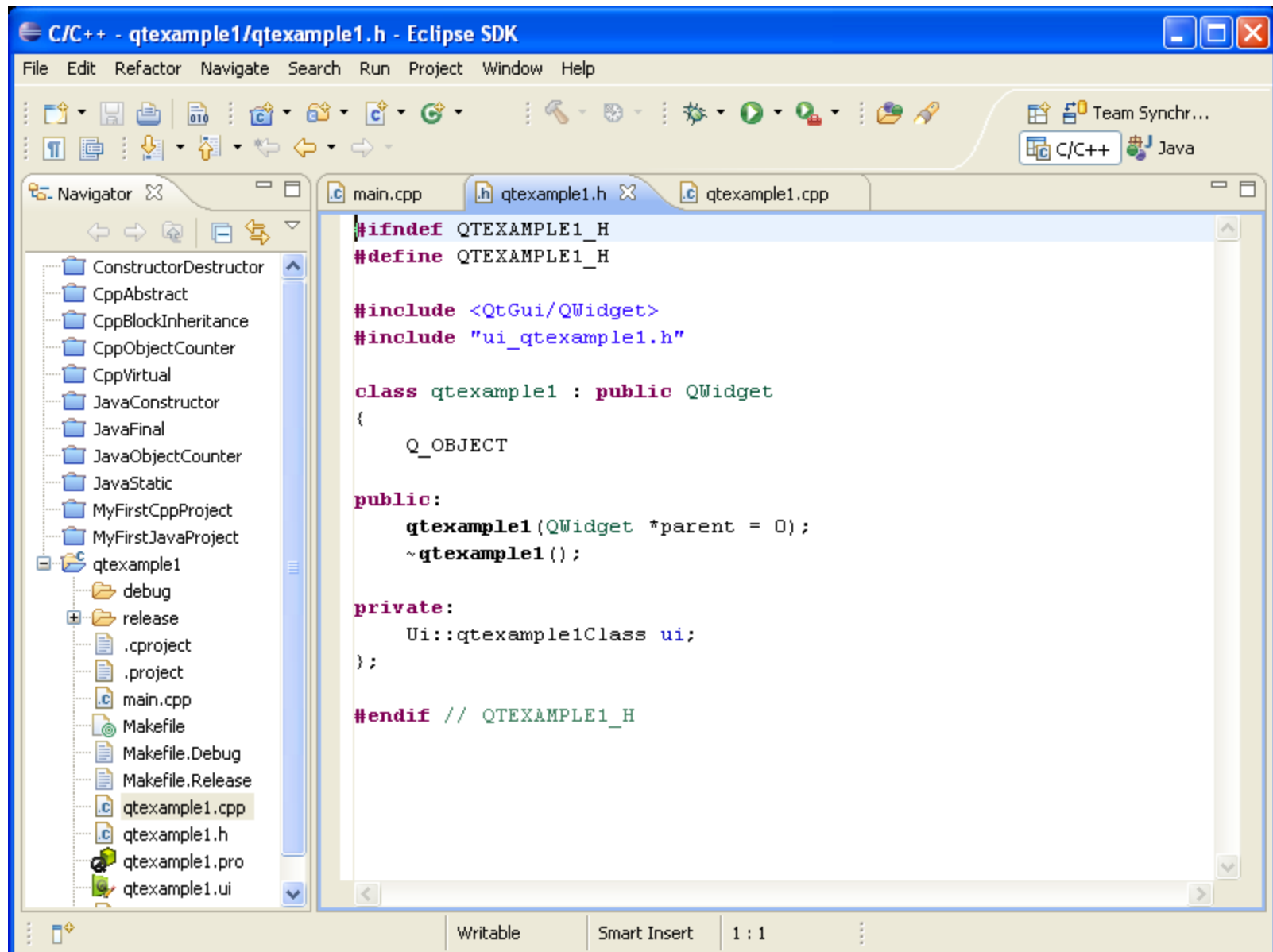


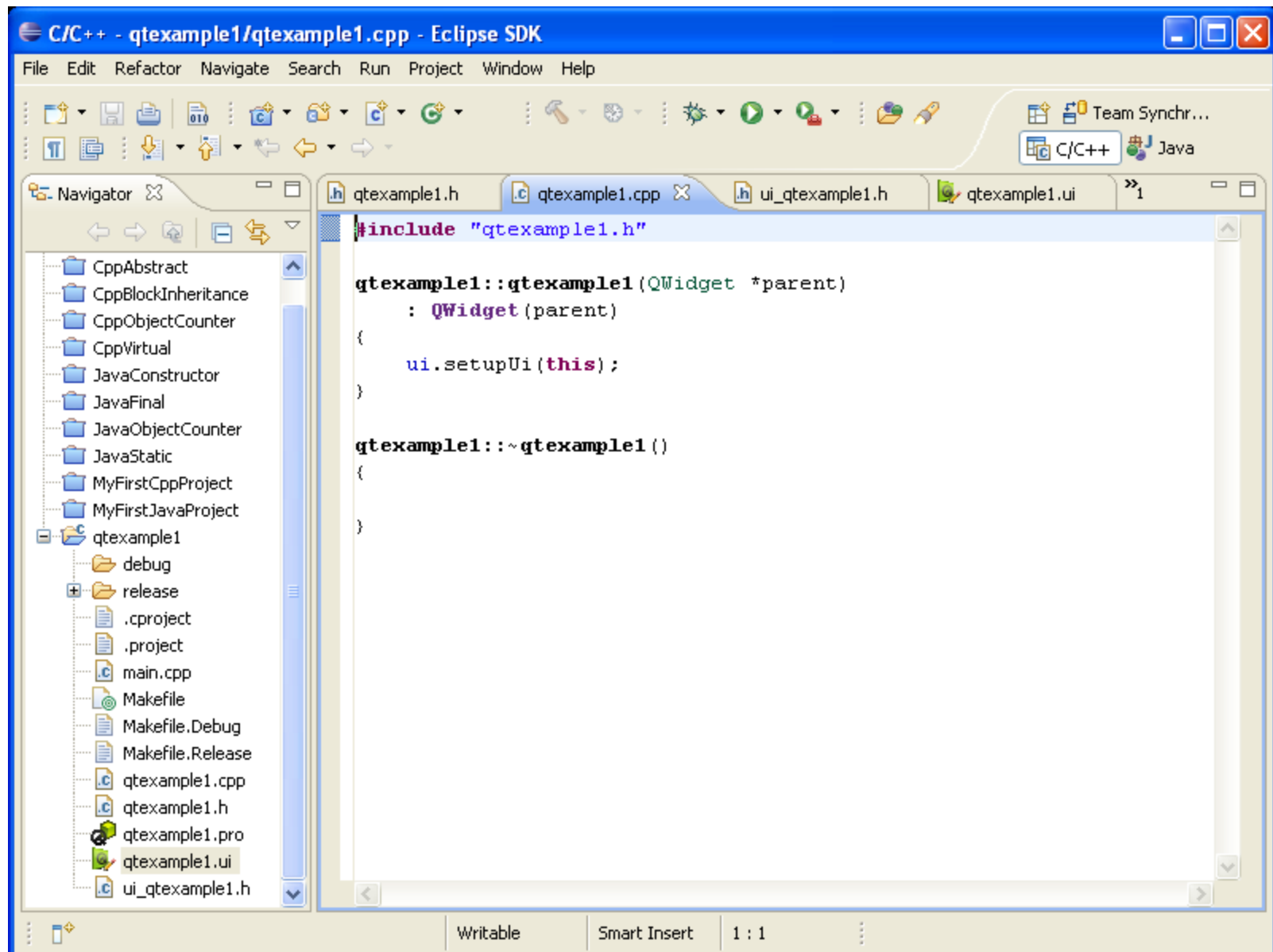


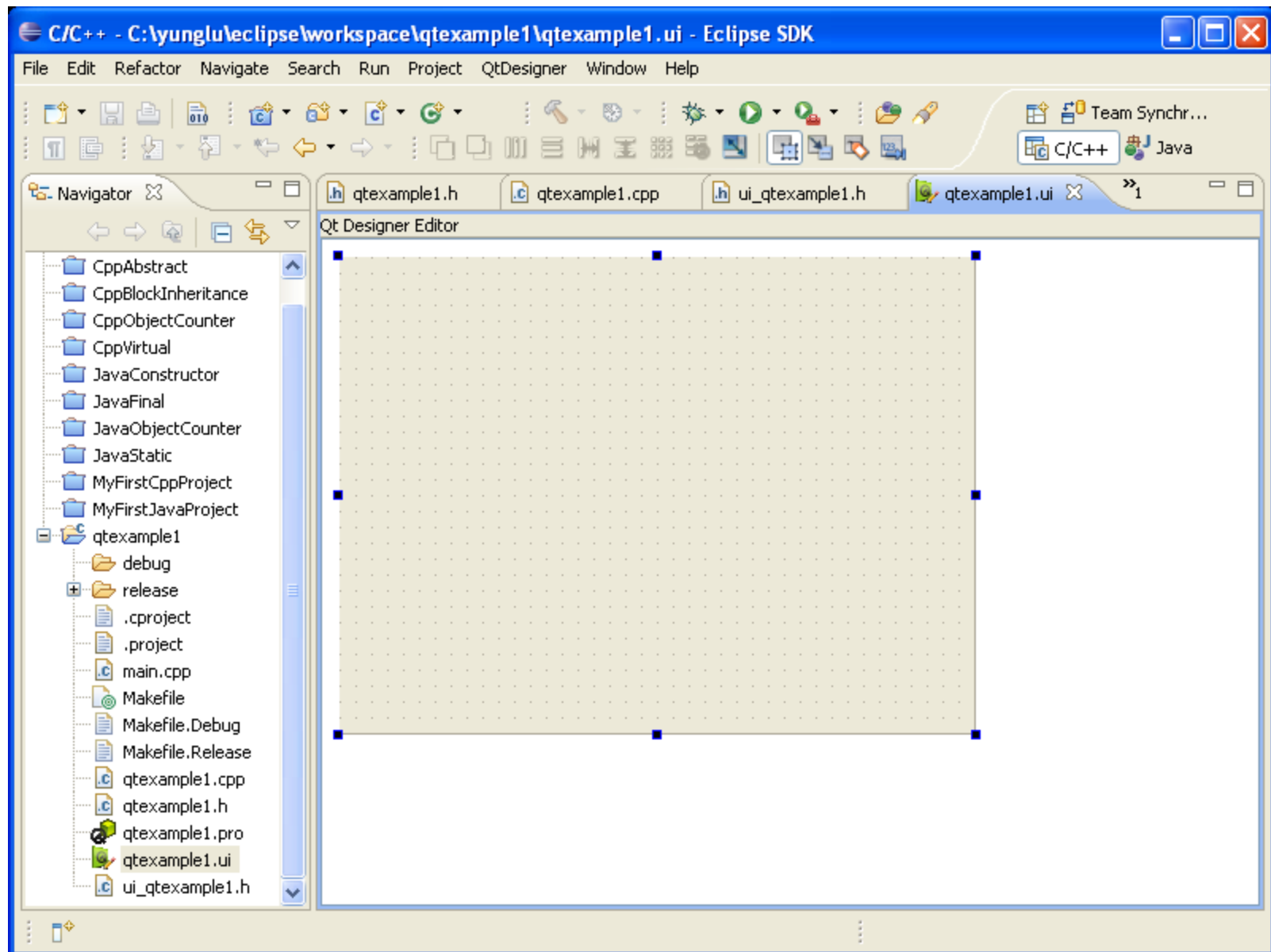
Qt\bin and MinGW\bin should be in the path



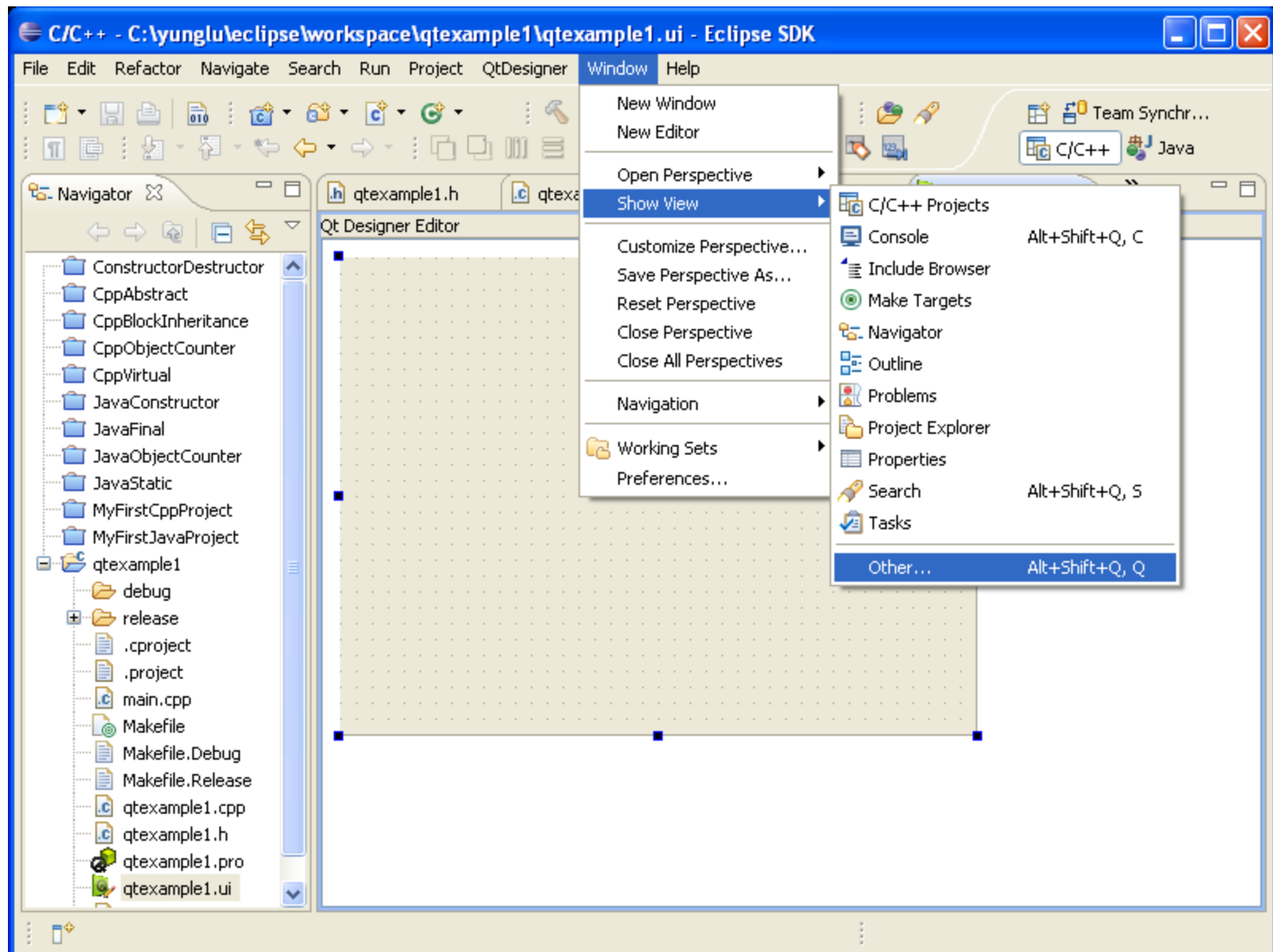


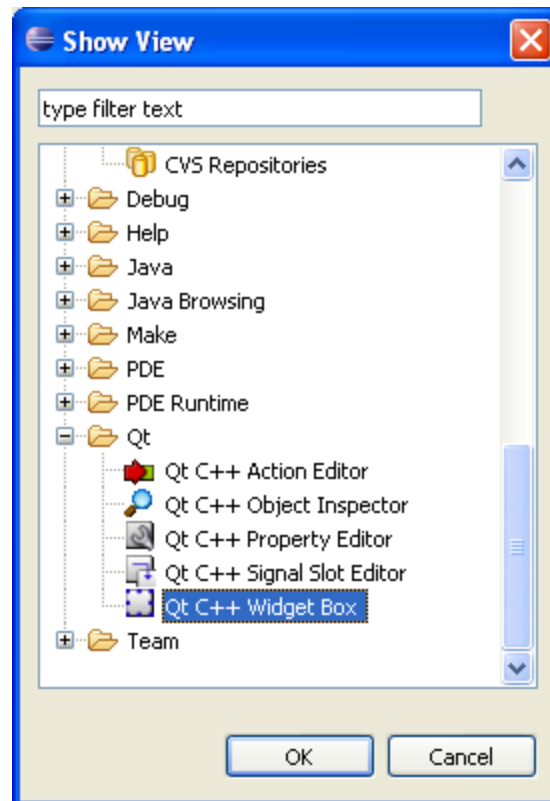


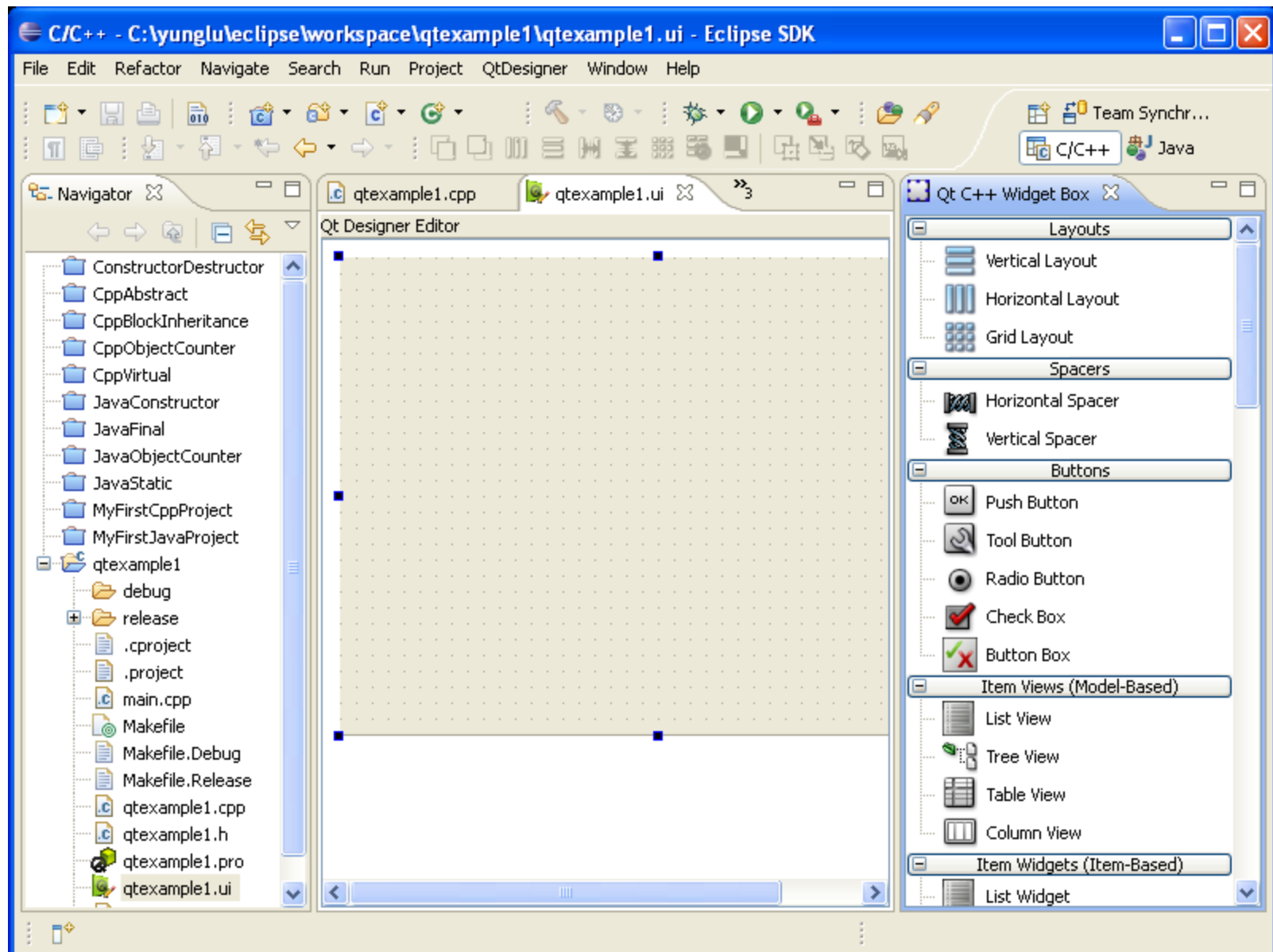


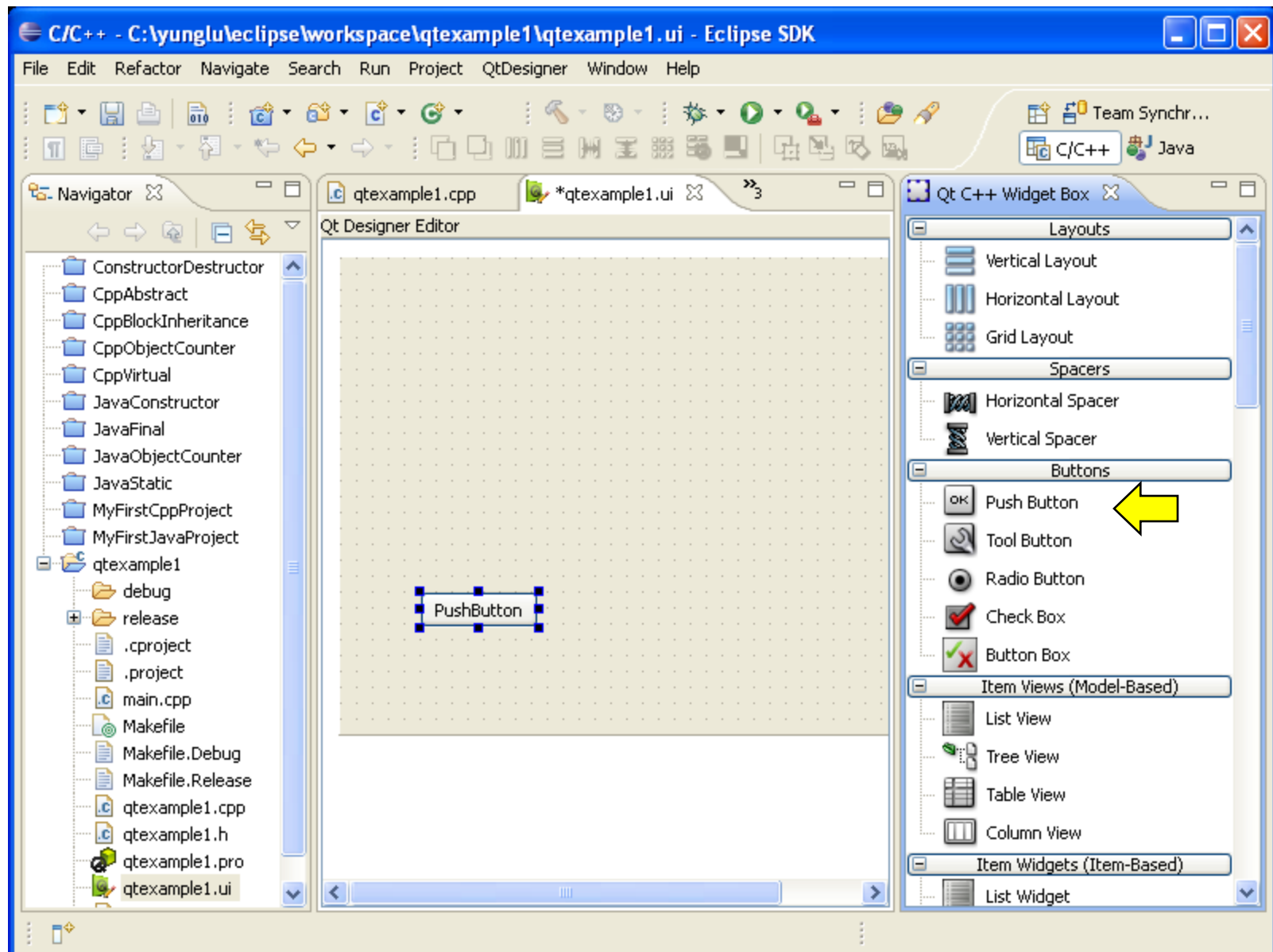


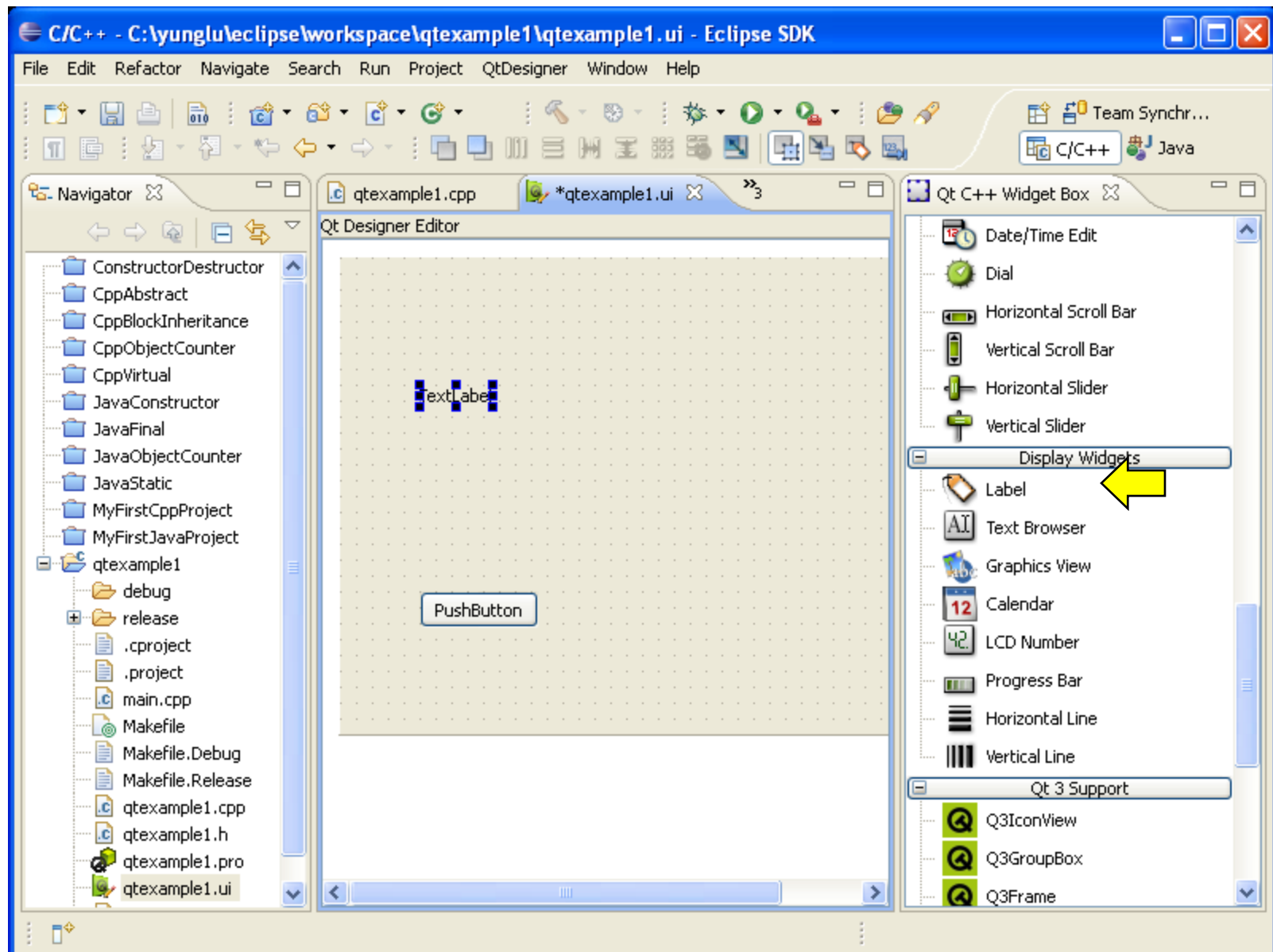
Qt Widget Box

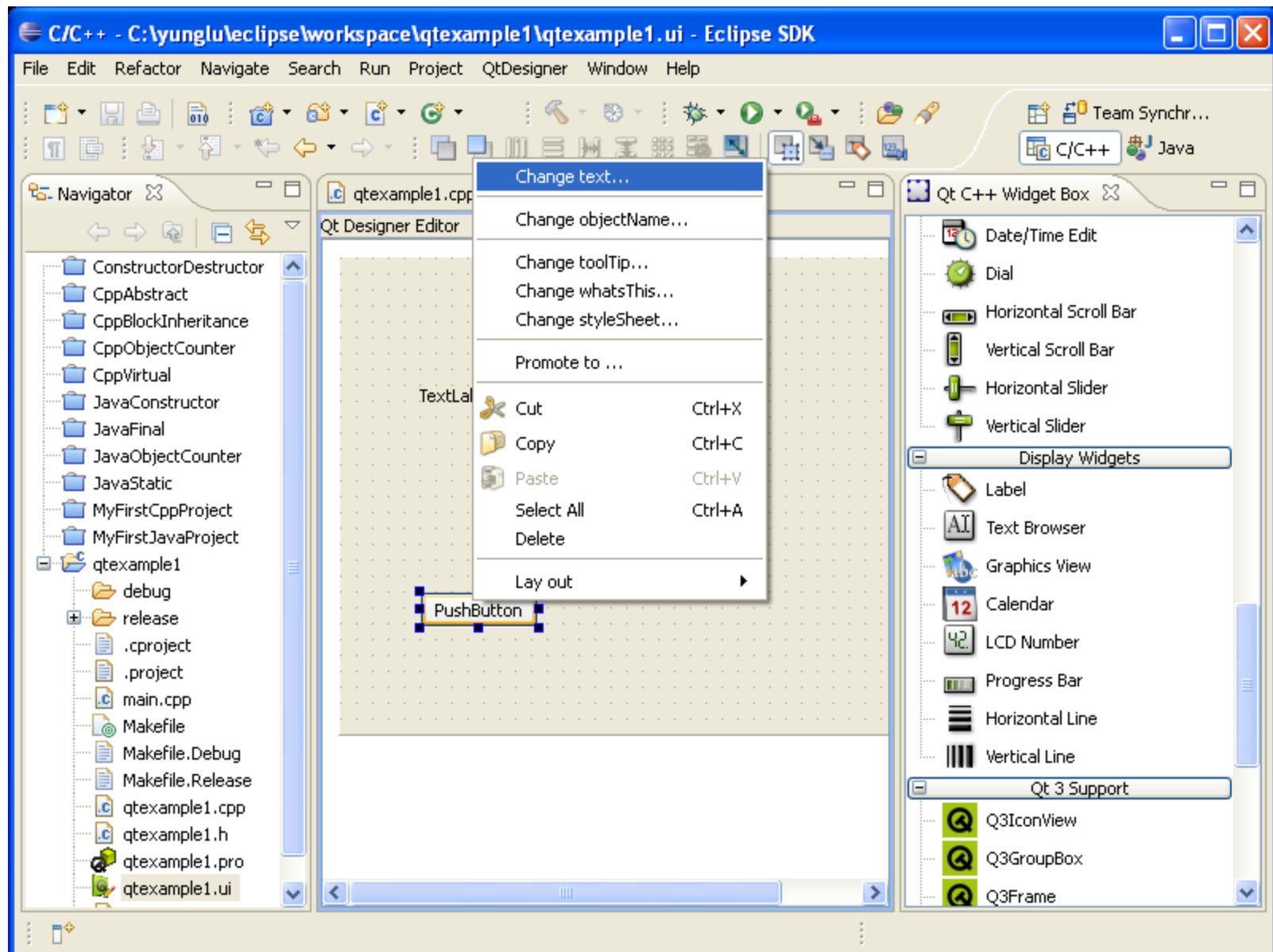


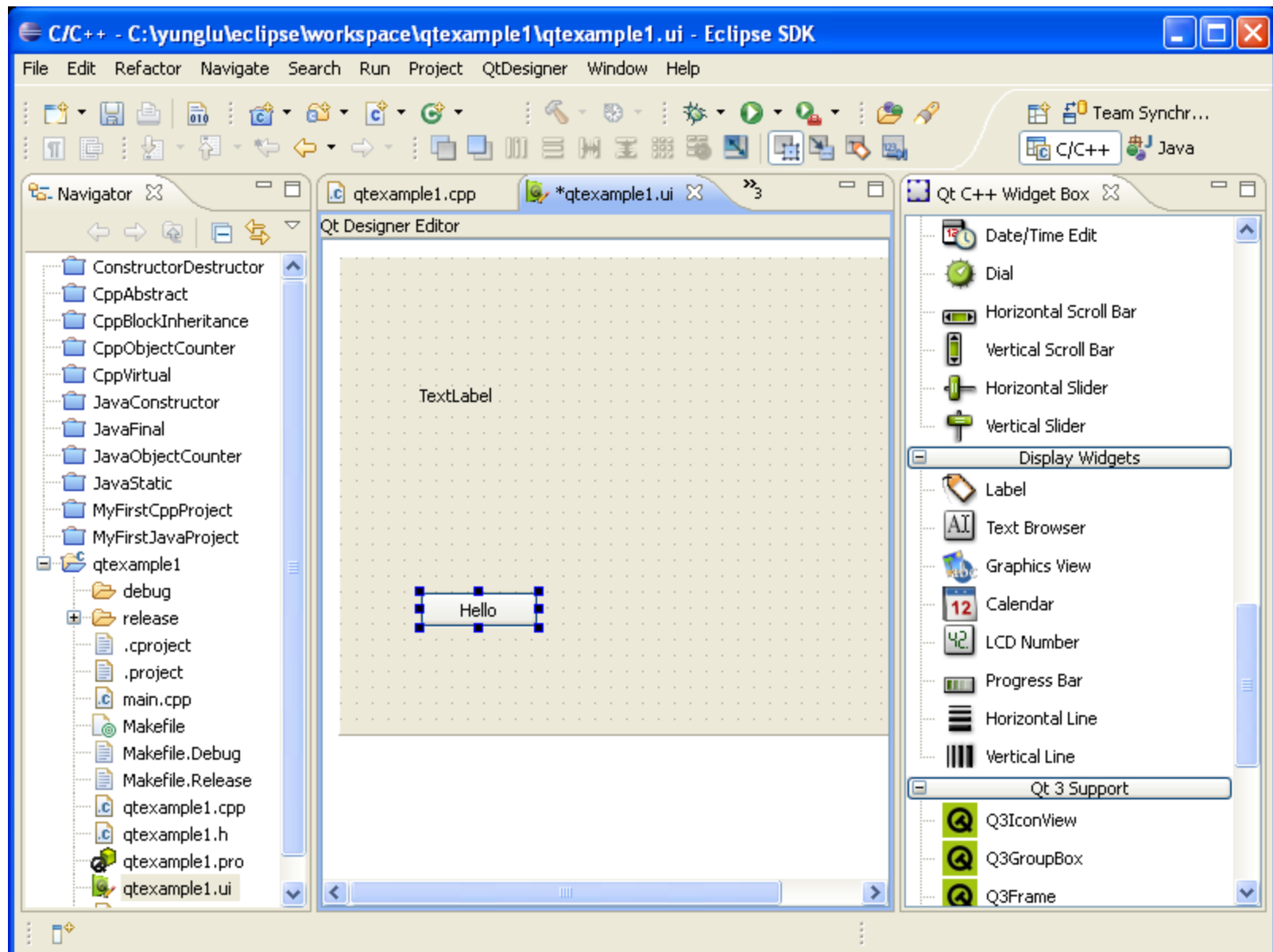


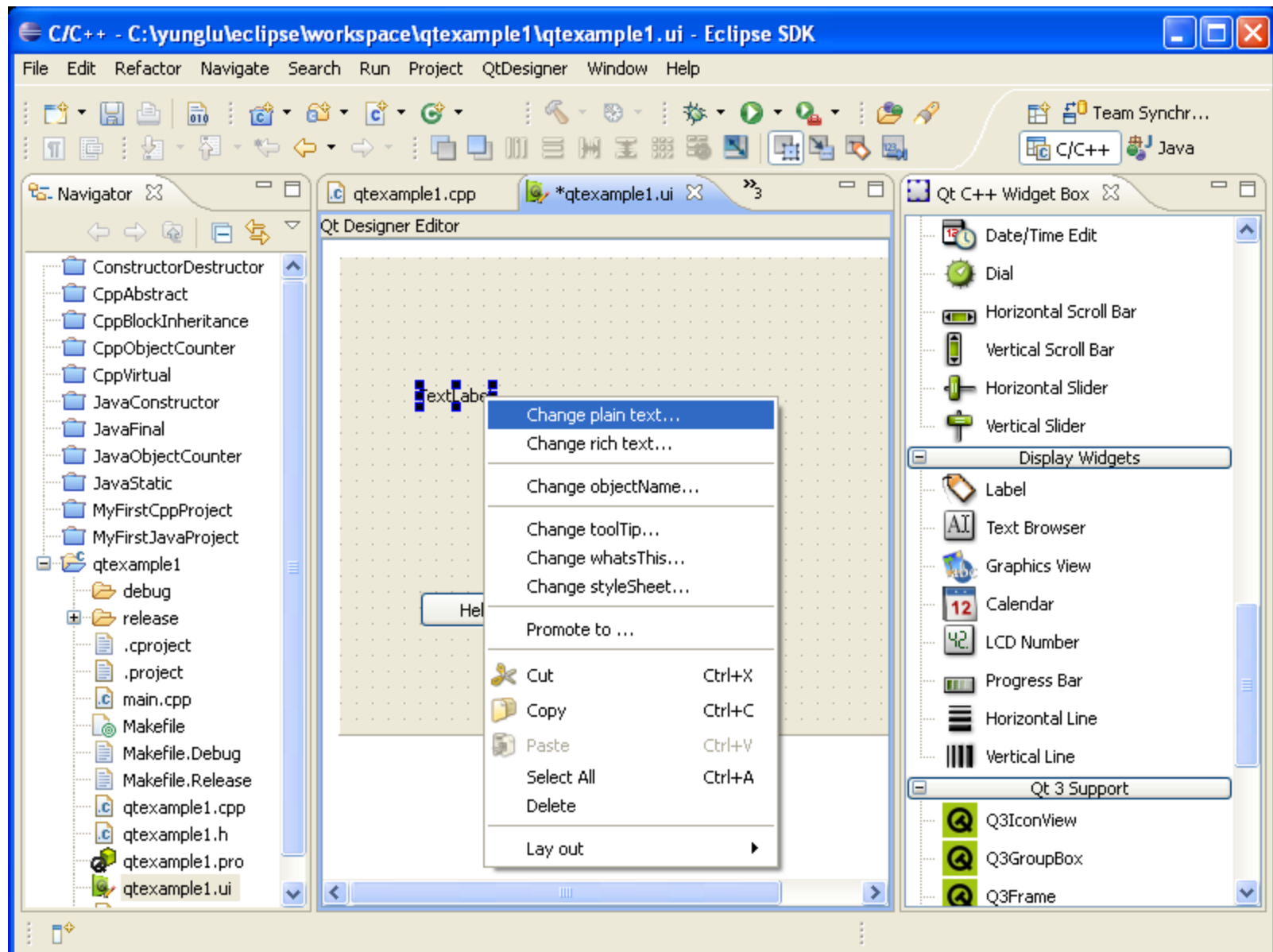


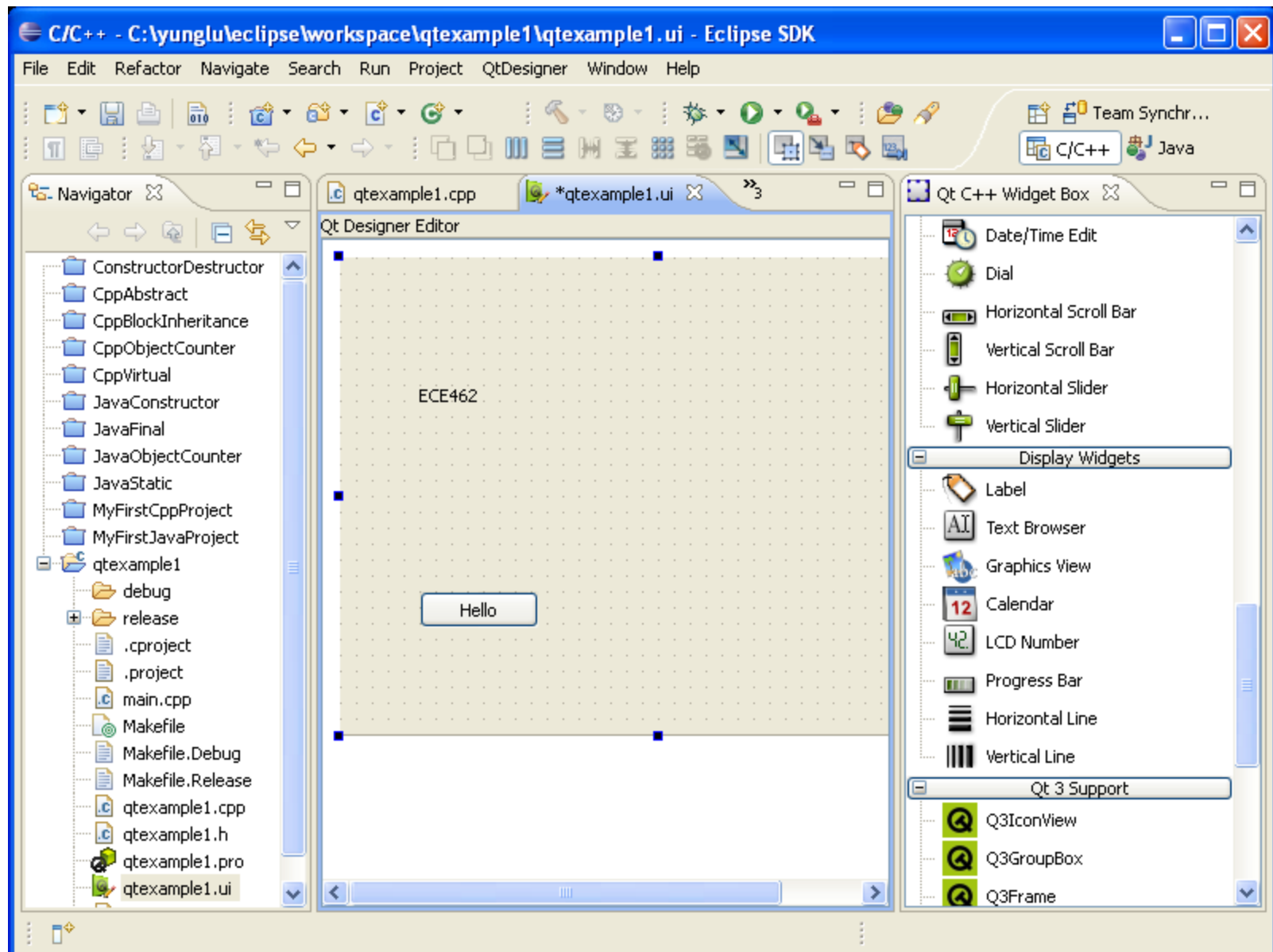


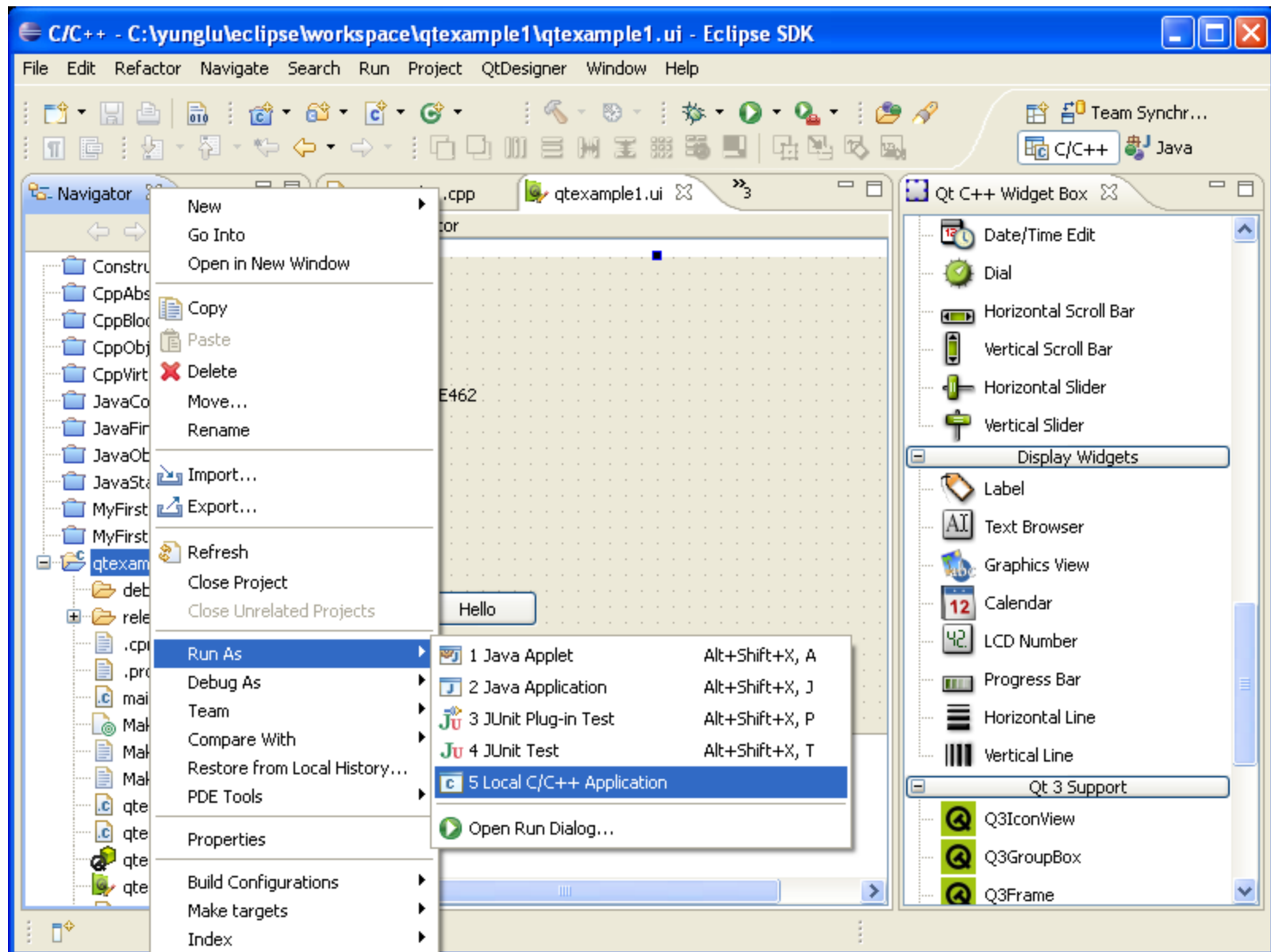


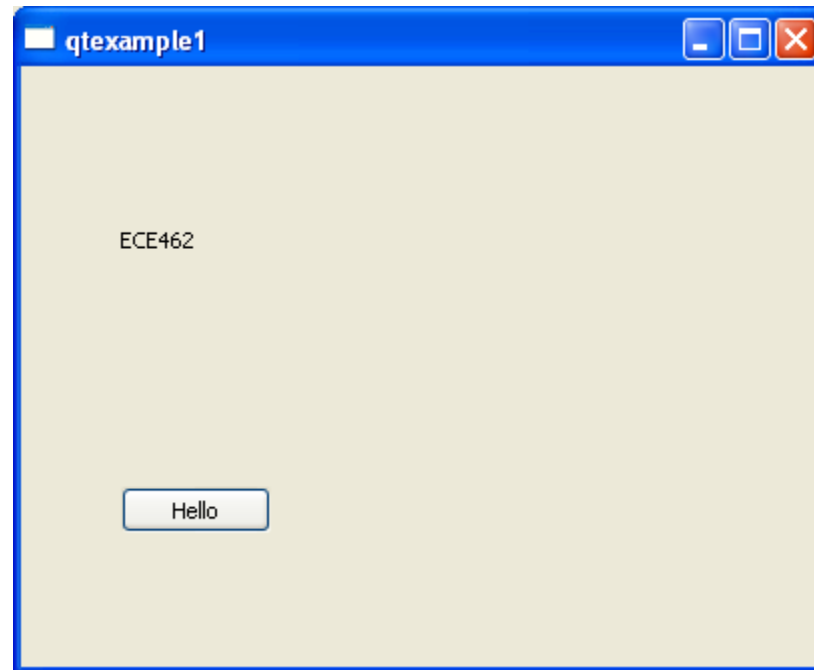












Signals and Slots

i.e. events and handlers



Qt 4.3: Signals and Slots - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/signalsandslots.html

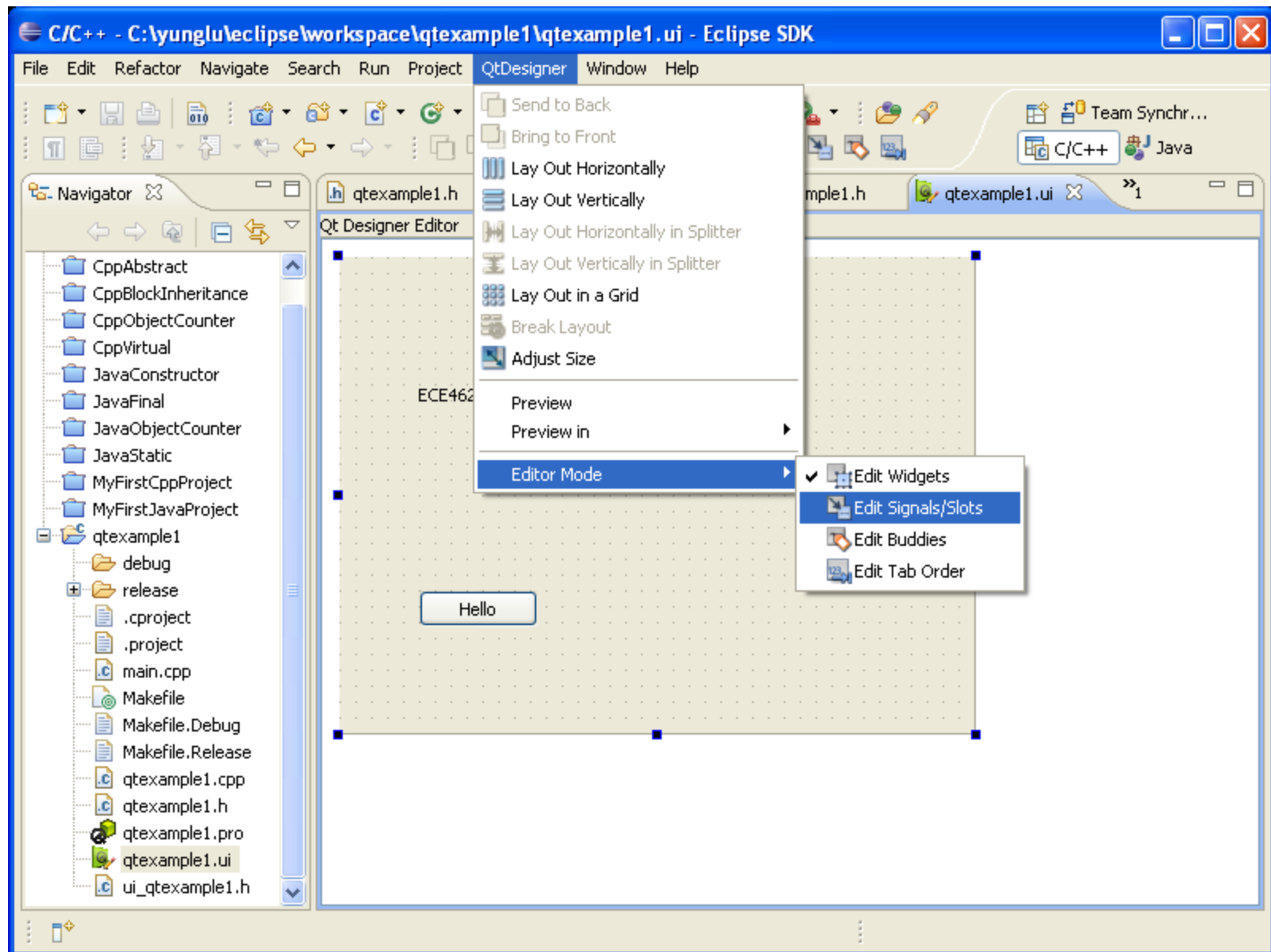
to subclass widgets and add your own slots so that you can handle the signals that you are interested in.

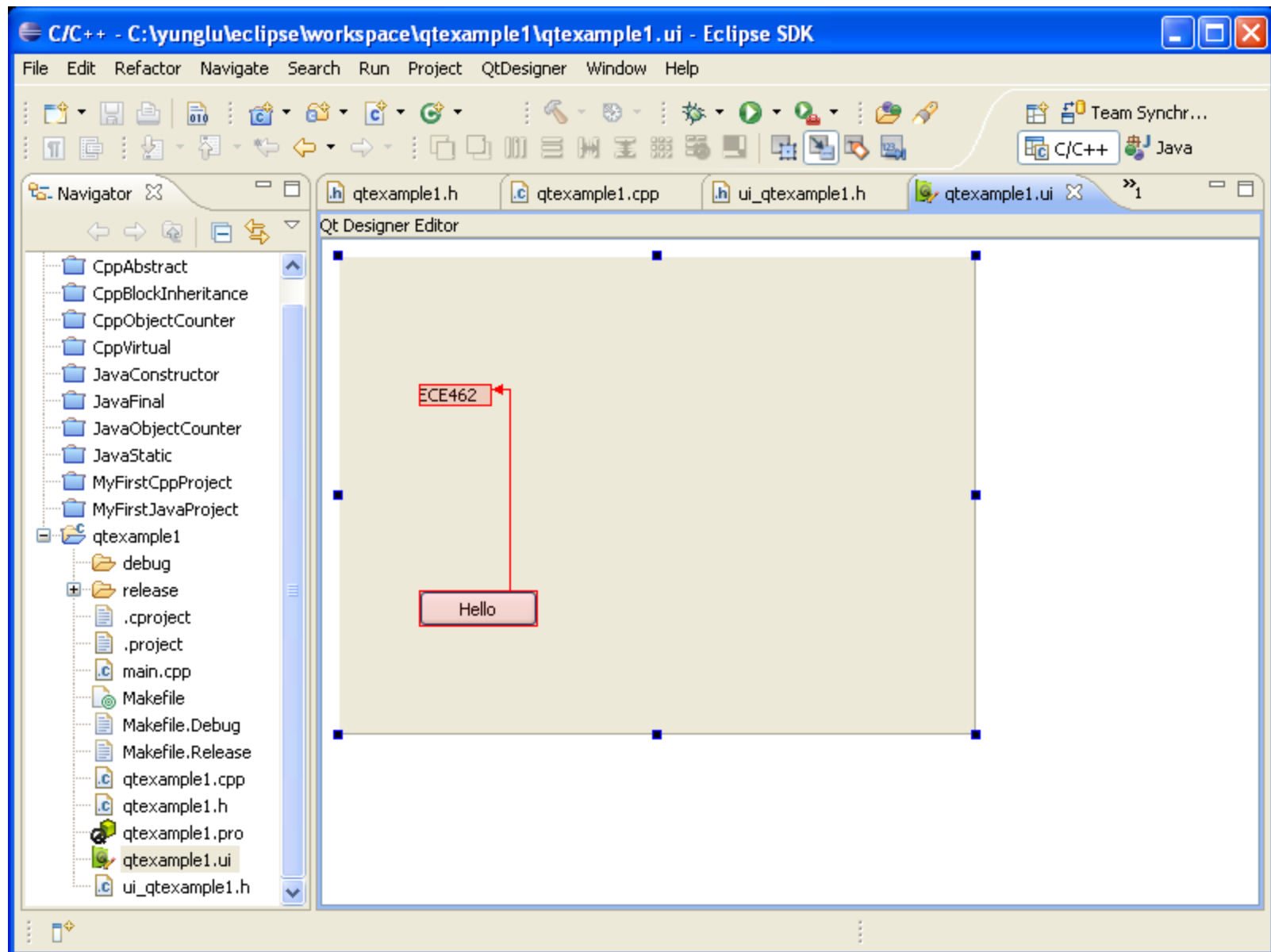
```
graph LR
    subgraph Object1
        direction TB
        O1_top[Object1]
        O1_sig1[signal1]
        O1_sig2[signal2]
    end
    subgraph Object2
        direction TB
        O2_top[Object2]
        O2_sig1[signal1]
        O2_slot1[slot1]
        O2_slot2[slot2]
    end
    subgraph Object3
        direction TB
        O3_top[Object3]
        O3_sig1[signal1]
        O3_slot1[slot1]
    end
    subgraph Object4
        direction TB
        O4_top[Object4]
        O4_slot1[slot1]
    end
    O1_sig1 --> O2_slot1
    O1_sig1 --> O2_slot2
    O1_sig2 --> O4_slot1
    O3_sig1 --> O4_slot1
```

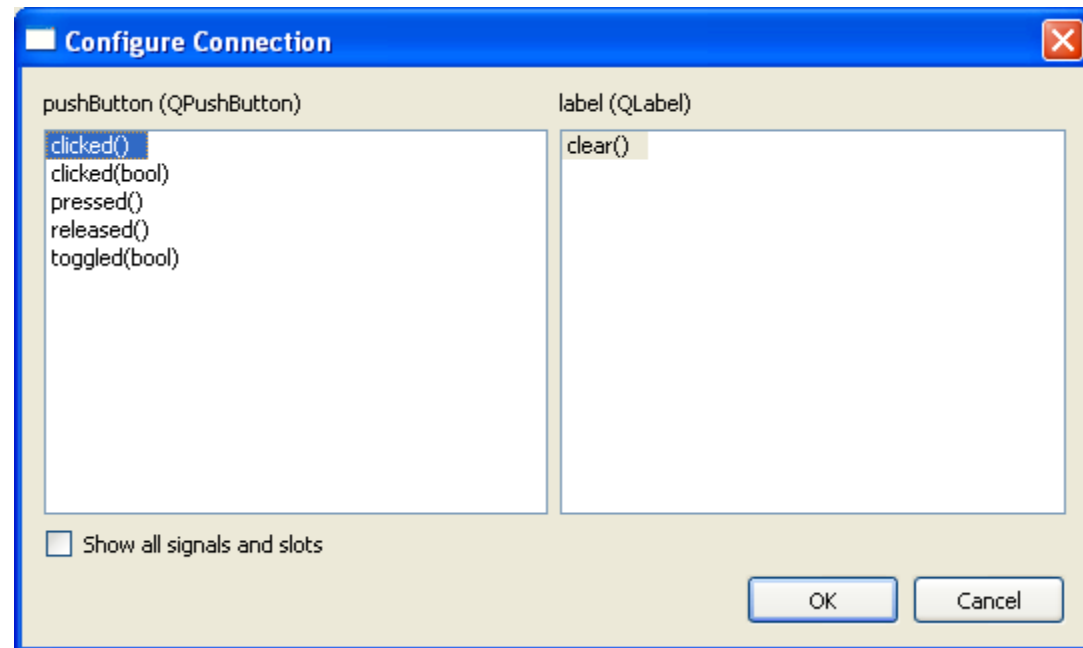
connect(Object1, signal1, Object2, slot1)
connect(Object1, signal1, Object2, slot2)

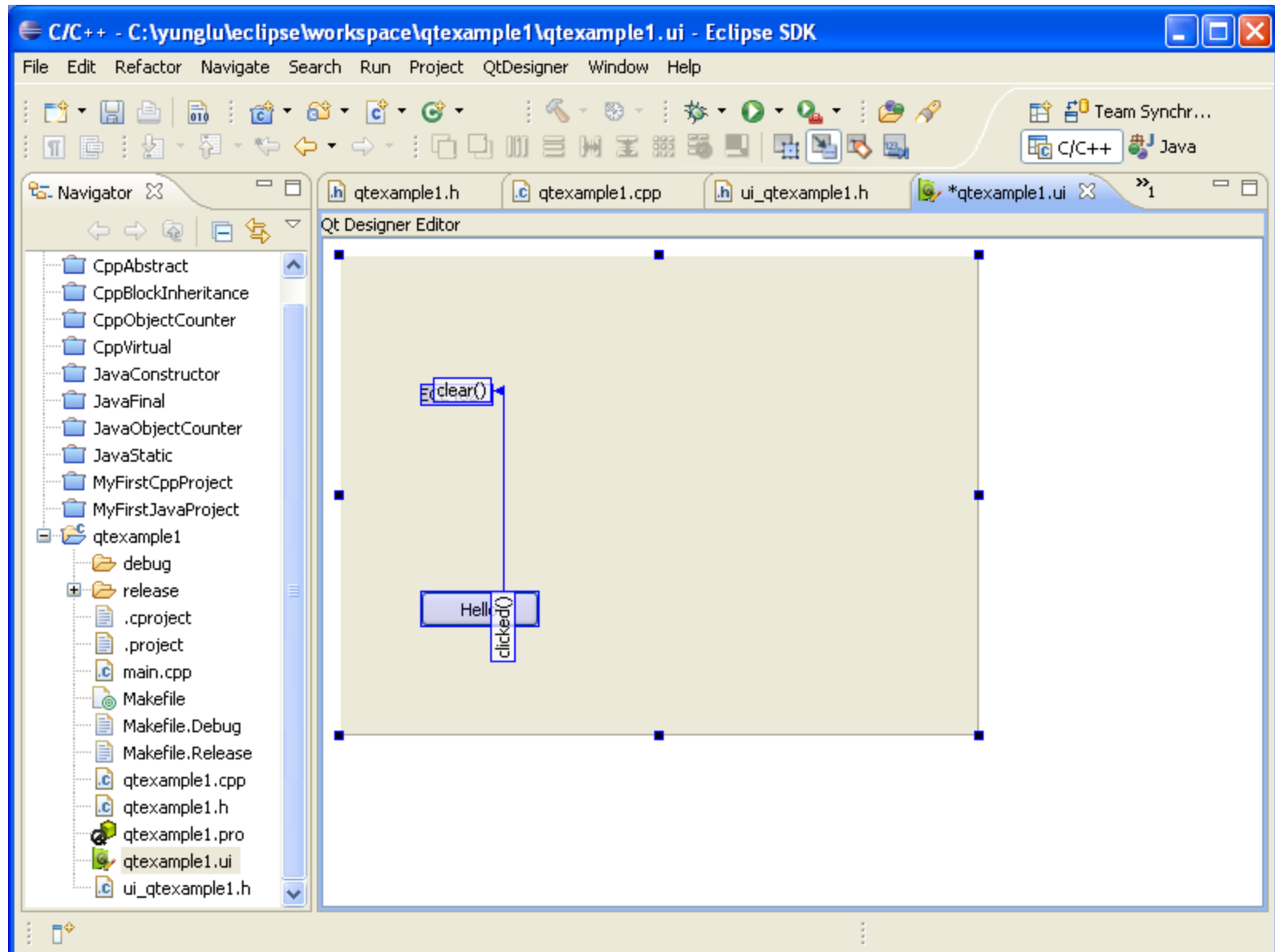
connect(Object1, signal2, Object4, slot1)

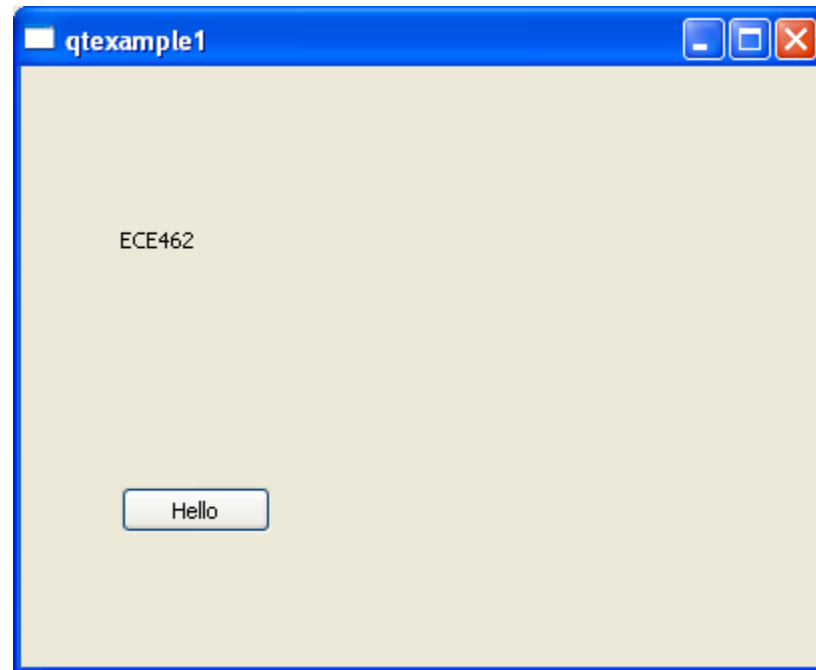
Done

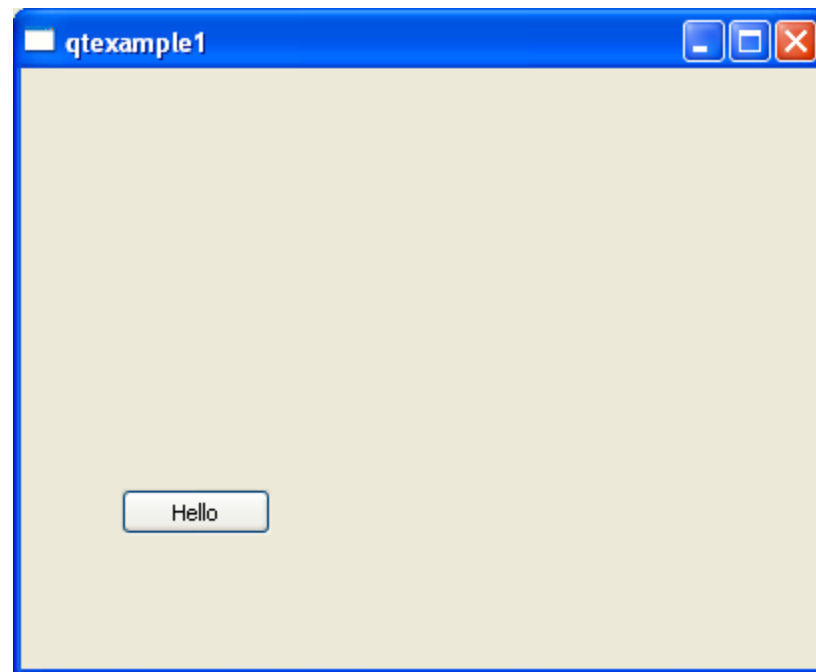


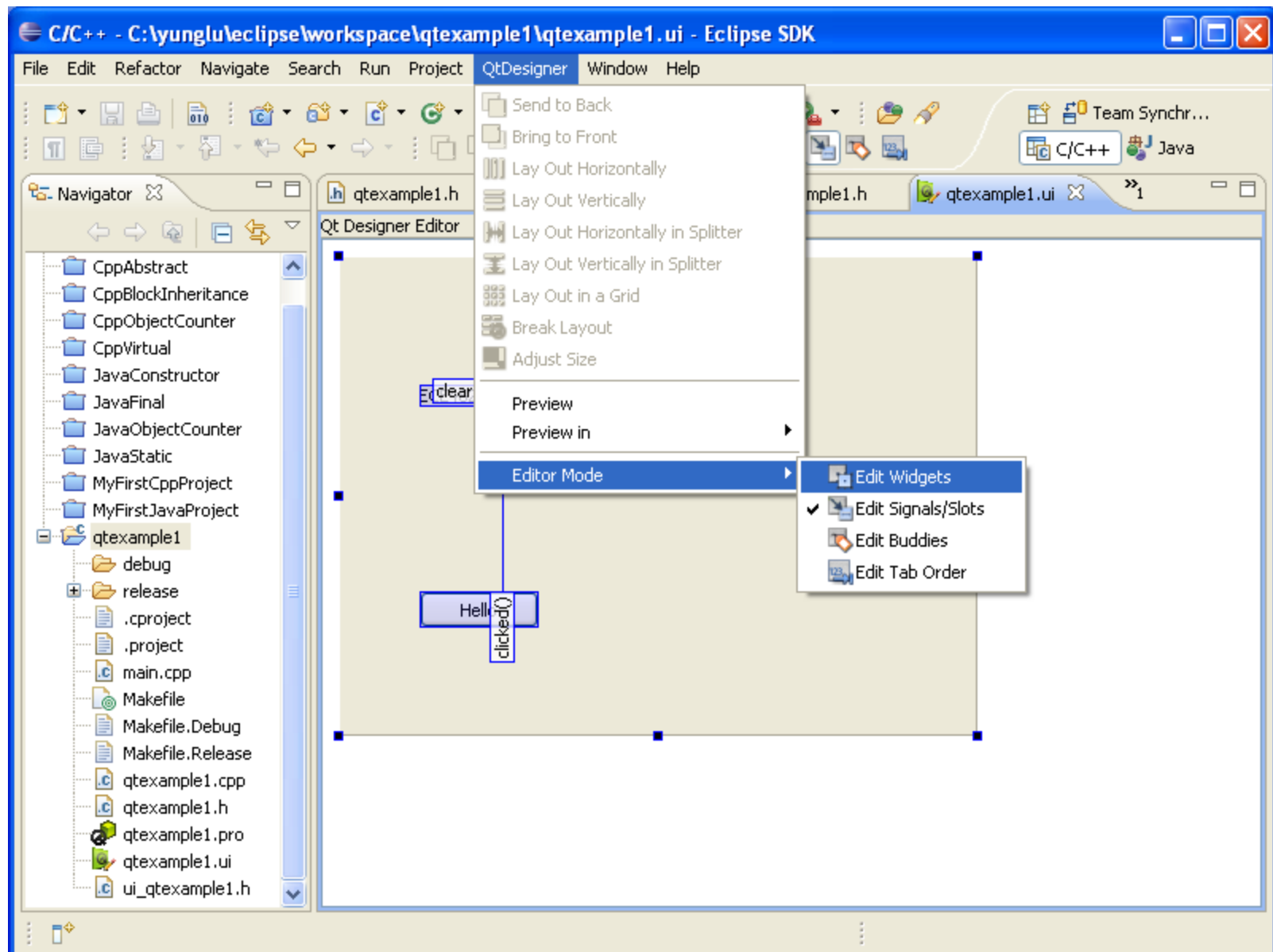


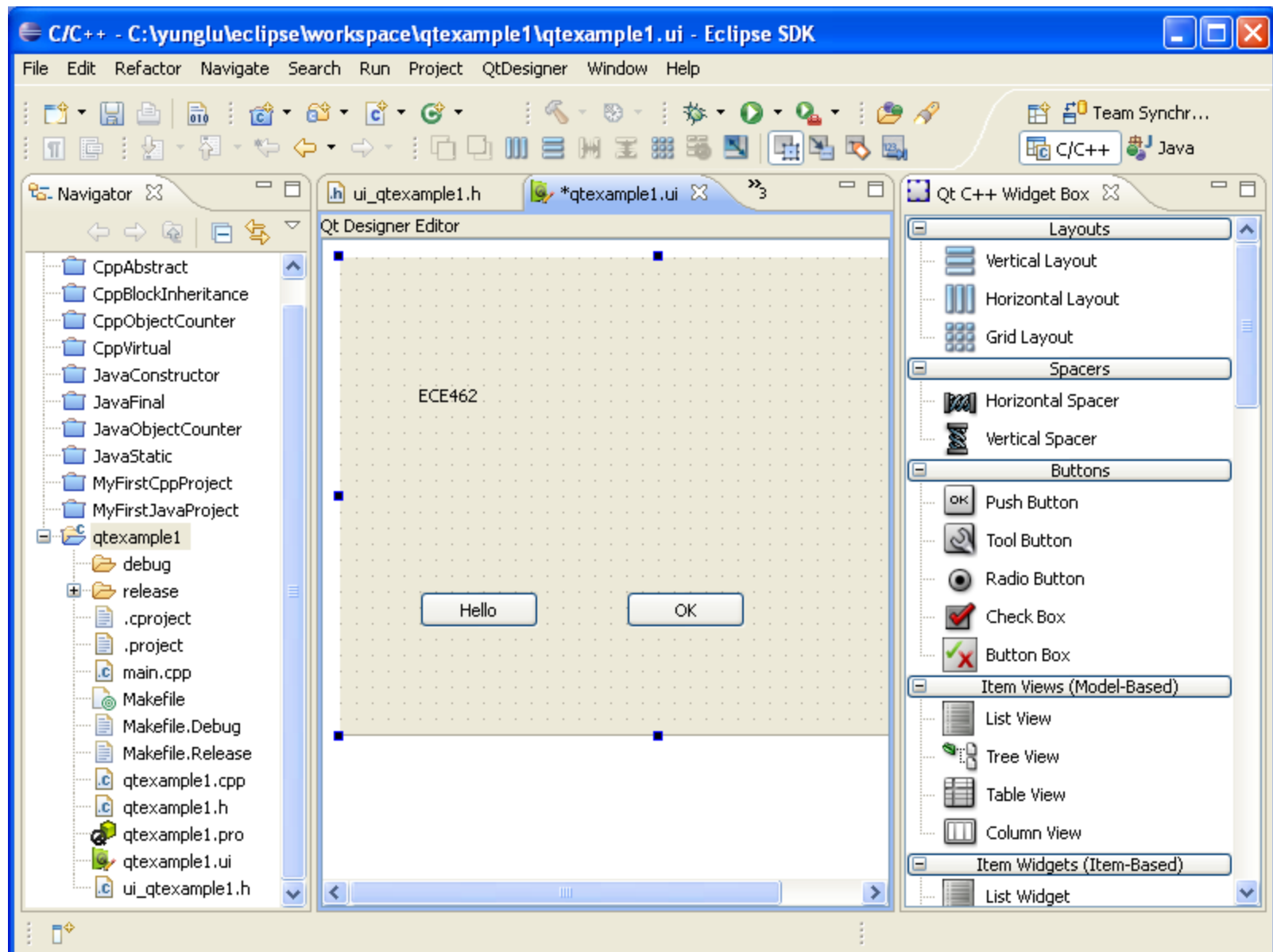


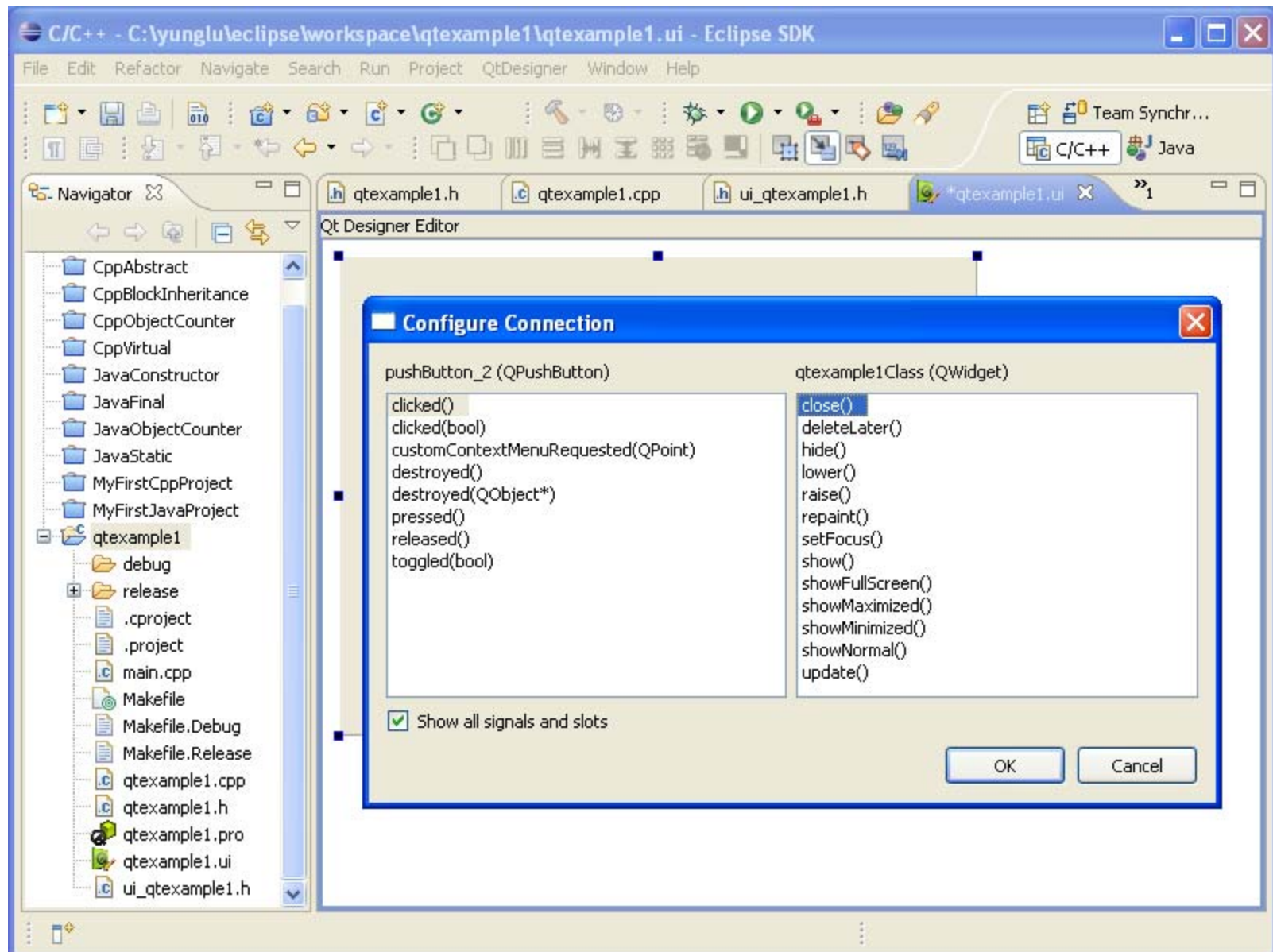


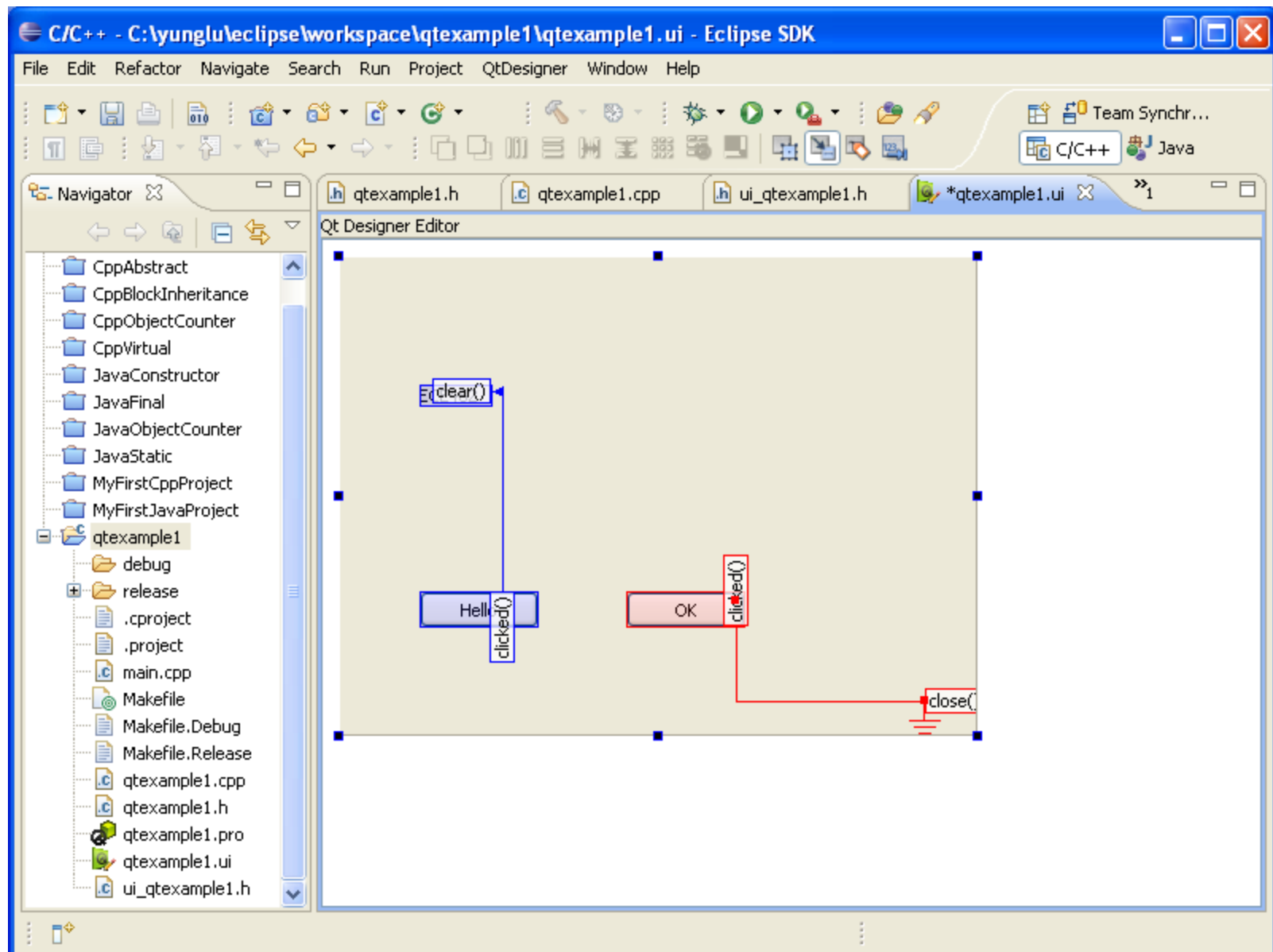


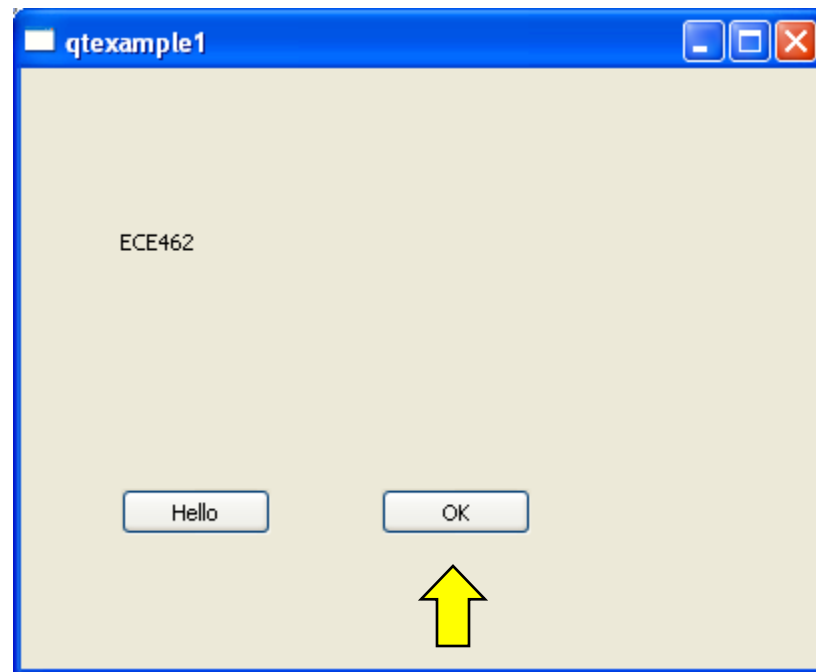


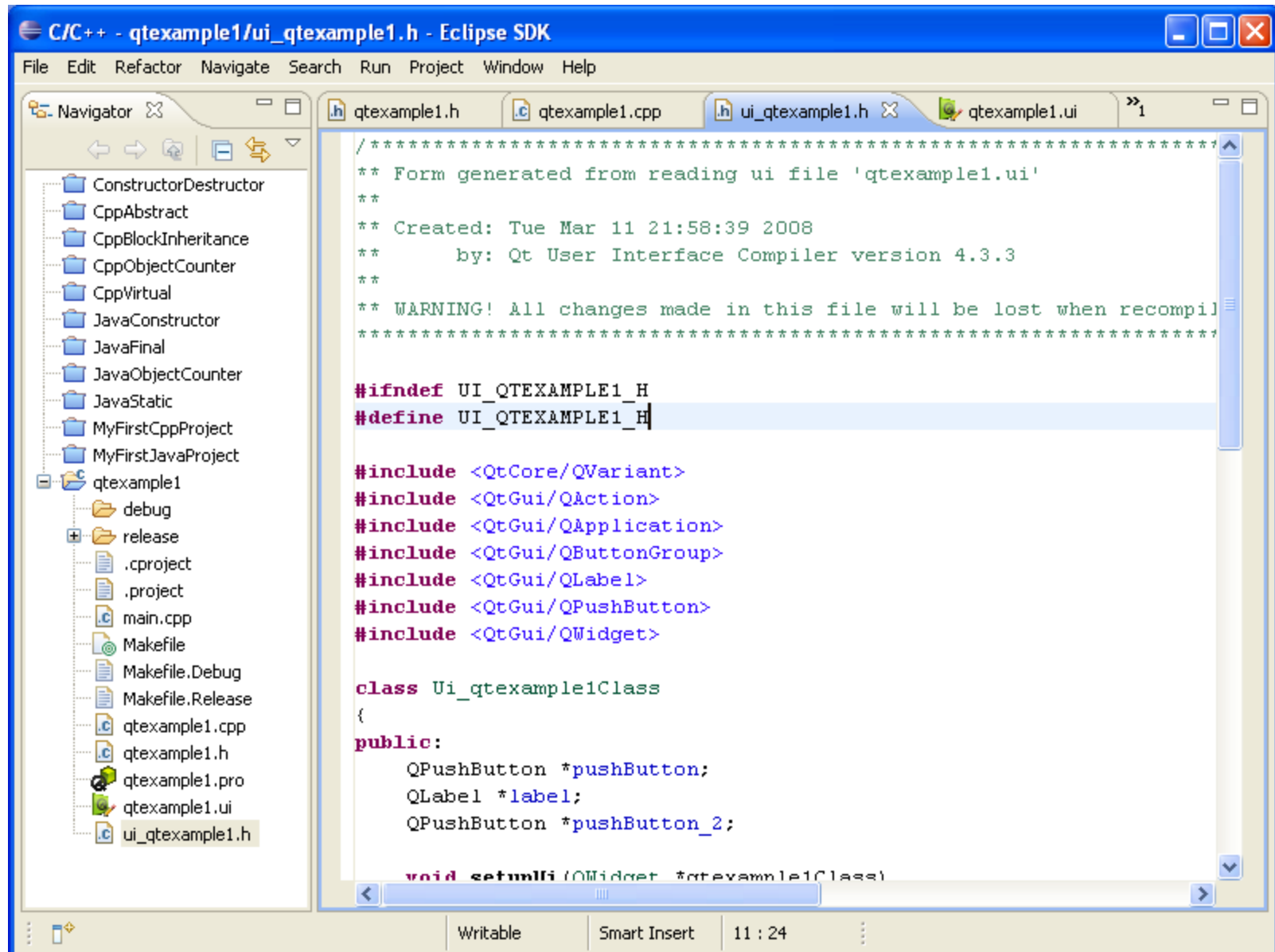


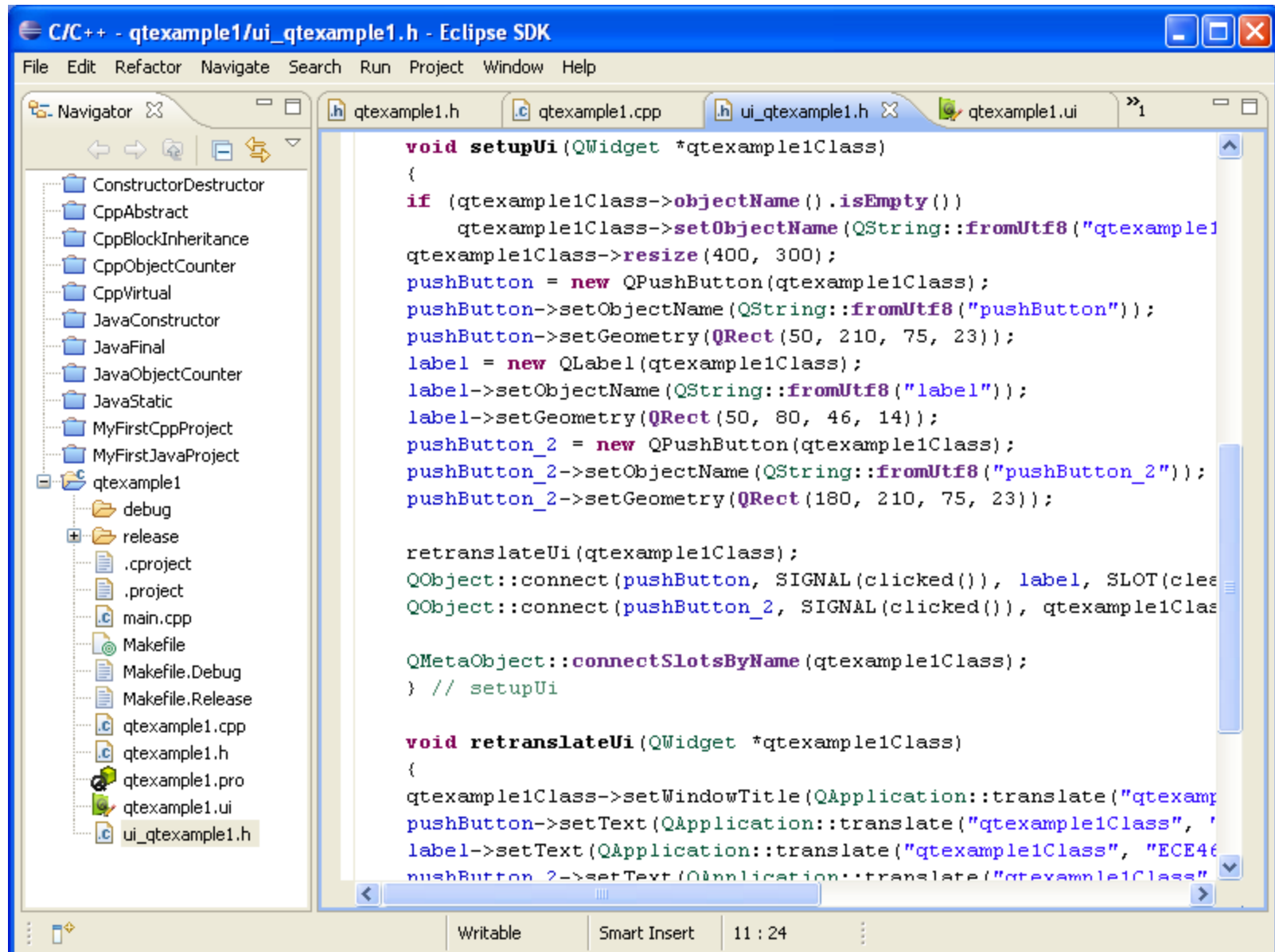














Qt 4.3: QPushButton Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qpushbutton.html

Google

1 property inherited from [QObject](#)

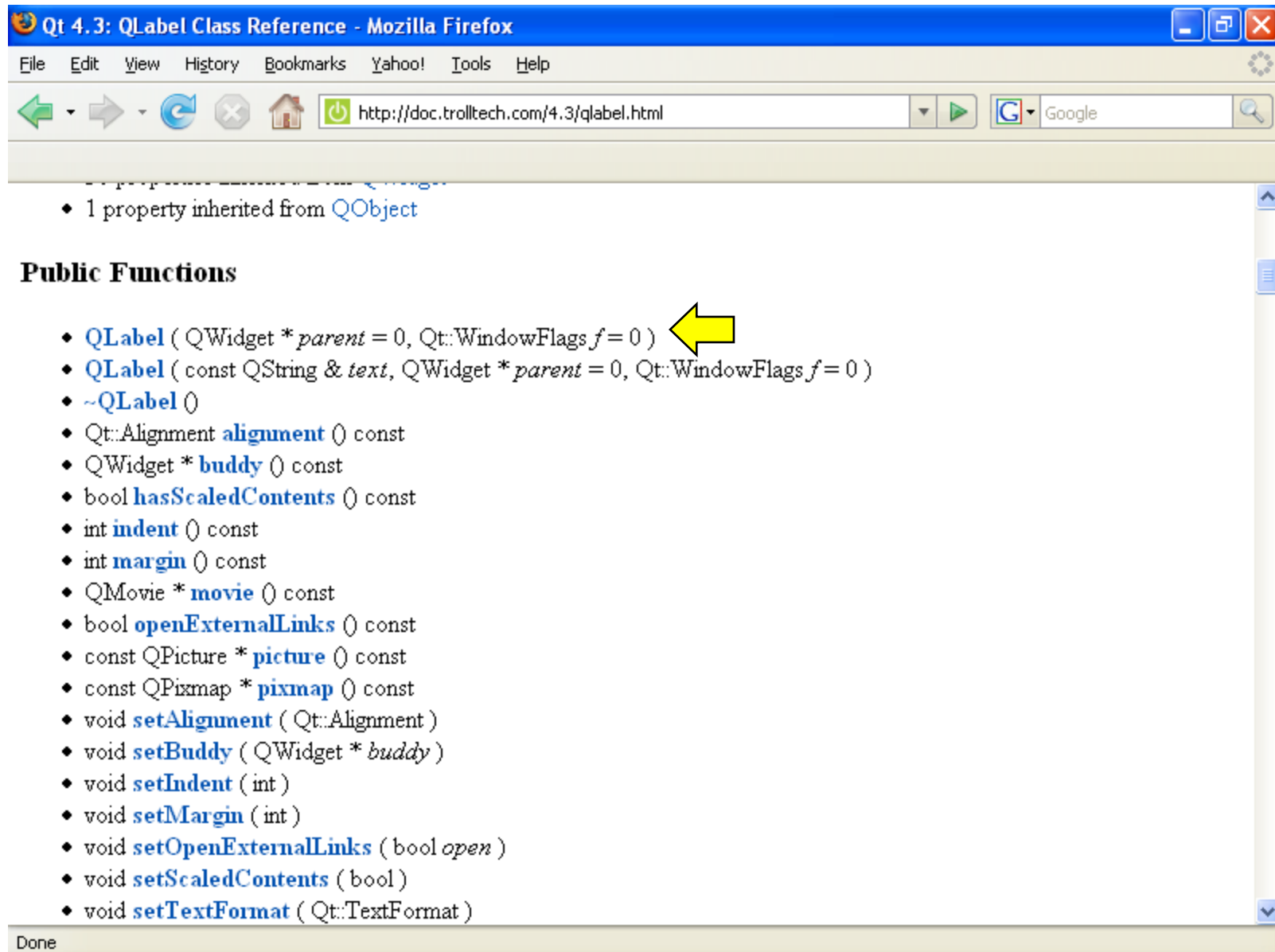
Public Functions

- [QPushButton](#) ([QWidget](#) *parent = 0)
- [QPushButton](#) (const [QString](#) &text, [QWidget](#) *parent = 0)
- [QPushButton](#) (const [QIcon](#) &icon, const [QString](#) &text, [QWidget](#) *parent = 0)
- [~QPushButton](#) ()
- bool [autoDefault](#) () const
- bool [isDefault](#) () const
- bool [isFlat](#) () const
- [QMenu](#) * [menu](#) () const
- void [setAutoDefault](#) (bool)
- void [setDefault](#) (bool)
- void [setFlat](#) (bool)
- void [setMenu](#) ([QMenu](#) *menu)

- 21 public functions inherited from [QAbstractButton](#)
- 201 public functions inherited from [QWidget](#)
- 29 public functions inherited from [QObject](#)
- 12 public functions inherited from [QPaintDevice](#)

Public Slots

Done



Default Parameter Values

Default Parameter Values in C++

- always start from the last parameter

`func1(int x, int y, int z = 0) ...`

`func1(int x, int y = 1, int z = 0) ...`

`func1(int x = 2, int y = 1, int z = 0) ...`

$\Rightarrow$ `func1(1,2,3)` means $x = 1, y = 2, z = 3$

$\Rightarrow$ `func1(4, 5)` means $x = 4, y = 5, z = 0$

$\Rightarrow$ `func1(6)` means $x = 6, y = 1, z = 0$

$\Rightarrow$ `func1()` means $x = 2, y = 1, z = 0$

- cannot skip any parameter (otherwise, ambiguous)

`func1(int x = 2, int y, int z = 0) ...`

$\Rightarrow$ what does `func1(3, 4)` mean?

Graphical User Interface (GUI)

- Most GUIs are built using OOP because OOP is a natural way to support GUI.
- GUI has "containers", such as QWidget, that can include buttons, labels, scrollbars ...
- When the widget is redrawn (restore from minimized, maximized, or content update), the frame asks each component to redraw itself.
- Each object knows how to redraw itself because `paintEvent(C++ / Qt)` or `paintComponent (Java)` is overridden.

GUI using Java

JFrame (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://java.sun.com/javase/6/docs/api/javax/swing/JFrame.html

Overview Package **Class** Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Java™ Platform
Standard Ed. 6

javax.swing

Class JFrame

[java.lang.Object](#)

- [java.awt.Component](#)
 - [java.awt.Container](#)
 - [java.awt.Window](#)
 - [java.awt.Frame](#)
 - [javax.swing.JFrame](#)

← JFrame is a derived class of Container

All Implemented Interfaces:
[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#), [RootPaneContainer](#), [WindowConstants](#)

```
public class JFrame
extends Frame
implements WindowConstants, Accessible, RootPaneContainer
```

An extended version of `java.awt.Frame` that adds support for the JFC/Swing component architecture. You can find task-oriented documentation about using `JFrame` in *The Java Tutorial*, in the section [How to Make Frames](#).

The `JFrame` class is slightly incompatible with `Frame`. Like all other JFC/Swing top-level containers, a `JFrame` contains a `JRootPane` as

Done

JBButton (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://java.sun.com/javase/6/docs/api/javax/swing/JButton.html

Overview Package **Class** Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

FRAMES NO FRAMES All Classes

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Java™ Platform
Standard Ed. 6

javax.swing

Class JButton

[java.lang.Object](#)

- [java.awt.Component](#)
 - [java.awt.Container](#)
 - [javax.swing.JComponent](#)
 - [javax.swing.AbstractButton](#)
 - javax.swing.JButton**

All Implemented Interfaces:
[ImageObserver](#), [ItemSelectable](#), [MenuContainer](#), [Serializable](#), [Accessible](#), [SwingConstants](#)

Direct Known Subclasses:
[BasicArrowButton](#), [MetalComboBoxButton](#)

```
public class JButton
extends AbstractButton
implements Accessible
```

An implementation of a "push" button.

Done

JComponent (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://java.sun.com/javase/6/docs/api/javax/swing/JComponent.html

Overview Package **Class** Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

FRAMES NO FRAMES [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Java™ Platform
Standard Ed. 6

javax.swing

Class JComponent

[java.lang.Object](#)

- [java.awt.Component](#)
 - [java.awt.Container](#)
 - [javax.swing.JComponent](#)

All Implemented Interfaces:
[ImageObserver](#), [MenuContainer](#), [Serializable](#)

Direct Known Subclasses:
[AbstractButton](#), [BasicInternalFrameTitlePane](#), [Box](#), [Box.Filler](#), [JColorChooser](#), [JComboBox](#), [JFileChooser](#), [JInternalFrame](#), [JInternalFrame.JDesktopIcon](#), [JLabel](#), [JLayeredPane](#), [JList](#), [JMenuBar](#), [JOptionPane](#), [JPanel](#), [JPopupMenu](#), [JProgressBar](#), [JRootPane](#), [JScrollBar](#), [JScrollPane](#), [JSeparator](#), [JSlider](#), [JSpinner](#), [JSplitPane](#), [JTabbedPane](#), [JTable](#), [JTableHeader](#), [JTextComponent](#), [JToolBar](#), [JToolTip](#), [JTree](#), [JViewport](#)

```
public abstract class JComponent
extends Container
implements Serializable
```

[http://java.sun.com/javase/6/docs/api/javax/swing/JComponent.html#paintComponent\(java.awt.Graphics\)](#)

many derived classes



JComponent (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://java.sun.com/javase/6/docs/api/javax/swing/JComponent.html

	operation.
boolean	isPaintingTile() Returns true if the component is currently painting a tile.
boolean	isRequestFocusEnabled() Returns true if this JComponent should get focus; otherwise returns false.
boolean	isValidateRoot() If this method returns true, revalidate calls by descendants of this component will cause the entire tree beginning with this root to be validated.
void	paint(Graphics g) Invoked by Swing to draw components.
protected void	paintBorder(Graphics g) Paints the comp
protected void	paintChildren(Graphics g) Paints this comp
protected void	paintComponent(Graphics g) Calls the UI delegate's paint method, if the UI delegate is non-null.
void	paintImmediately(int x, int y, int w, int h) Paints the specified region in this component and all of its descendants that overlap the region, immediately.
void	paintImmediately(Rectangle r) Paints the specified region now.
protected String	 paramString()

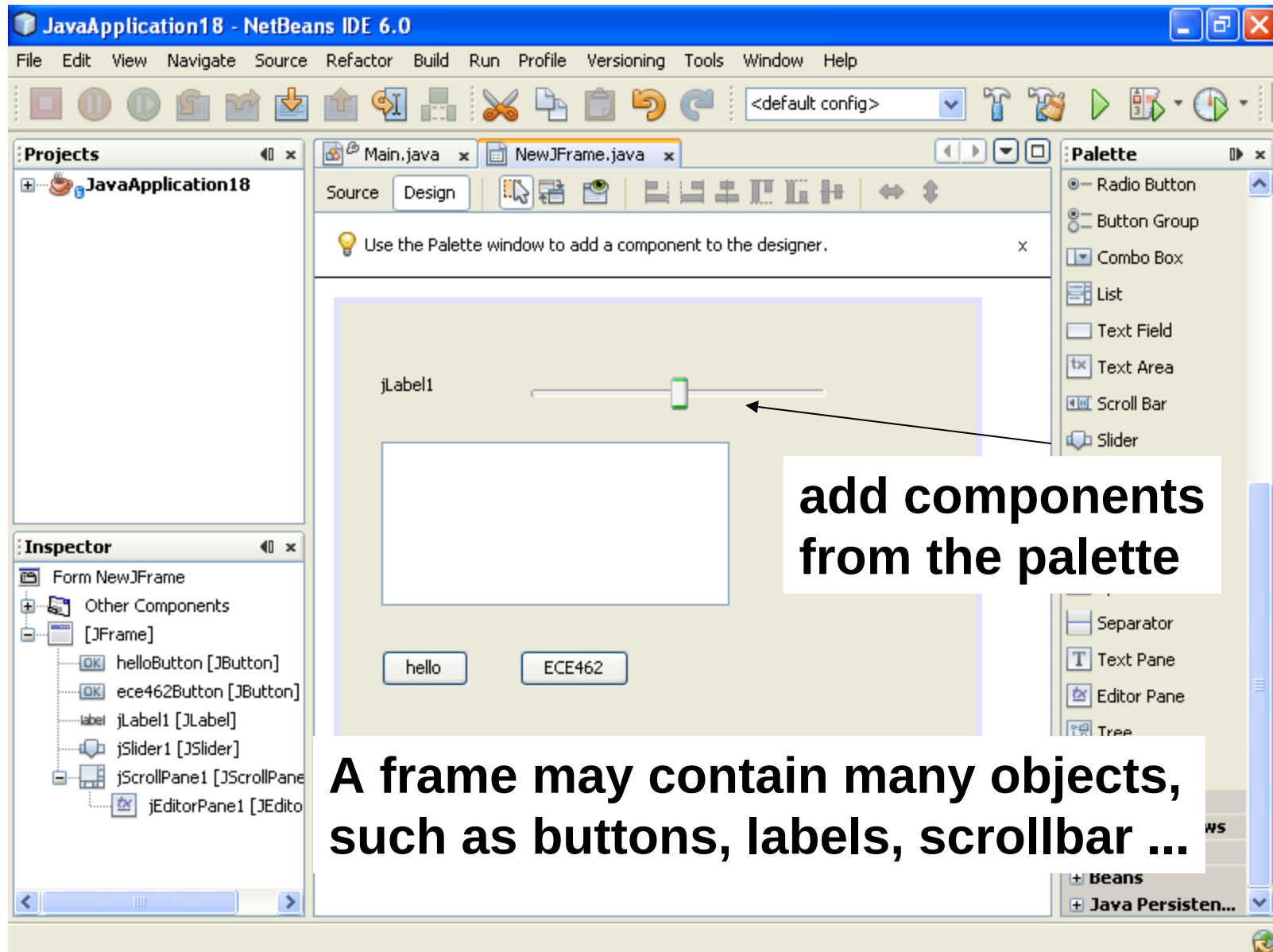
Done

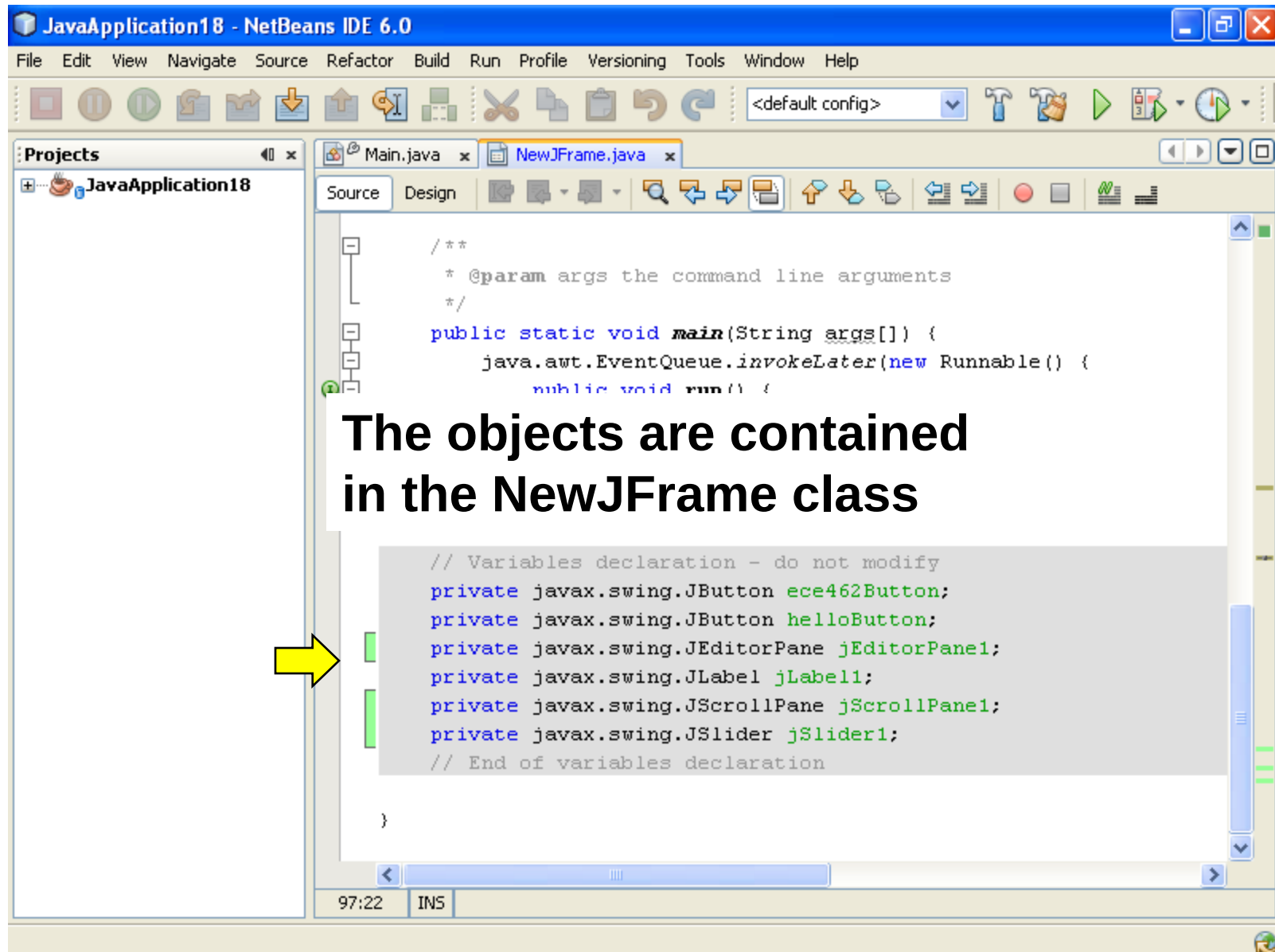
**Derived classes
override paintComponent**

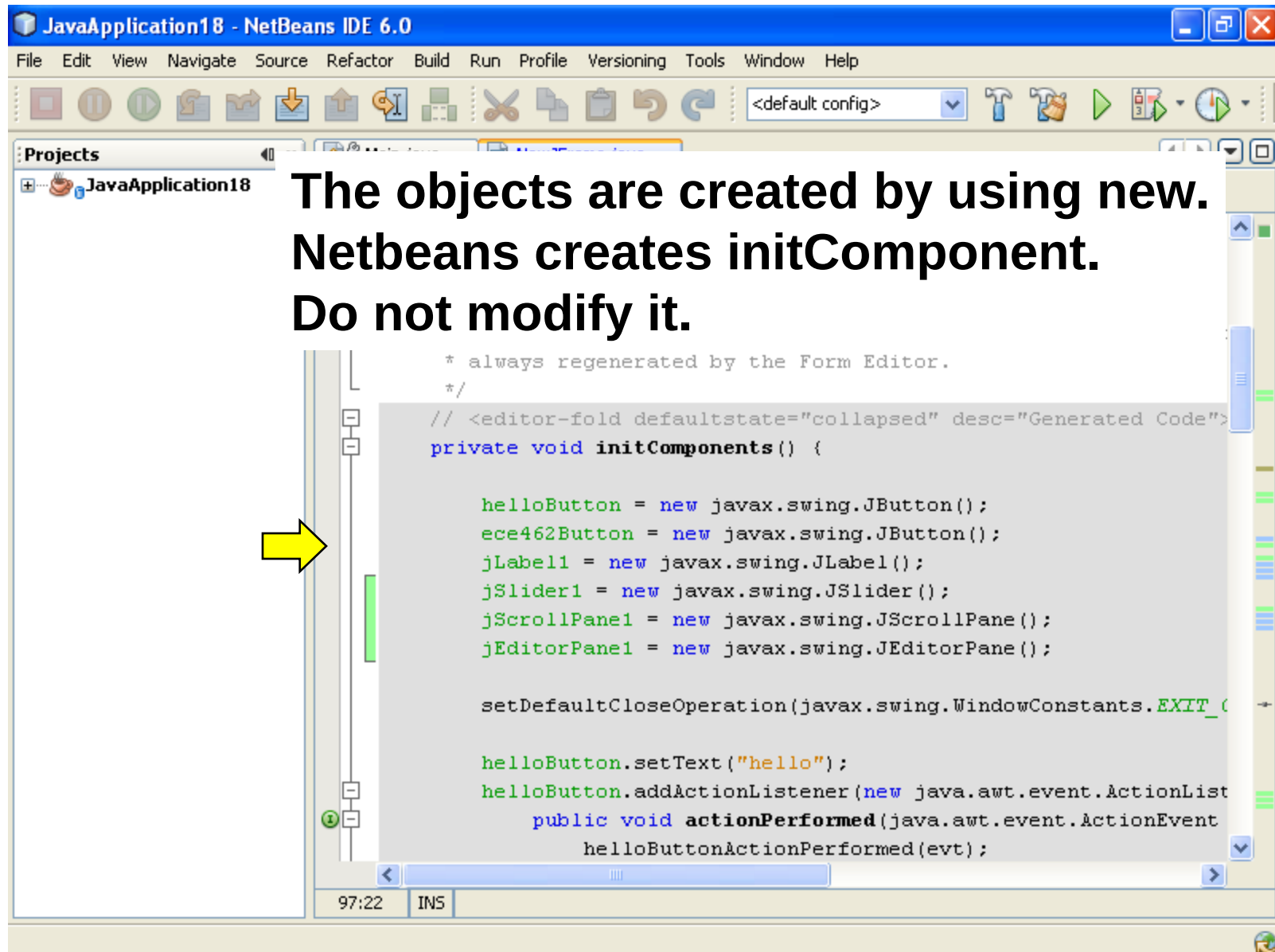


JFrame

similar to QWidget







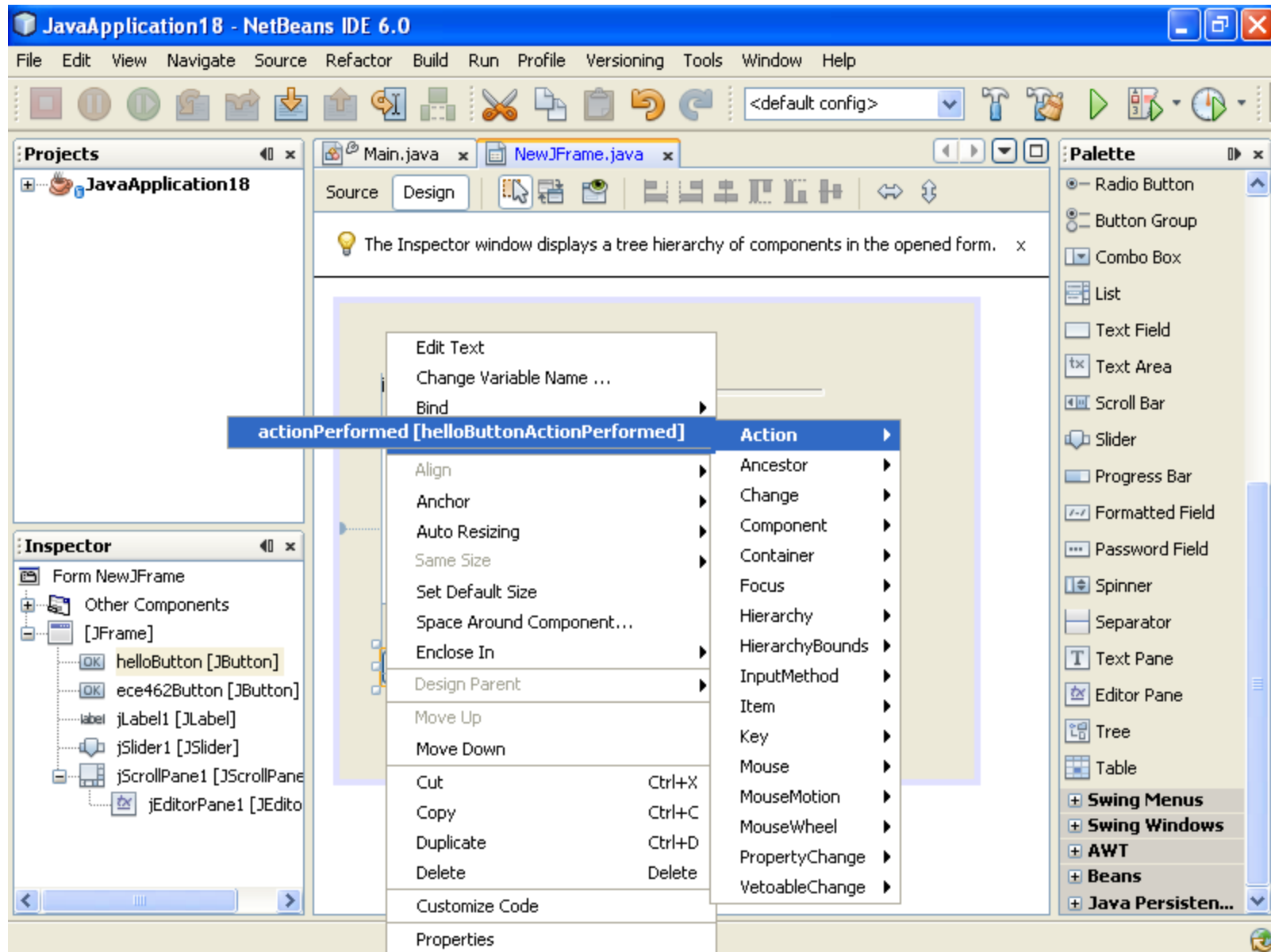
Handle User Actions

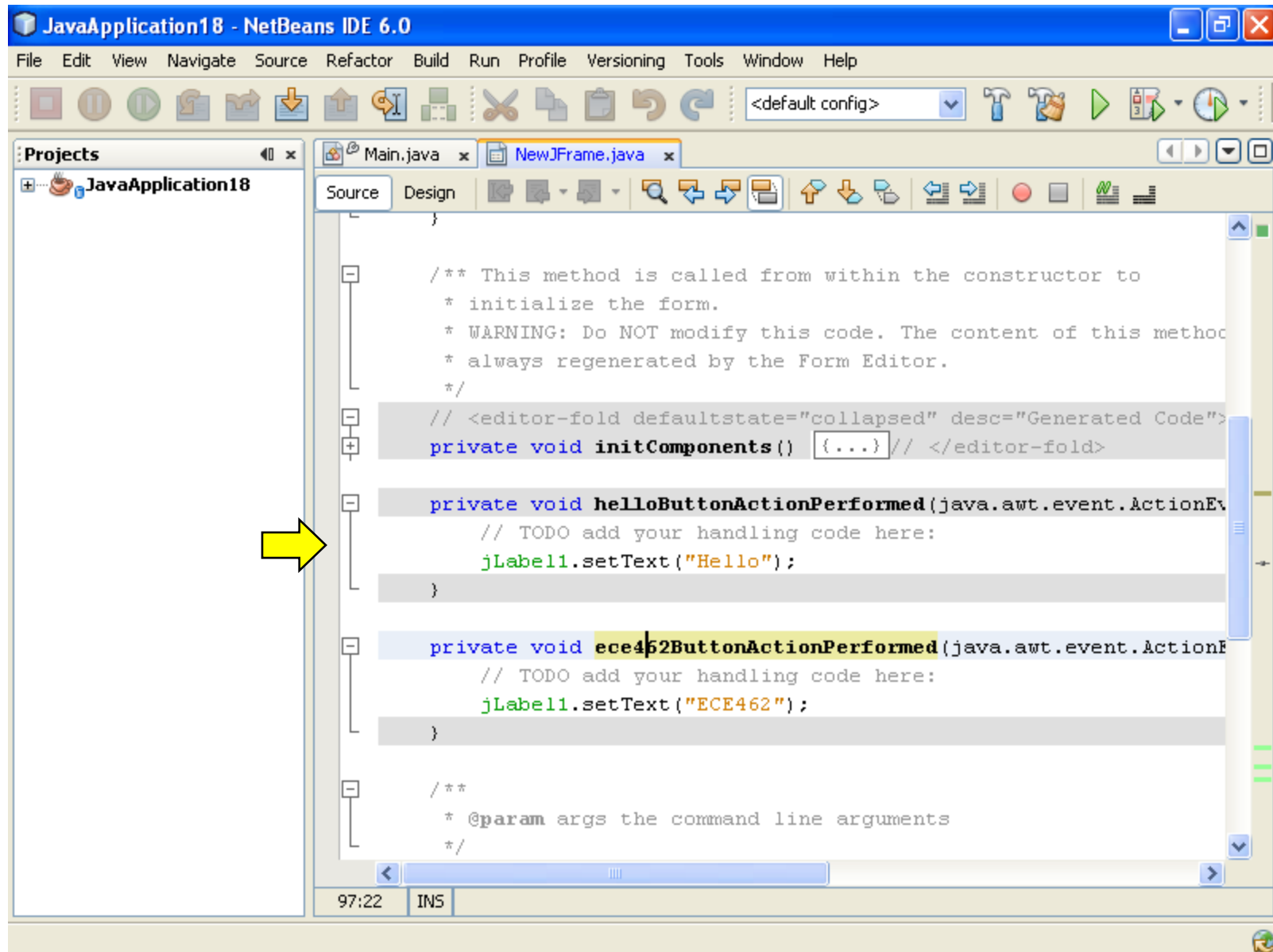
Handle Button Click

Events

Action

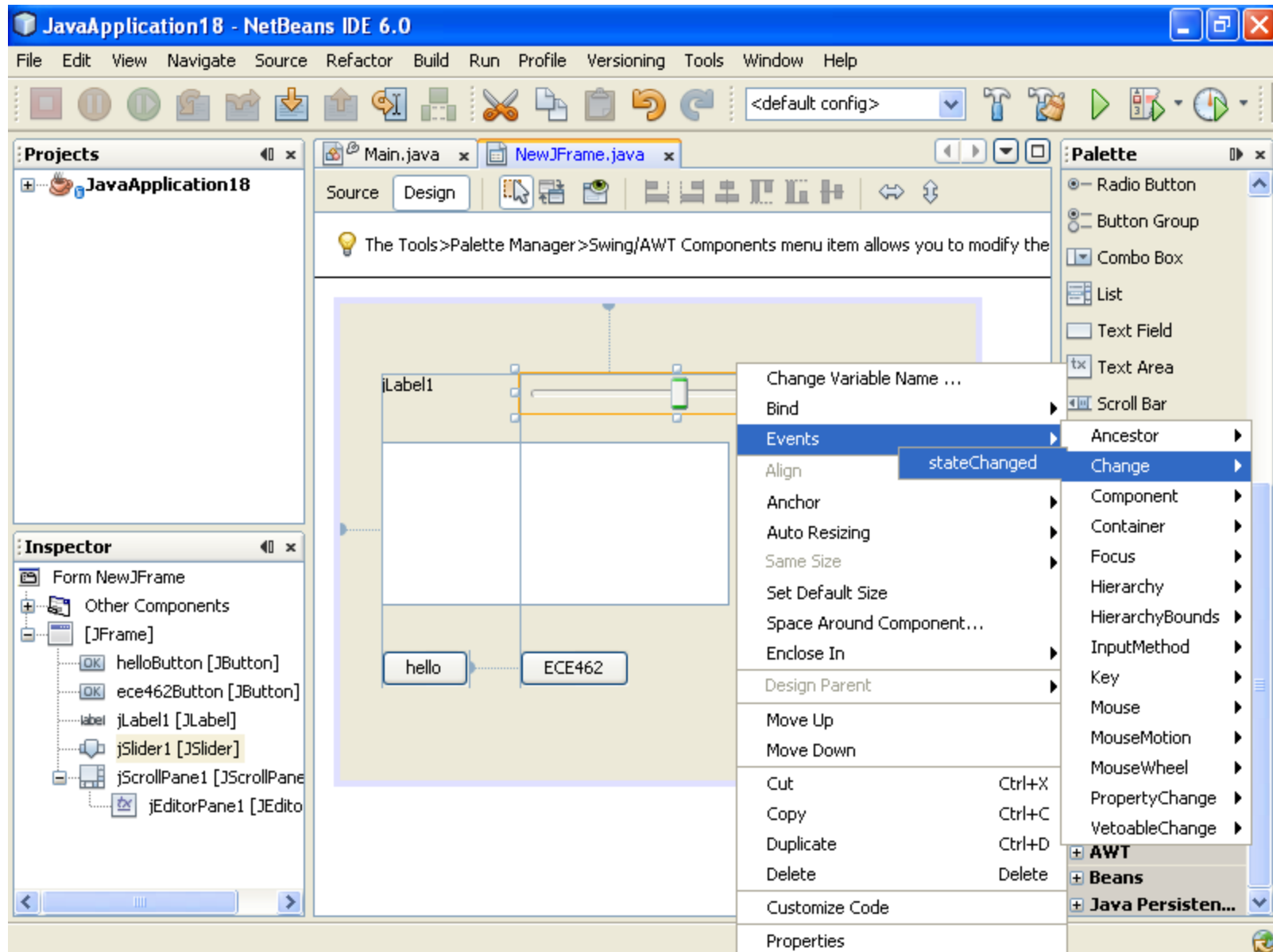
Action Performed

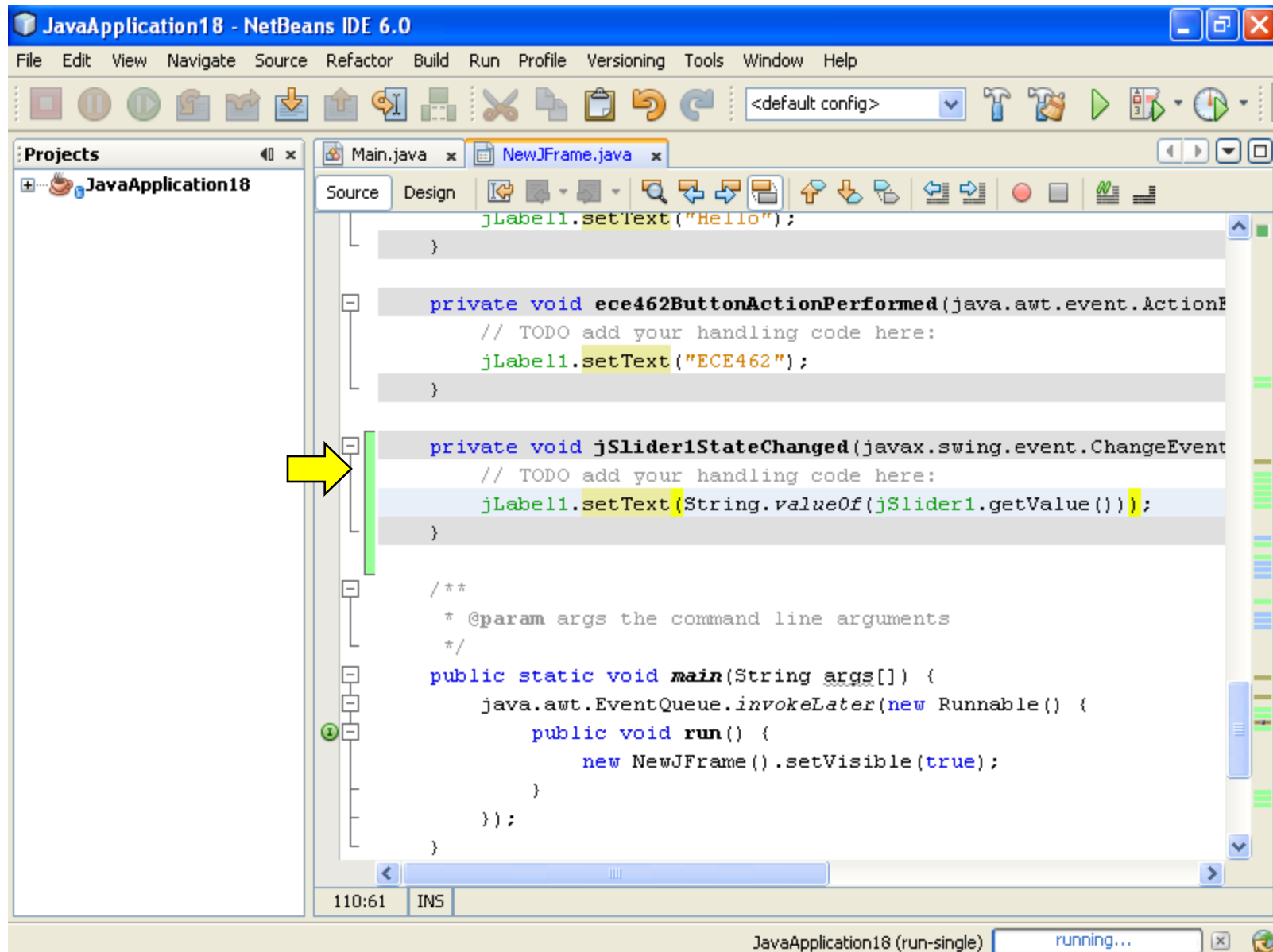


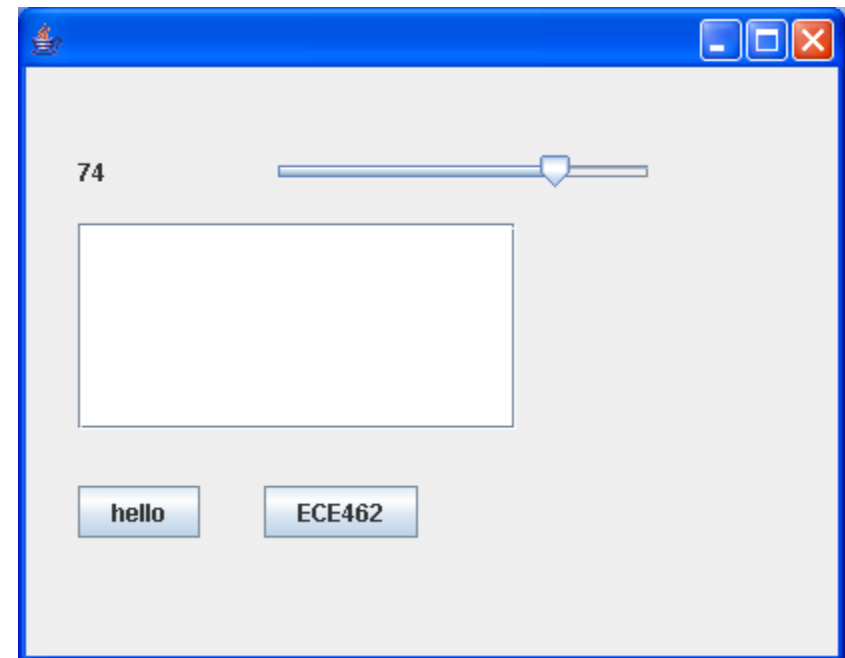
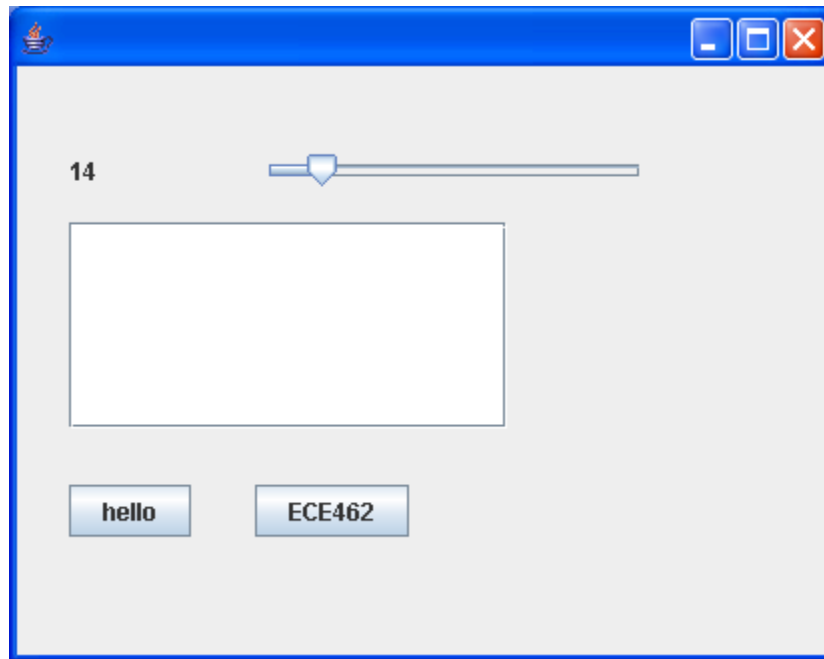


Handle Scrollbar Events

Events
Change
State Changed







JSlider (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://java.sun.com/javase/6/docs/api/javax/swing/JSlider.html

Overview Package **Class** Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Java™ Platform
Standard Ed. 6

javax.swing

Class JSlider

[java.lang.Object](#)

- [java.awt.Component](#)
 - [java.awt.Container](#)
 - [javax.swing.JComponent](#)
 - javax.swing.JSlider**

All Implemented Interfaces:
[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#), [SwingConstants](#)

```
public class JSlider
extends JComponent
implements SwingConstants, Accessible
```

A component that lets the user graphically select a value by sliding a knob within a bounded interval.

The slider can show both major tick marks, and minor tick marks between the major ones. The number of values between the tick marks is controlled with `setMajorTickSpacing` and `setMinorTickSpacing`. Painting of tick marks is controlled by `setPaintTicks`.

Done

JSlider (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://java.sun.com/javase/6/docs/api/javax/swing/JSlider.html

Google

Fields inherited from interface [java.awt.image.ImageObserver](#)

[ABORT](#), [ALLBITS](#), [ERROR](#), [FRAMEBITS](#), [HEIGHT](#), [PROPERTIES](#), [SOMEBITS](#), [WIDTH](#)

Constructor Summary

JSlider()
Creates a horizontal slider with the range 0 to 100 and an initial value of 50. ←

JSlider([BoundedRangeModel](#) brm)
Creates a horizontal slider using the specified [BoundedRangeModel](#).

JSlider(int orientation)
Creates a slider using the specified orientation with the range 0 to 100 and an initial value of 50.

JSlider(int min, int max)
Creates a horizontal slider using the specified min and max with an initial value equal to the average of the min plus max.

JSlider(int min, int max, int value)
Creates a horizontal slider using the specified min, max and value.

JSlider(int orientation, int min, int max, int value)
Creates a slider with the specified orientation and the specified minimum, maximum, and initial values.

Method Summary

void	addChangeListener (ChangeListener l) Adds a ChangeListener to the slider.
------	--

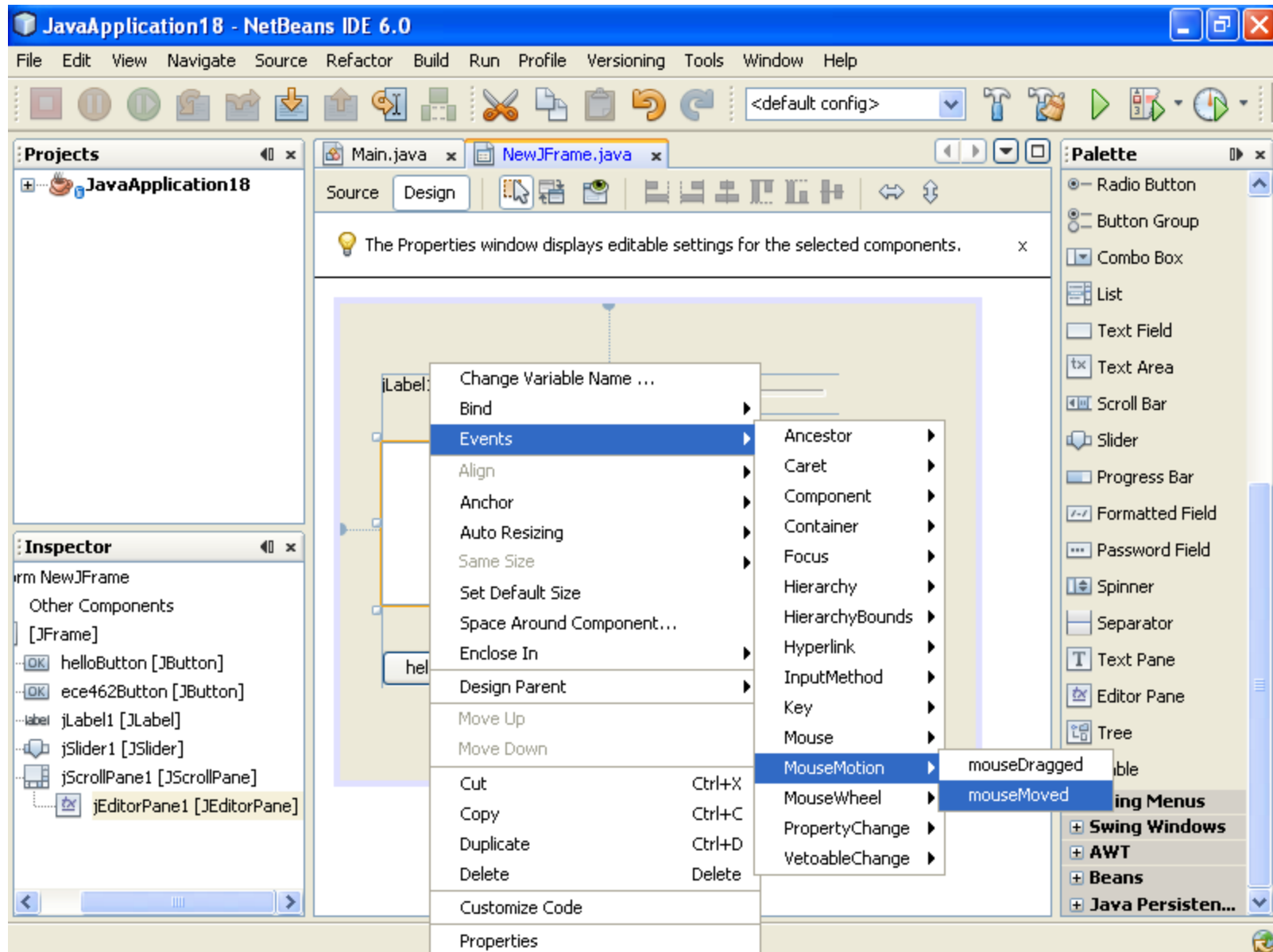
Done

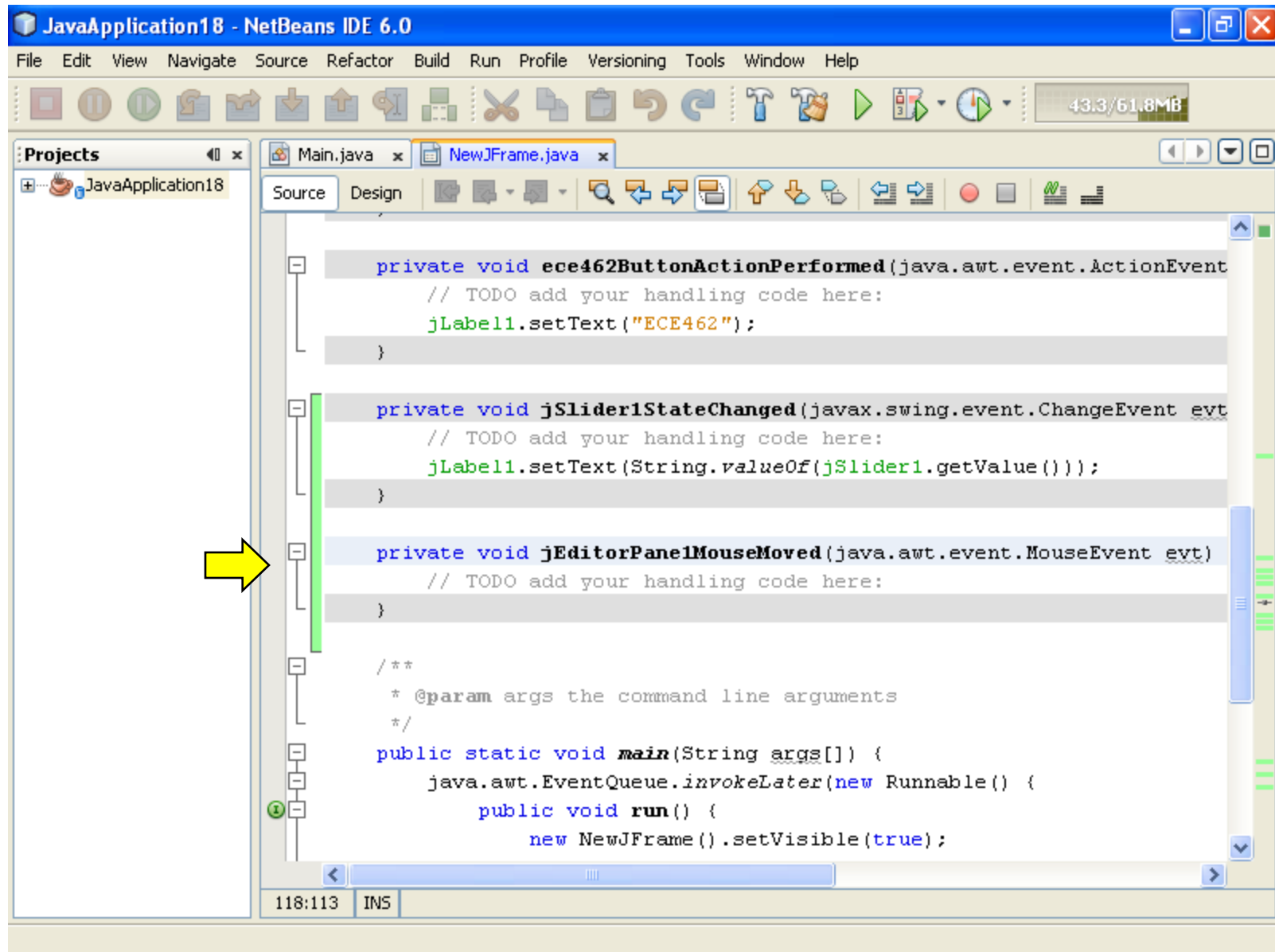
Handle Mouse Motion

Events

Mouse Motion

MouseMoved





Self Test

ECE 462

Object-Oriented Programming

using C++ and Java

Qt Signals and Slots

Yung-Hsiang Lu
yunlu@purdue.edu

Write Qt Programs without Eclipse (still Eclipse for version control)

A better way to know how it works.

Many examples are obtained by modified Qt samples. Copyright Trolltech.

Qt Examples

Qt 4.3: Qt Reference Documentation (Open Source Edition) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/index.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

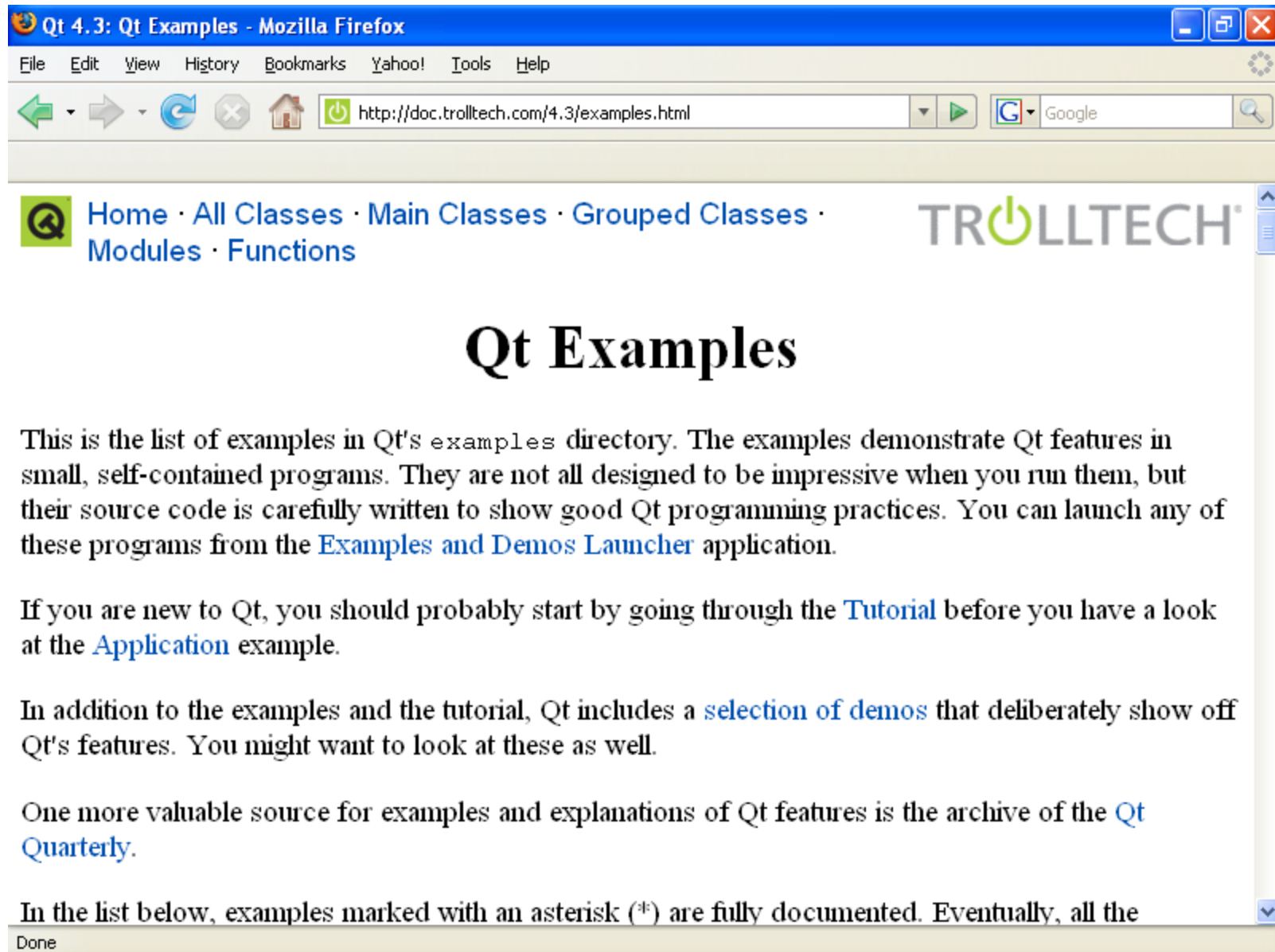
TROLLTECH

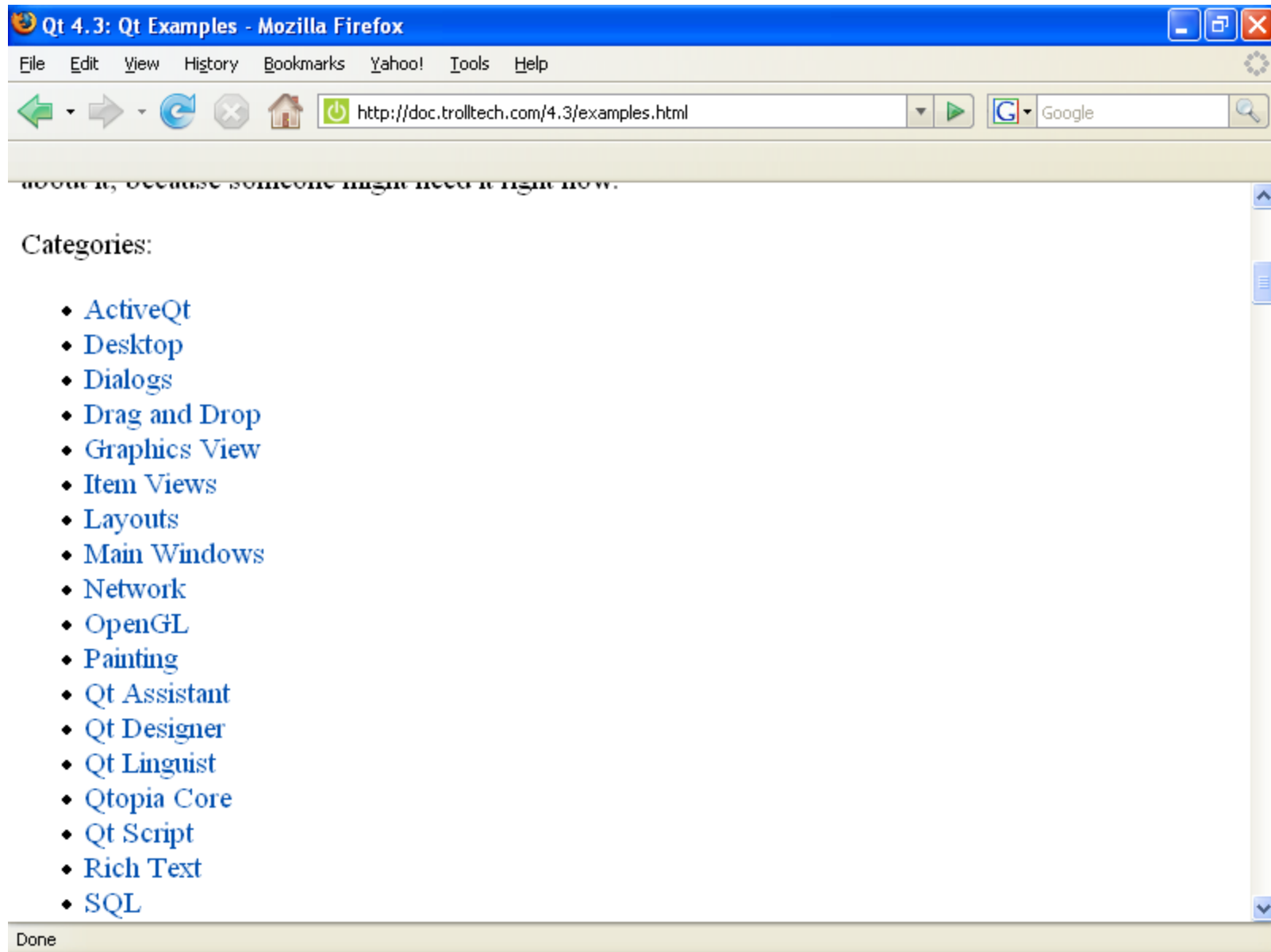
Qt Reference Documentation (Open Source Edition)

Note: This edition is for the development of [Free and Open Source](#) software only; see [Qt Commercial Editions](#).

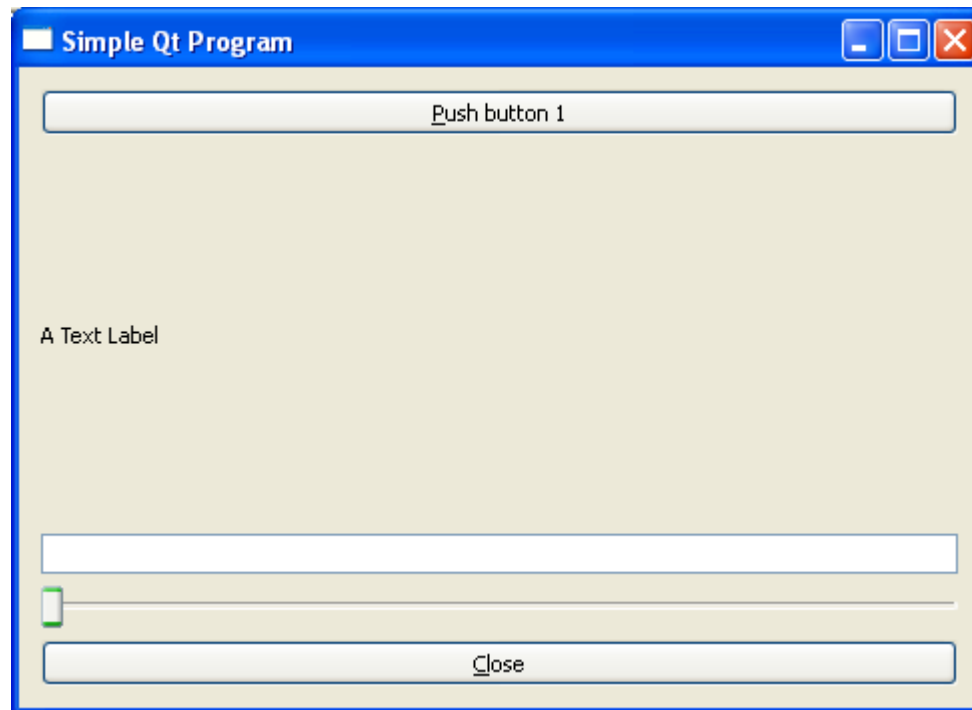
Getting Started	General	Developer Resources
• What's New in Qt 4.3 • How to Learn Qt • Installation • Tutorial and Examples • Porting from Qt 3 to Qt 4	• About Qt • About Trolltech • Commercial Edition • Open Source Edition • Frequently Asked Questions	• Mailing Lists • Qt Community Web Sites • Qt Quarterly • How to Report a Bug • Other Online Resources

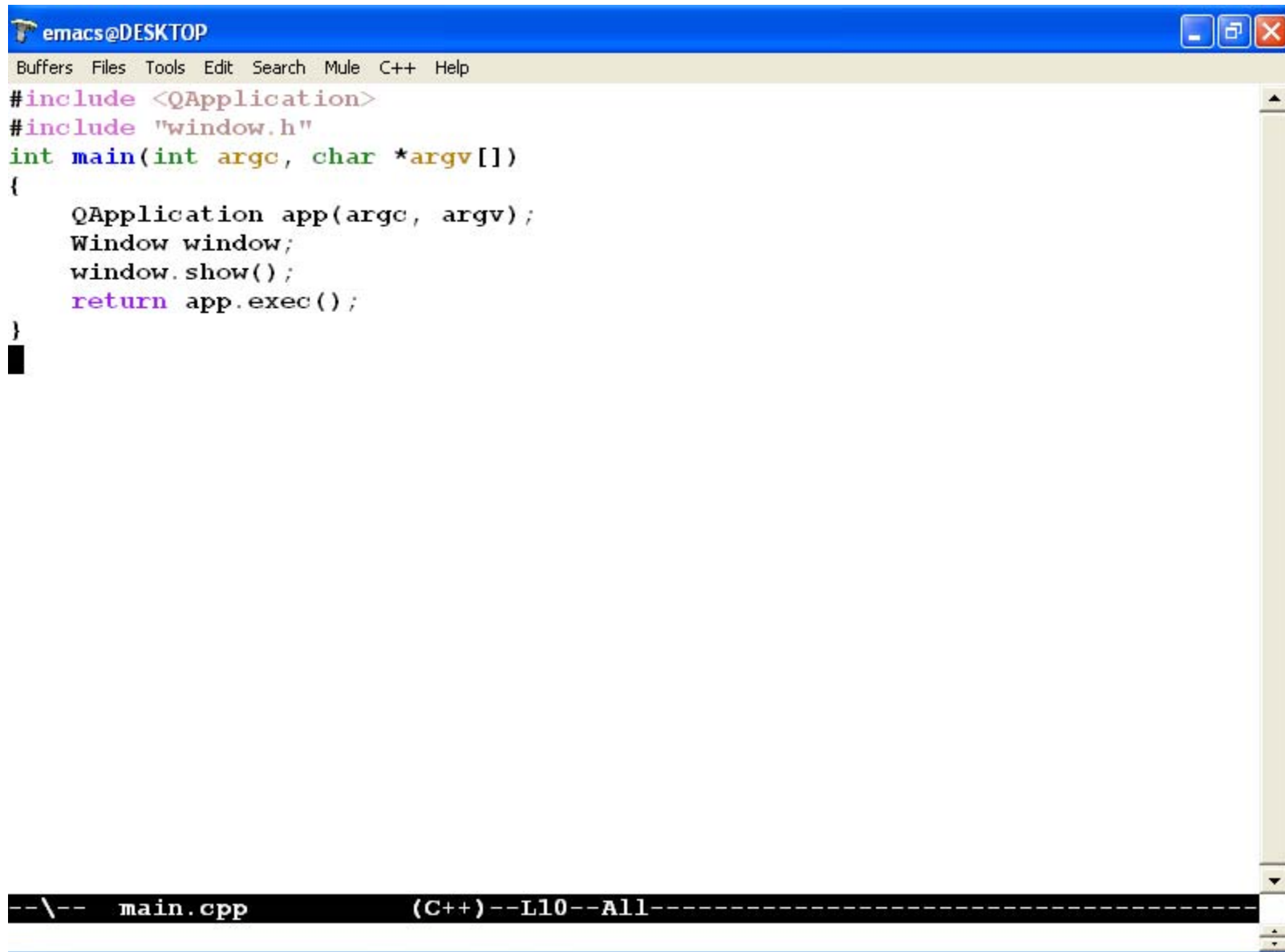
Done





Simple Example Vertical Layout





```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help
#include <QApplication>
#include "window.h"
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    Window window;
    window.show();
    return app.exec();
}

--\--  main.cpp          (C++)--L10--A11-----
```

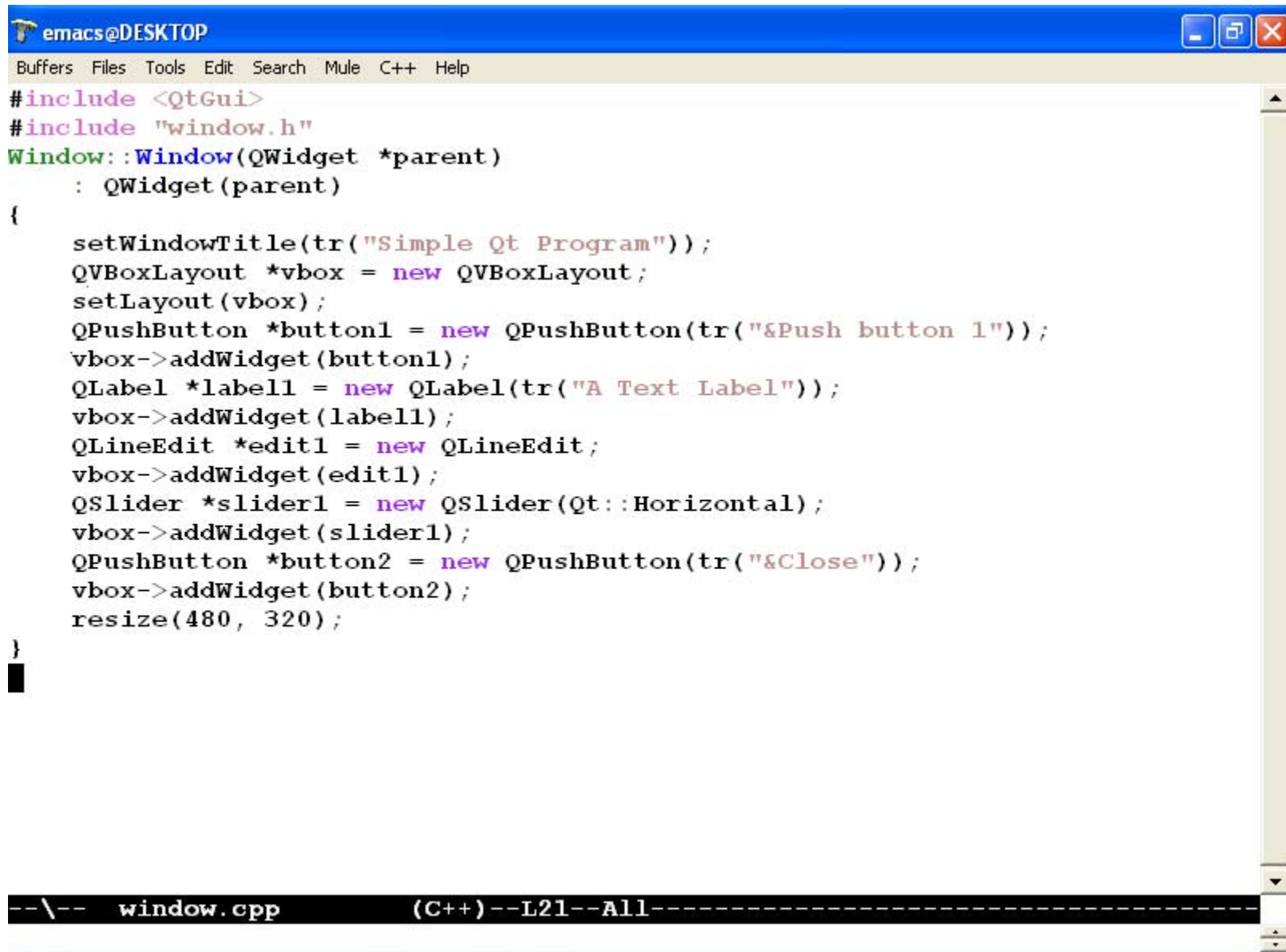
```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C  Help

#ifndef WINDOW_H
#define WINDOW_H
#include <QWidget>
class QGroupBox;
class Window : public QWidget
{
    Q_OBJECT

public:
    Window(QWidget *parent = 0);
private:
};

#endif
```

--\-- window.h (C)--L15--All-----



```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help
#include <QtGui>
#include "window.h"
Window::Window(QWidget *parent)
    : QWidget(parent)
{
    setWindowTitle(tr("Simple Qt Program"));
    QVBoxLayout *vbox = new QVBoxLayout;
    setLayout(vbox);
    QPushButton *button1 = new QPushButton(tr("&Push button 1"));
    vbox->addWidget(button1);
    QLabel *label1 = new QLabel(tr("A Text Label"));
    vbox->addWidget(label1);
    QLineEdit *edit1 = new QLineEdit;
    vbox->addWidget(edit1);
    QSlider *slider1 = new QSlider(Qt::Horizontal);
    vbox->addWidget(slider1);
    QPushButton *button2 = new QPushButton(tr("&Close"));
    vbox->addWidget(button2);
    resize(480, 320);
}

--\-- window.cpp      (C++)--L21--All-----
```

Create Qt Executable

1. `qmake -project`
 2. `qmake`
 3. `make`
- This works for both Linux and Windows.

```
C:\ Qt 4.3.3
C:\yunglu\ece462\qtexamples\qtgui1>dir
Volume in drive C has no label.
Volume Serial Number is 78E2-246B

Directory of C:\yunglu\ece462\qtexamples\qtgui1

03/14/2008  11:20 PM    <DIR>          .
03/14/2008  11:20 PM    <DIR>          ..
03/14/2008  10:59 PM                185 main.cpp
03/14/2008  11:17 PM                658 window.cpp
03/14/2008  10:45 PM                191 window.h
               3 File(s)              1,034 bytes
               2 Dir(s)  4,812,394,496 bytes free

C:\yunglu\ece462\qtexamples\qtgui1>
```

```
C:\ Qt 4.3.3
03/14/2008 11:20 PM <DIR>
03/14/2008 10:59 PM      185 main.cpp
03/14/2008 11:17 PM     658 window.cpp
03/14/2008 10:45 PM     191 window.h
          3 File(s)      1,034 bytes
          2 Dir(s)  4,812,394,496 bytes free

C:\yunglu\ece462\qtexamples\qtgui1>qmake -project

C:\yunglu\ece462\qtexamples\qtgui1>dir
Volume in drive C has no label.
Volume Serial Number is 78E2-246B

Directory of C:\yunglu\ece462\qtexamples\qtgui1

03/14/2008 11:20 PM <DIR>      .
03/14/2008 11:20 PM <DIR>      ..
03/14/2008 10:59 PM      185 main.cpp
03/14/2008 11:20 PM     341 qtgui1.pro
03/14/2008 11:17 PM     658 window.cpp
03/14/2008 10:45 PM     191 window.h
          4 File(s)      1,375 bytes
          2 Dir(s)  4,812,394,496 bytes free

C:\yunglu\ece462\qtexamples\qtgui1>
```

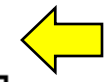


```
C:\ Qt 4.3.3
2 Dir(s) 4,812,394,496 bytes free
C:\yunglu\ece462\qtexamples\qtgui1>qmake
C:\yunglu\ece462\qtexamples\qtgui1>dir
Volume in drive C has no label.
Volume Serial Number is 78E2-246B

Directory of C:\yunglu\ece462\qtexamples\qtgui1

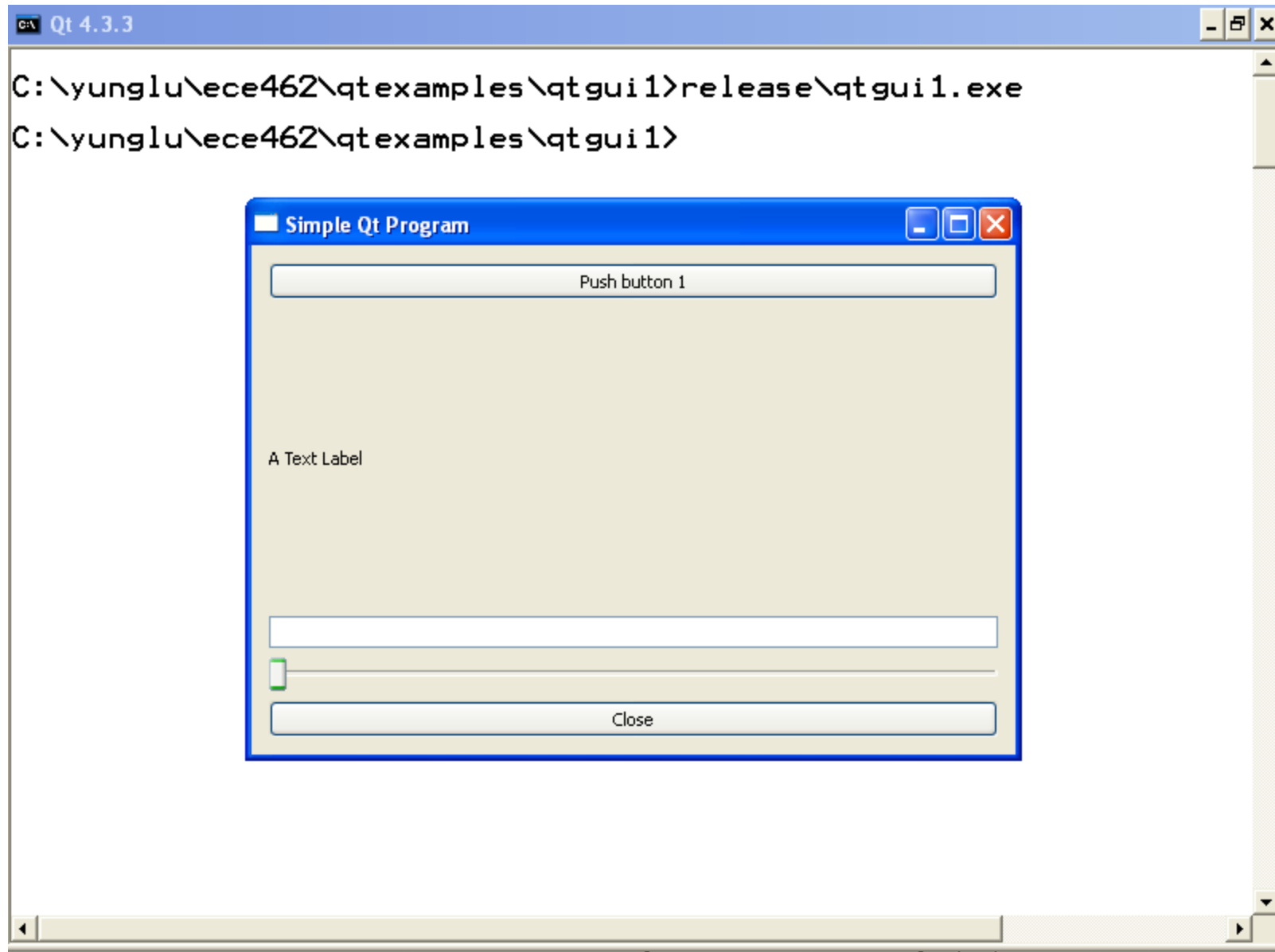
03/14/2008 11:22 PM <DIR> .
03/14/2008 11:22 PM <DIR> ..
03/14/2008 11:22 PM <DIR> debug
03/14/2008 10:59 PM 185 main.cpp
03/14/2008 11:22 PM 5,249 Makefile
03/14/2008 11:22 PM 5,105 Makefile.Debug
03/14/2008 11:22 PM 5,178 Makefile.Release
03/14/2008 11:20 PM 341 qtgui1.pro
03/14/2008 11:22 PM <DIR> release
03/14/2008 11:17 PM 658 window.cpp
03/14/2008 10:45 PM 191 window.h
7 File(s) 16,907 bytes
4 Dir(s) 4,812,173,312 bytes free

C:\yunglu\ece462\qtexamples\qtgui1>
```



```
Qt 4.3.3
C:\yunglu\ece462\qtexamples\qtgui1>make
mingw32-make -f Makefile.Release
mingw32-make[1]: Entering directory 'C:/yunglu/ece462/qtexamples/
g++ -c -O2 -frtti -fexceptions -mthreads -Wall -DUNICODE -DQT_LAR
-DQT_DLL -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -DQT_THREAD_SUP
-QMAIN -I"..\..\..\Qt\include\QtCore" -I"..\..\..\Qt\include\QtCo
\Qt\include\QtGui" -I"..\..\..\Qt\include\QtGui" -I"..\..\..\Qt\i
I"c:\yunglu\Qt\include\ActiveQt" -I"release" -I"." -i"..\..\..\Qt
g++" -o release\main.o main.cpp
g++ -c -O2 -frtti -fexceptions -mthreads -Wall -DUNICODE -DQT_LAR
-DQT_DLL -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -DQT_THREAD_SUP
-QMAIN -I"..\..\..\Qt\include\QtCore" -I"..\..\..\Qt\include\QtCo
\Qt\include\QtGui" -I"..\..\..\Qt\include\QtGui" -I"..\..\..\Qt\i
I"c:\yunglu\Qt\include\ActiveQt" -I"release" -I"." -i"..\..\..\Qt
g++" -o release>window.o window.cpp
C:\yunglu\Qt\bin\moc.exe -DUNICODE -DQT_LARGEFILE_SUPPORT -DQT_DL
-DQT_GUI_LIB -DQT_CORE_LIB -DQT_THREAD_SUPPORT -DQT_NEEDS_QMAIN
include\QtCore" -I"..\..\..\Qt\include\QtCore" -I"..\..\..\Qt\inc
..\..\..\Qt\include\QtGui" -I"..\..\..\Qt\include" -I"." -I"c:\yu
\ActiveQt" -I"release" -I"." -i"..\..\..\Qt\mkspecs\win32-g++" -D
2 window.h -o release\moc_window.cpp
g++ -c -O2 -frtti -fexceptions -mthreads -Wall -DUNICODE -DQT_LAR
-DQT_DLL -DQT_NO_DEBUG -DQT_GUI_LIB -DQT_CORE_LIB -DQT_THREAD_SUP
-QMAIN -I"..\..\..\Qt\include\QtCore" -I"..\..\..\Qt\include\QtCo
\Qt\include\QtGui" -I"..\..\..\Qt\include\QtGui" -I"..\..\..\Qt\i
I"c:\yunglu\Qt\include\ActiveQt" -I"release" -I"." -i"..\..\..\Qt
g++" -o release\moc_window.o release\moc_window.cpp
g++ -enable-stdcall-fixup -Wl,-enable-auto-import -Wl,-enable-run
oc -Wl,-s -mthreads -Wl -Wl,-subsystem,windows -o release\qtgui1.
n.o release>window.o release\moc_window.o -L"c:\yunglu\Qt\lib" -
in -lQtGui4 -lQtCore4
mingw32-make[1]: Leaving directory 'C:/yunglu/ece462/qtexamples/q
C:\yunglu\ece462\qtexamples\qtgui1>
```





```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C  Help

#ifndef WINDOW_H
#define WINDOW_H
#include <QWidget>
class QGroupBox;
class Window : public QWidget
{
    Q_OBJECT

public:
    Window(QWidget *parent = 0);
private:
};

#endif
```

derived class of QWidget

macro to enable signals and slots

default value of the parent widget

--\-- window.h (C) --L15--All-----

menu-bar buffer

emacs@DESKTOP

Buffers Files Tools Edit Search Mule C++ Help

```
#include <QtGui>
#include "window.h"
Window::Window(QWidget *parent)
    : QWidget(parent)
{
    setWindowTitle(tr("Simple Qt Program"));
    QVBoxLayout *vbox = new QVBoxLayout;
    setLayout(vbox);
    QPushButton *button1 = new QPushButton(tr("&Open"));
    vbox->addWidget(button1);
    QLabel *label1 = new QLabel(tr("A Text Label"));
    vbox->addWidget(label1);
    QLineEdit *edit1 = new QLineEdit;
    vbox->addWidget(edit1);
    QSlider *slider1 = new QSlider(Qt::Horizontal);
    vbox->addWidget(slider1);
    QPushButton *button2 = new QPushButton(tr("&Close"));
    vbox->addWidget(button2);
    resize(480, 320);
}
```

--\-- window.cpp (C++) --L5--All-----

this includes many Qt classes

window's title

vertical layout

add a button

add a label

add a slider

GUI and OOP

```
vbox->addWidget(button1);  
vbox->addWidget(label1);  
vbox->addWidget(edit1);  
vbox->addWidget(slider1);
```

- These are objects of **different** classes.
- These classes are **derived class** of QWidget.
- If a function requires an object of QWidget, the function can be called with an object of QLabel.
- Any QLabel object is also a QWidget object (**wrong** in the other direction).

Qt 4.3: QVBoxLayout Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qvboxlayout.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TRÖLLTECH

QVBoxLayout Class Reference

[QtGui module]

The QVBoxLayout class lines up widgets vertically. [More...](#)

```
#include <QVBoxLayout>
```

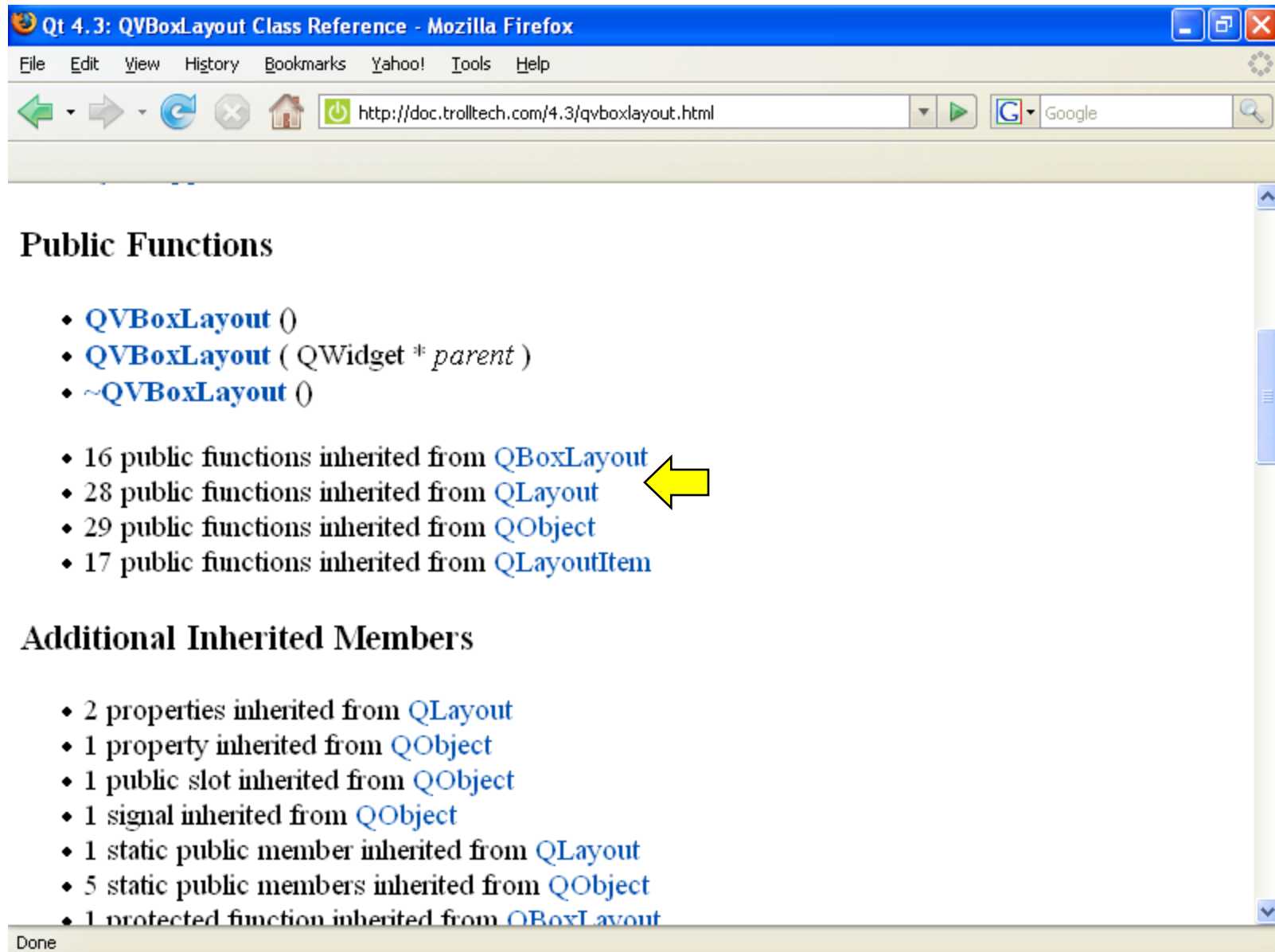
Inherits [QBoxLayout](#).

- [List of all members, including inherited members](#)
- [Qt 3 support members](#)

Public Functions

- [QVBoxLayout\(\)](#)

Done



Qt 4.3: QLayout Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qlayout.html#public-functions

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TROLLTECH

QLayout Class Reference

[QtGui module]

The QLayout class is the base class of geometry managers. [More...](#)

```
#include <QLayout>
```

Inherits [QObject](#) and [QLayoutItem](#).

Inherited by [QBoxLayout](#), [QGridLayout](#), and [QStackedLayout](#).

- ♦ [List of all members, including inherited members](#)
- ♦ [Obsolete members](#)
- ♦ [Qt 3 support members](#)

Done

Qt 4.3: QLayout Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qlayout.html#public-functions

Google

- ♦ 1 property inherited from **QObject**

Public Functions

- ♦ **QLayout** (QWidget * *parent*)
- ♦ **QLayout** ()
- ♦ bool **activate** ()
- ♦ virtual void **addItem** (QLayoutItem * *item*) = 0
- ♦ void **addWidget** (QWidget * *w*)
- ♦ QRect **contentsRect** () const
- ♦ virtual int **count** () const = 0
- ♦ virtual Qt::Orientations **expandingDirections** () const
- ♦ void **getContentsMargins** (int * *left*, int * *top*, int * *right*, int * *bottom*) const
- ♦ virtual int **indexOf** (QWidget * *widget*) const
- ♦ bool **isEnabled** () const
- ♦ virtual QLayoutItem * **itemAt** (int *index*) const = 0
- ♦ virtual QSize **maximumSize** () const
- ♦ QWidget * **menuBar** () const
- ♦ virtual QSize **minimumSize** () const
- ♦ QWidget * **parentWidget** () const

Done

**QPushButton, QLineEdit, and
QSlider all inherit QWidget**

Qt 4.3: QLineEdit Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qlineedit.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TROLLTECH

QLineEdit Class Reference

[QtGui module]

The QLineEdit widget is a one-line text editor. [More...](#)

```
#include <QLineEdit>
```

Inherits [QWidget](#). 

- [List of all members, including inherited members](#)
- [Qt 3 support members](#)

Public Types

- [enum **EchoMode**](#) { Normal, NoEcho, Password, PasswordEchoOnEdit }

Done

Qt 4.3: QSlider Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qslider.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TROLLTECH

QSlider Class Reference

[QtGui module]

The QSlider widget provides a vertical or horizontal slider. [More...](#)

```
#include <QSlider>
```

Inherits [QAbstractSlider](#).

- List of all members, including inherited members
- Qt 3 support members

Public Types

- enum [TickPosition](#) / NoTicks TicksBothSides TicksAbove TicksBelow TicksLeft

Done

Qt 4.3: QAbstractSlider Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qabstractslider.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TROLLTECH

QAbstractSlider Class Reference

[QtGui module]

The QAbstractSlider class provides an integer value within a range. [More...](#)

```
#include <QAbstractSlider>
```

Inherits [QWidget](#). 

Inherited by [QDial](#), [QScrollBar](#), and [QSlider](#).

- [List of all members, including inherited members](#)
- [Qt 3 support members](#)

Public Types

Done

Qt 4.3: QPushButton Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qpushbutton.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TRÖLLTECH

QPushButton Class Reference

[QtGui module]

The QPushButton widget provides a command button. [More...](#)

```
#include <QPushButton>
```

Inherits [QAbstractButton](#).

- List of all members, including inherited members
- Qt 3 support members

Properties

- [autoDefault](#) · bool

Done

Qt 4.3: QAbstractButton Class Reference - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.3/qabstractbutton.html

Google

Home · All Classes · Main Classes · Grouped Classes · Modules · Functions

TROLLTECH

QAbstractButton Class Reference

[QtGui module]

The QAbstractButton class is the abstract base class of button widgets, providing functionality common to buttons. [More...](#)

```
#include <QAbstractButton>
```

Inherits [QWidget](#). 

Inherited by [Q3Button](#), [QCheckBox](#), [QPushButton](#), [QRadioButton](#), and [QToolButton](#).

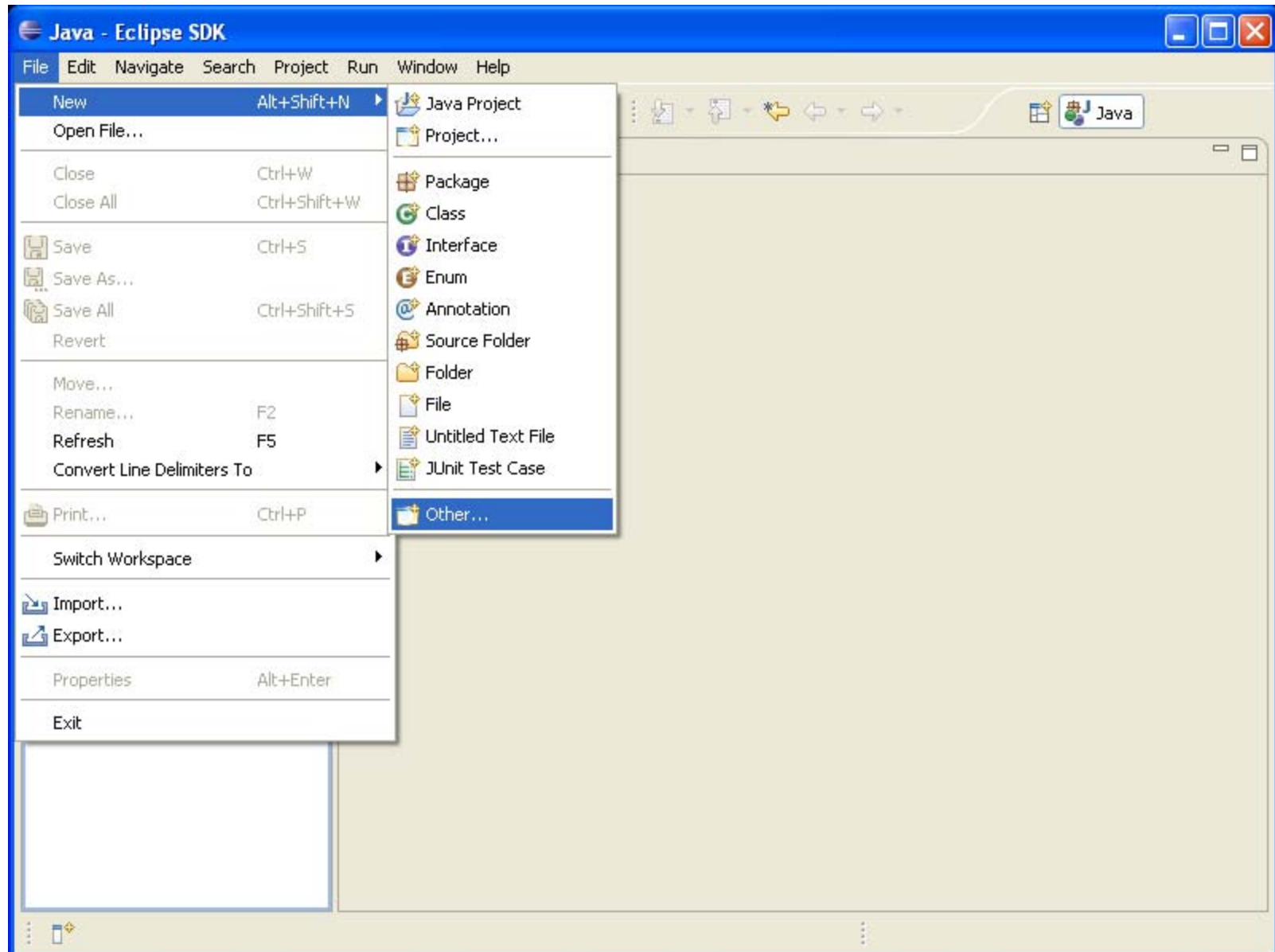
- [List of all members, including inherited members](#)
- [Qt 3 support members](#)

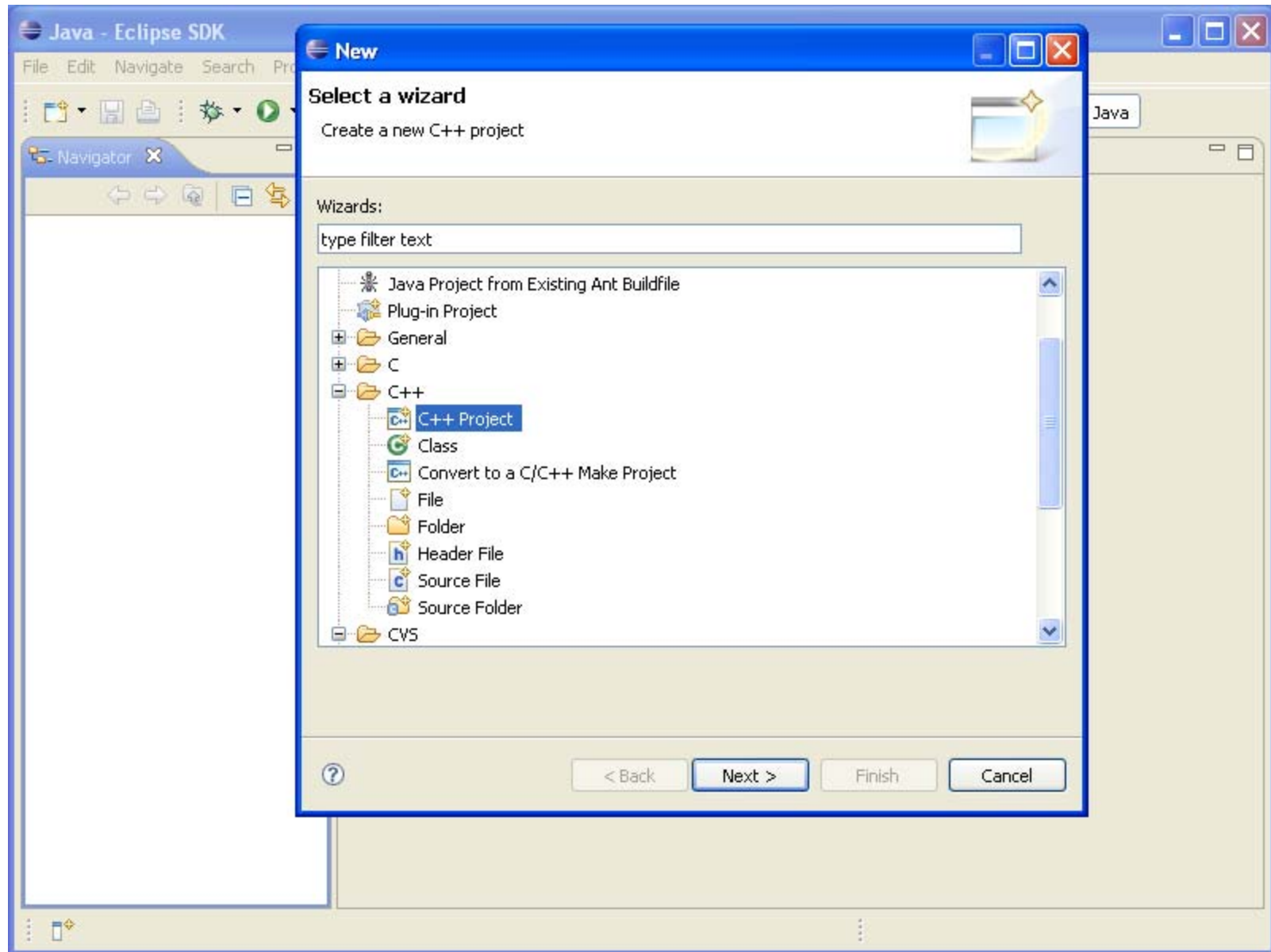
Done

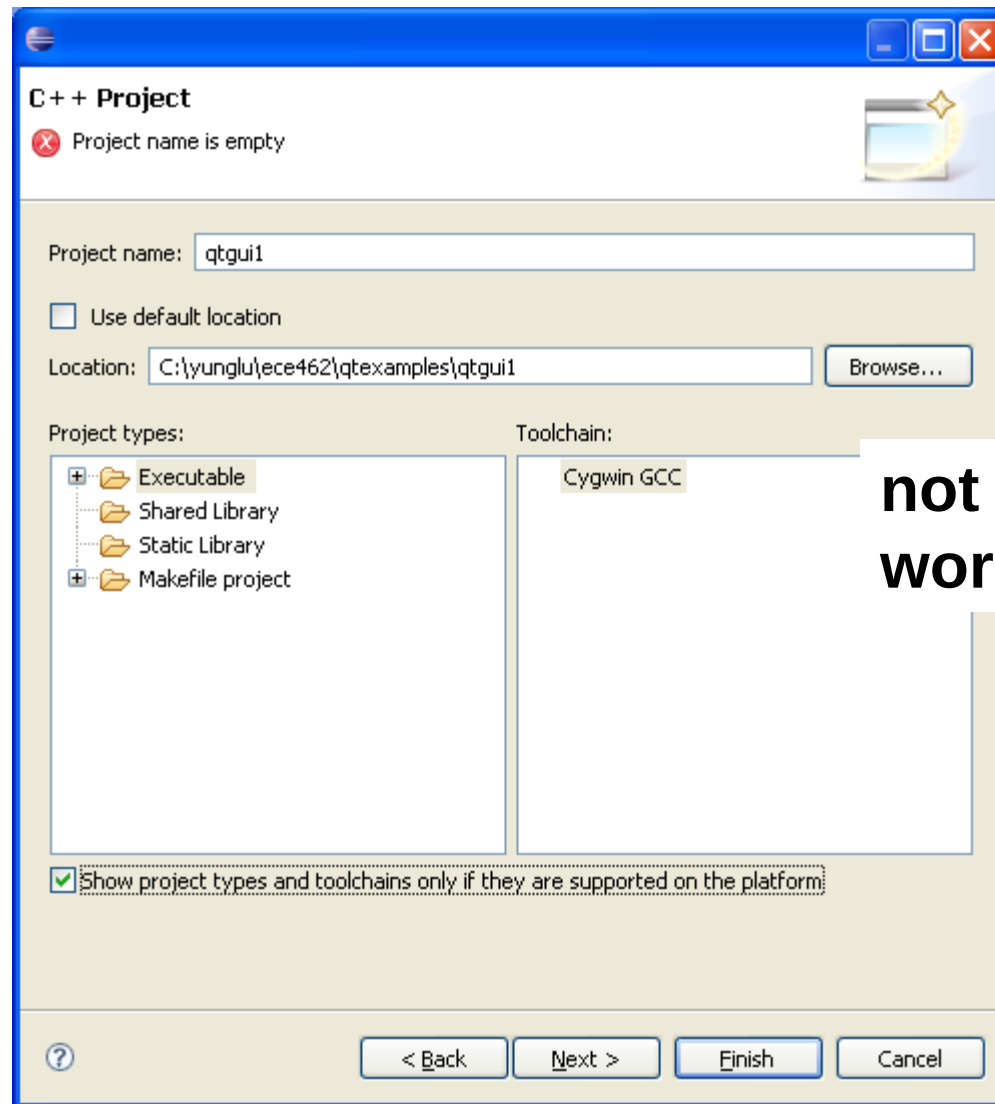
**Now, you should understand why
GUI is often built with OOP.**

Eclipse + Emacs

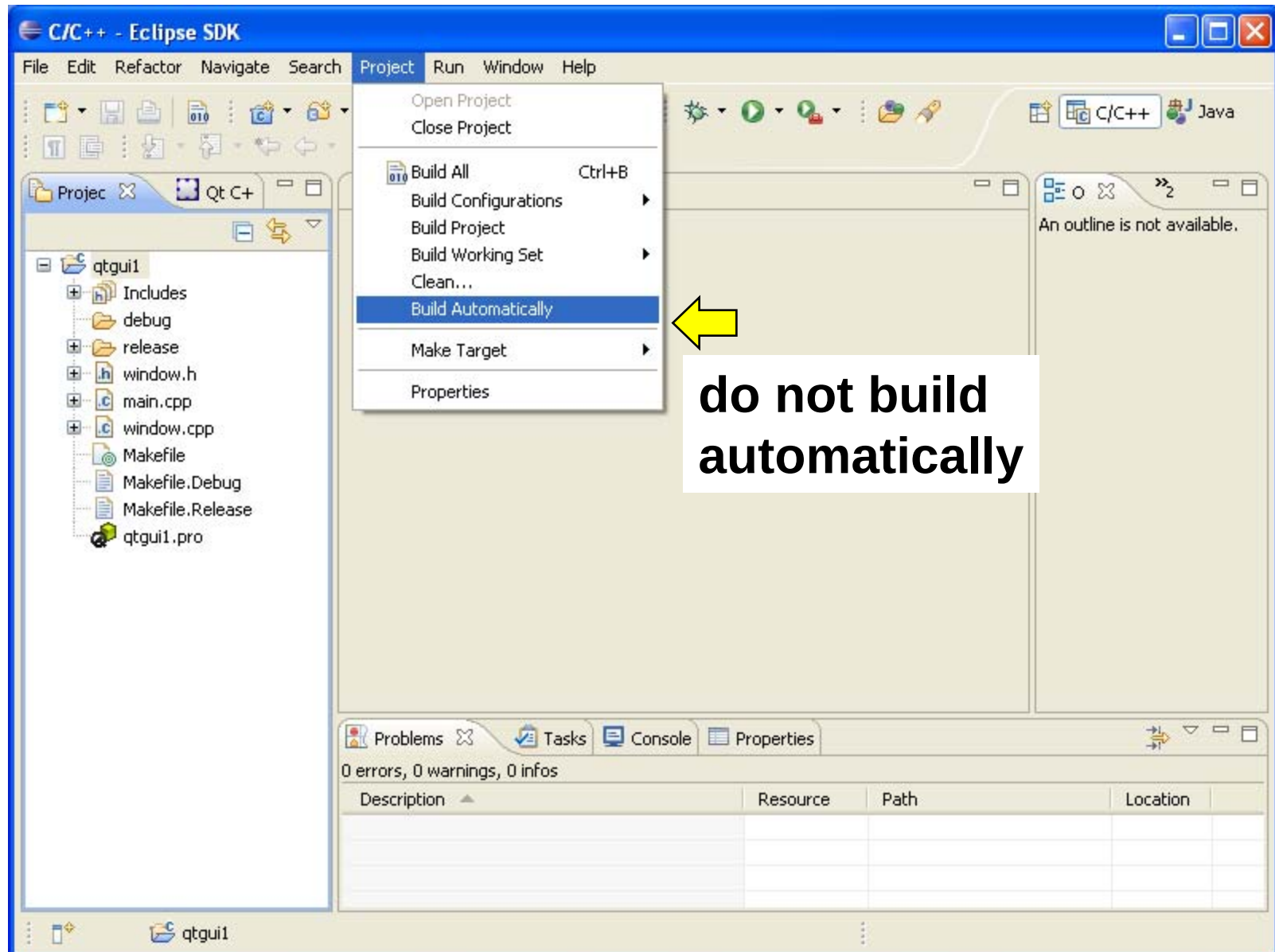
use the version control in Eclipse
edit code using Emacs

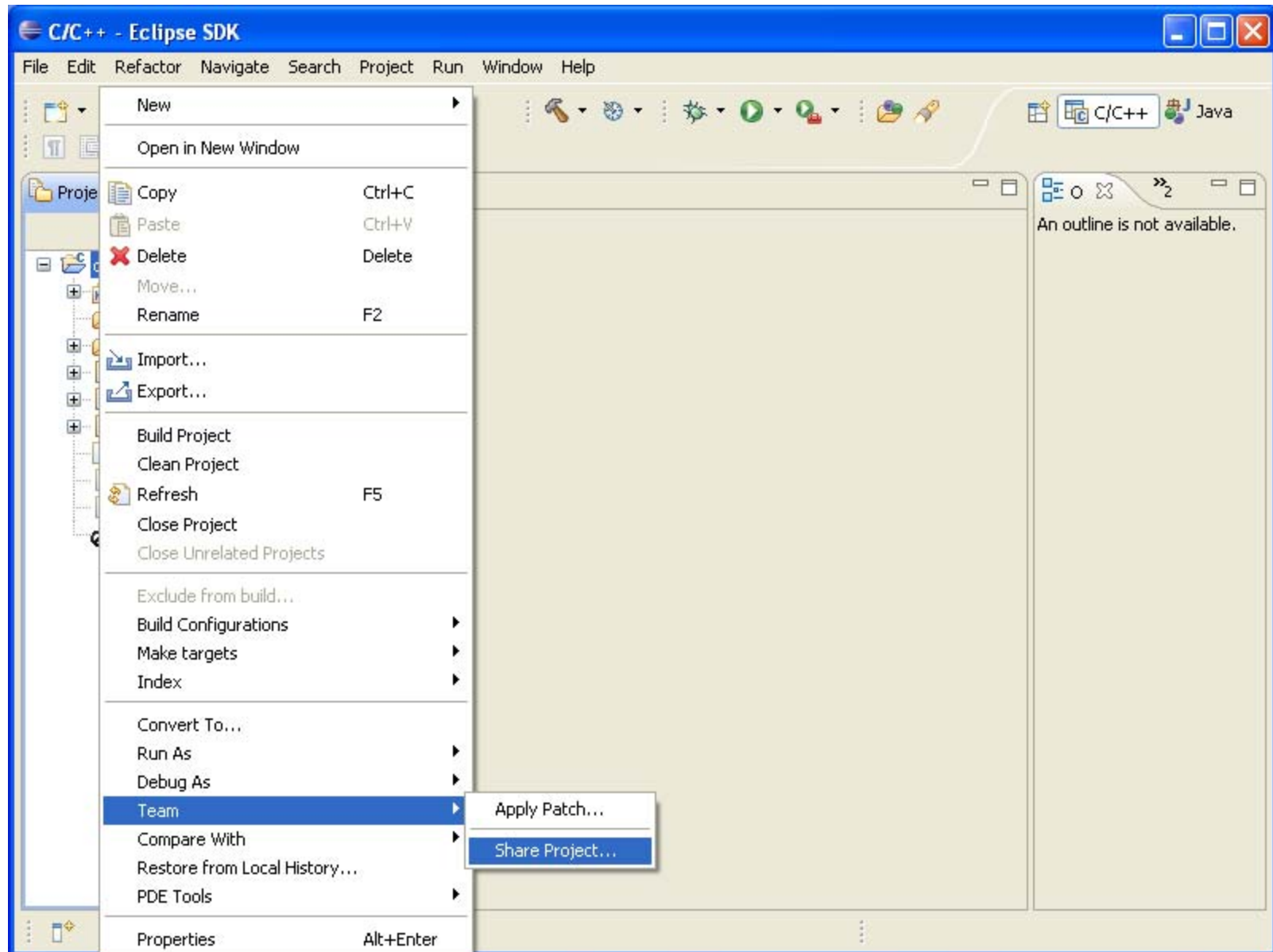






**not the default
workspace**





Share Project

Enter Repository Location Information

Define the location and protocol required to connect with an existing CVS repository.

Location

Host: msee190pc3.ecn.purdue.edu

Repository path: /home/shay/a/ee462b30/projects

Authentication

User: ee462b30

Password: ●●●●●●●●

Connection

Connection type: extssh

☒ Use default port

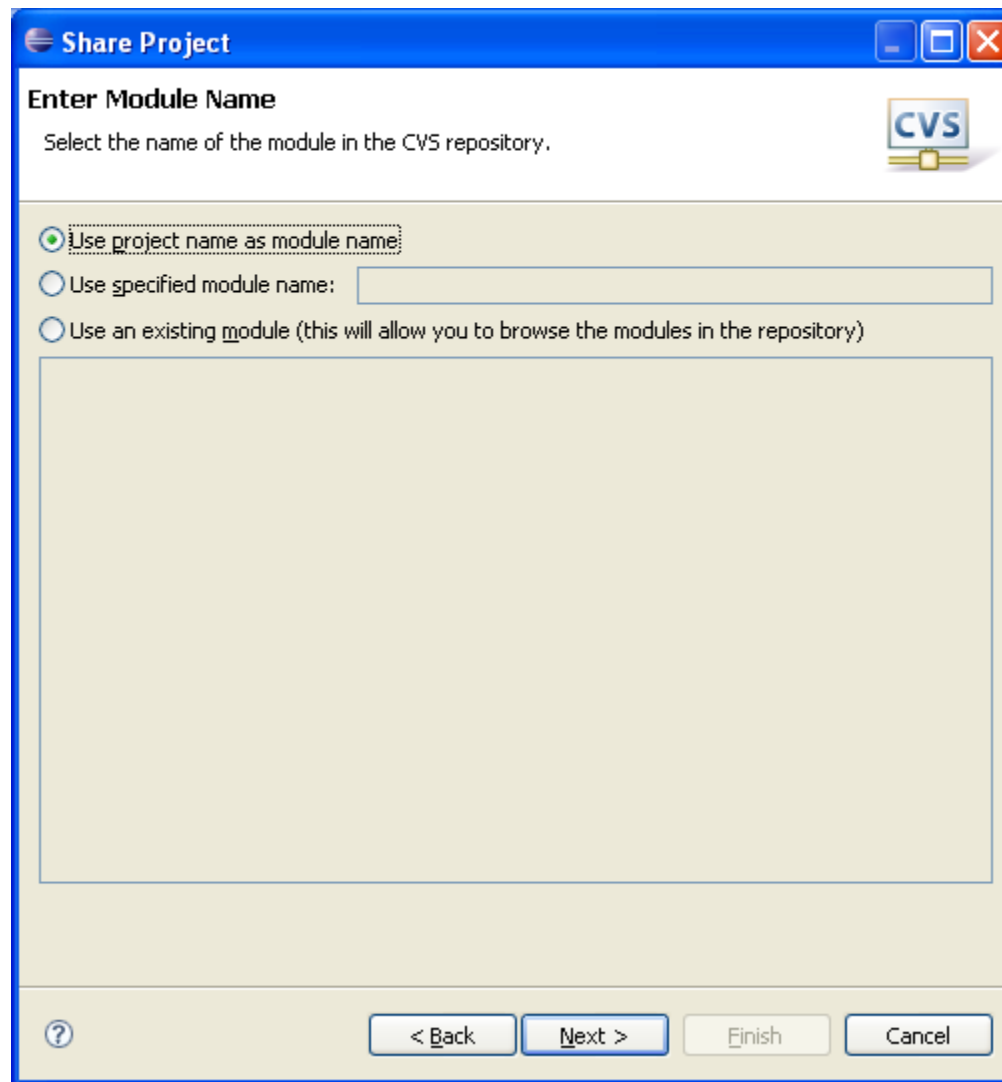
☐ Use port:

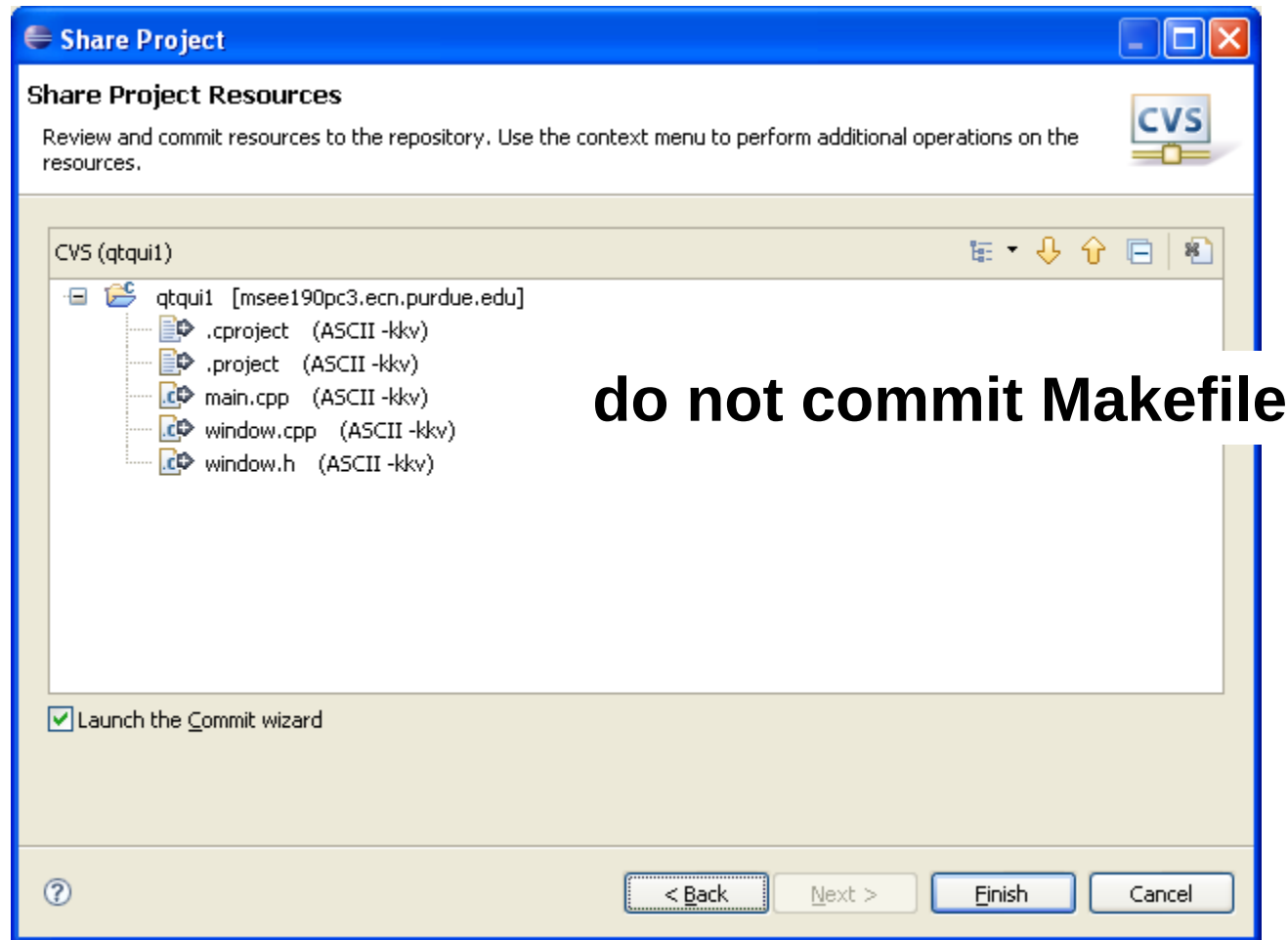
☐ Save password

⚠ Saved passwords are stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

[Configure connection preferences...](#)

? < Back Next > Finish Cancel





Add Slots and Connect Signals

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C  Help

#ifndef WINDOW_H
#define WINDOW_H
#include <QtGui>
class Window : public QWidget
{
    Q_OBJECT

public:
    Window(QWidget *parent = 0) :
private slots:
    void button1Clicked();
    void lineEditChanged();
    void sliderChanged();
private:
    void connectSignals();
    QPushButton *button1;
    QLabel *label1;
    QLineEdit *edit1;
    QSlider *slider1;
    QPushButton *button2;
};
#endif

--\-- window.h (C CVS:1.1)--L22--All-----
Find file: c:/yunglu/ece462/qtexamples/qtgui1/
```

add private functions (slots)



make the objects attributes
because they are used in
multiple functions



CVS version number
shown in emacs



```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

#include <QtGui>
#include "window.h"
Window::Window(QWidget *parent)
    : QWidget(parent)
{
    setWindowTitle(tr("Simple Qt Program"));
    QVBoxLayout *vbox = new QVBoxLayout;
    setLayout(vbox);
    button1 = new QPushButton(tr("&Push button 1"));
    vbox->addWidget(button1);
    label1 = new QLabel(tr("A Text Label"));
    vbox->addWidget(label1);
    edit1 = new QLineEdit;
    vbox->addWidget(edit1);
    slider1 = new QSlider(Qt::Horizontal);
    vbox->addWidget(slider1);
    button2 = new QPushButton(tr("&Close"));
    vbox->addWidget(button2);
    resize(480, 320);
    connectSignals();
}

void Window::connectSignals()
--\-- window.cpp (C++ CVS:1.1)--L23--Top-----
Find file: c:/yunglu/ece462/qtexamples/qtgui1/
```

objects

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

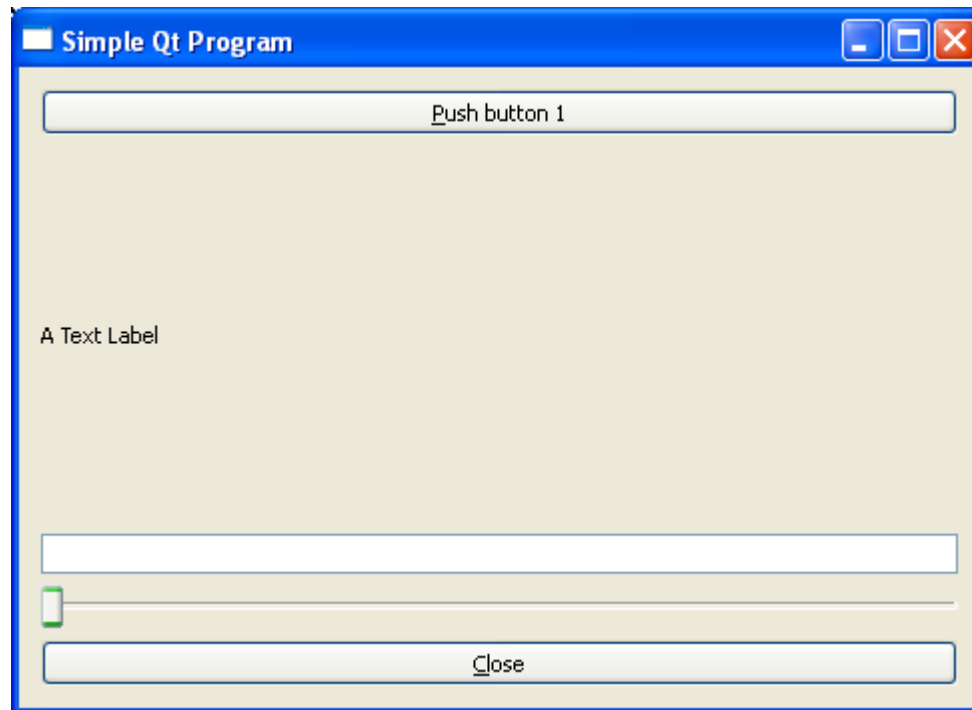
void Window::connectSignals()
{
    connect(button1, SIGNAL(clicked()), this, SLOT(button1Clicked()));
    connect(button2, SIGNAL(clicked()), this, SLOT(close()));
    connect(edit1, SIGNAL(textChanged(QString)),
            this, SLOT(lineEditChanged()));
    connect(slider1, SIGNAL(sliderMoved(int)),
            this, SLOT(sliderChanged()));
}

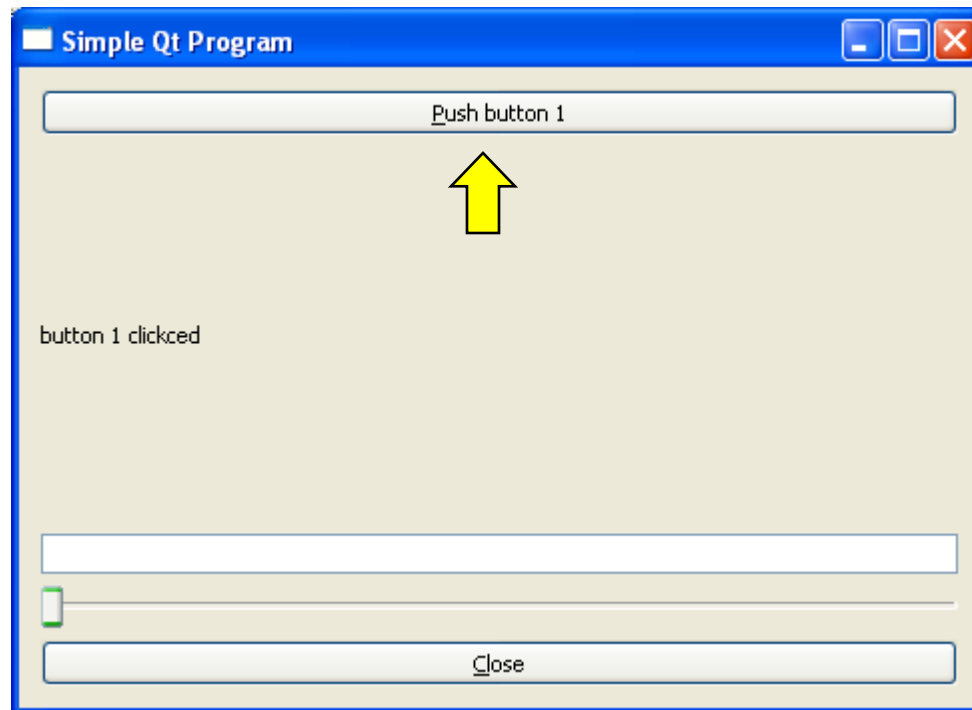
void Window::button1Clicked()
{ label1 -> setText("button 1 clickced"); }

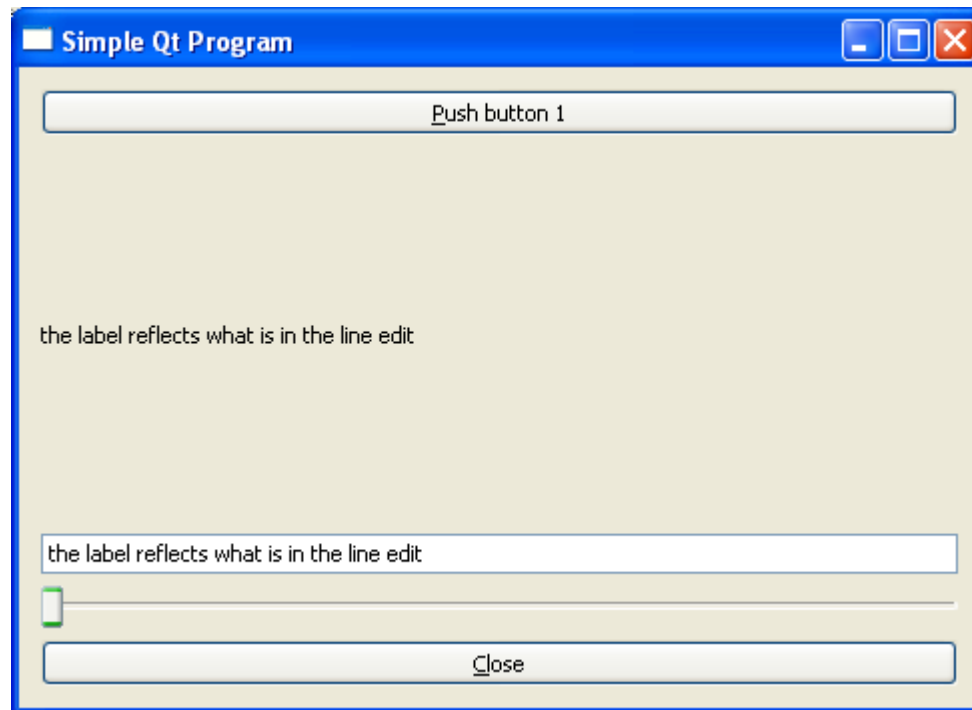
void Window::lineEditChanged()
{ label1 -> setText(edit1 -> text()); }

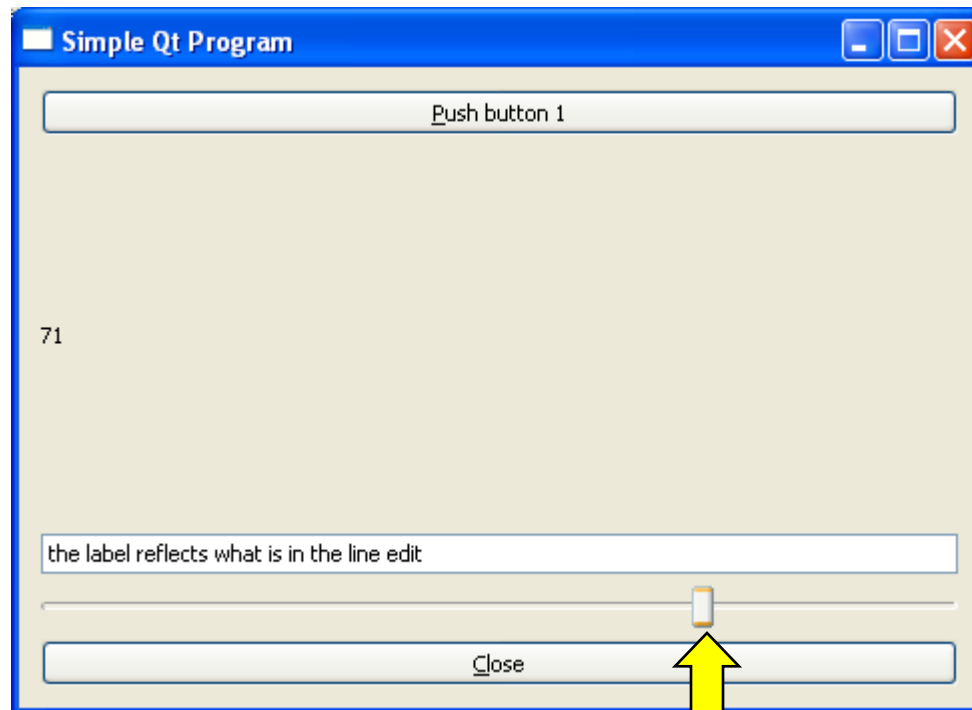
void Window::sliderChanged()
{ label1 -> setText(QString::number(slider1 -> value())); }

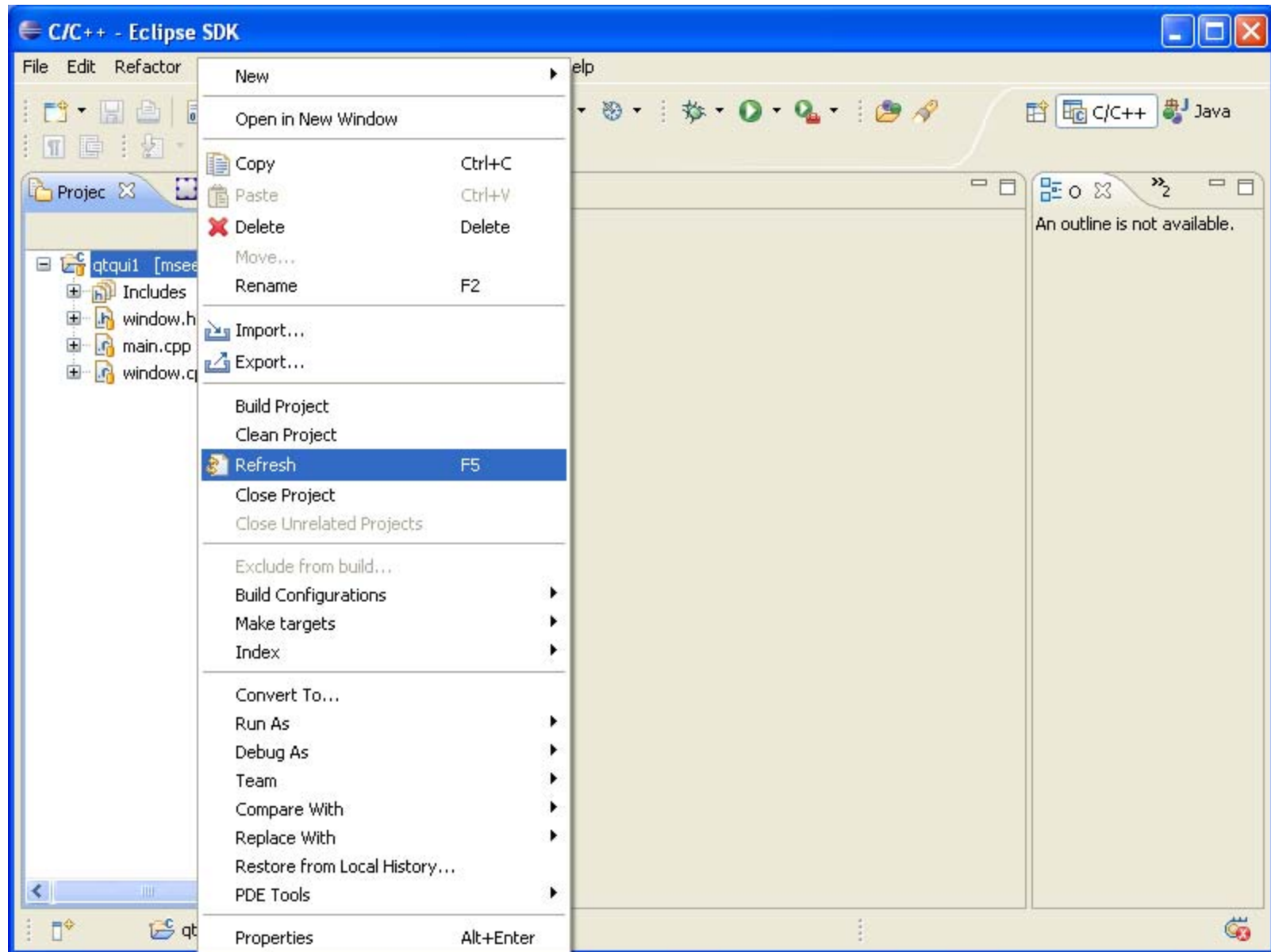
--\--  window.cpp      (C++ CVS:1.1)--L41--Bot-----
Find file: c:/yunglu/ece462/qtexamples/qtgui1/
```

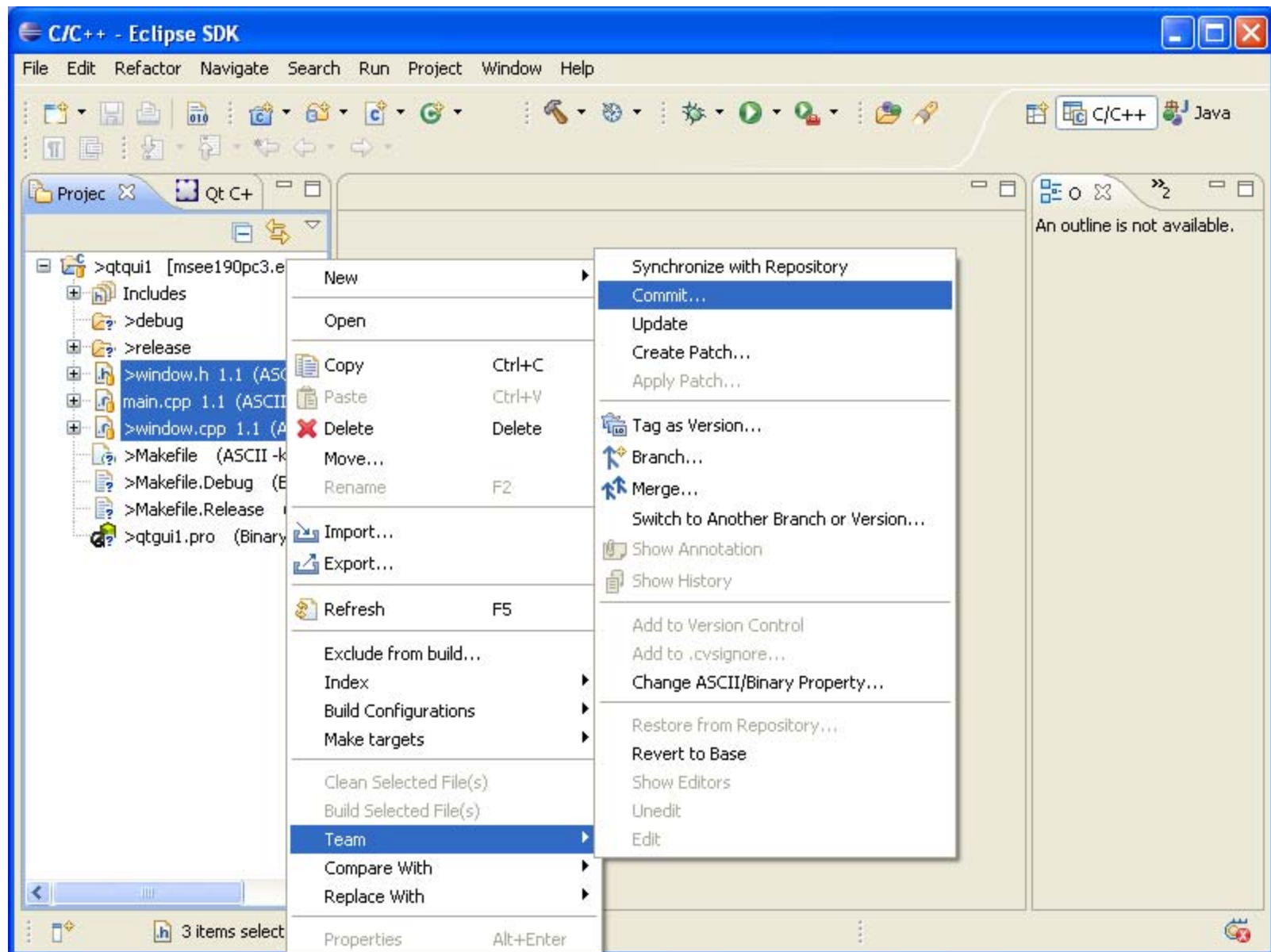


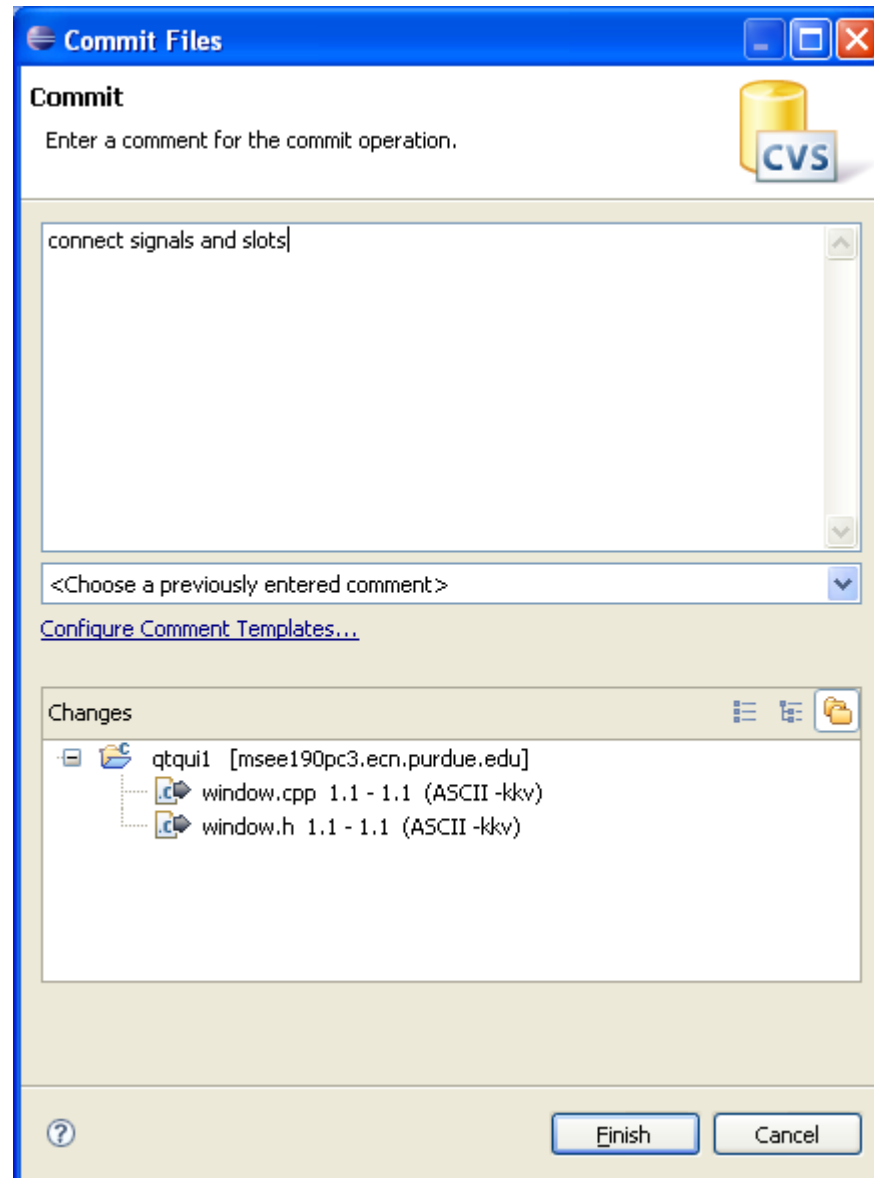




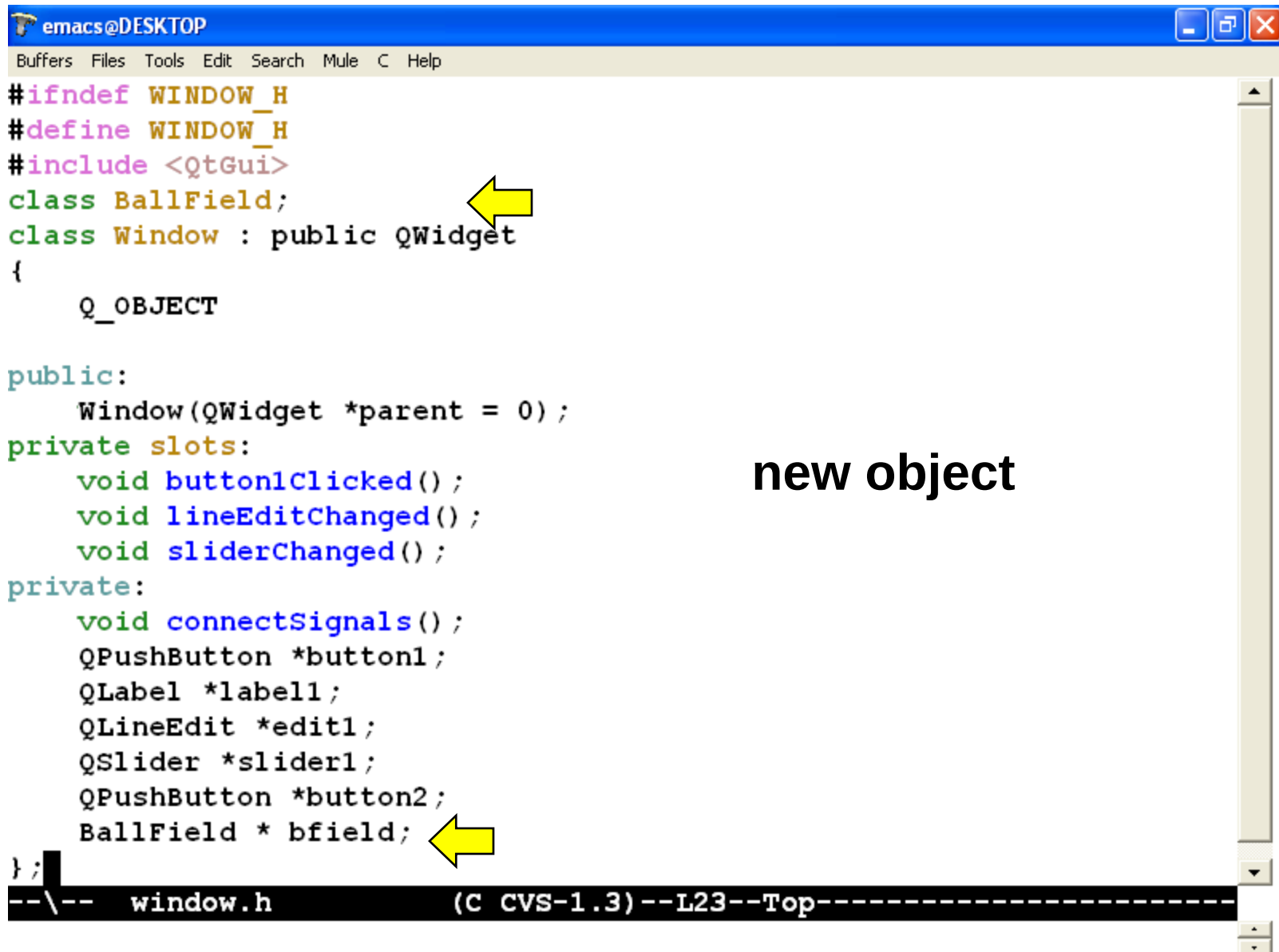








Draw Geometric Shapes



```
#ifndef WINDOW_H
#define WINDOW_H
#include <QtGui>
class BallField;
class Window : public QWidget
{
    Q_OBJECT

public:
    Window(QWidget *parent = 0);
private slots:
    void button1Clicked();
    void lineEditChanged();
    void sliderChanged();
private:
    void connectSignals();
    QPushButton *button1;
    QLabel *label1;
    QLineEdit *edit1;
    QSlider *slider1;
    QPushButton *button2;
    BallField * bfield;
};
```

--\-- window.h (C CVS-1.3)--L23--Top--

new object

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

#include <QtGui>
#include "window.h"
#include "BallField.h"
Window::Window(QWidget *parent)
    : QWidget(parent)
{
    setWindowTitle(tr("Simple Qt Program"));
    QVBoxLayout *vbox = new QVBoxLayout;
    setLayout(vbox);
    button1 = new QPushButton(tr("&Push button 1"));
    vbox->addWidget(button1);
    bfield = new BallField;
    vbox->addWidget(bfield);
    label1 = new QLabel(tr("A Text Label"));
    vbox->addWidget(label1);
    edit1 = new QLineEdit;
    vbox->addWidget(edit1);
    slider1 = new QSlider(Qt::Horizontal);
    vbox->addWidget(slider1);
    button2 = new QPushButton(tr("&Close"));
    vbox->addWidget(button2);
    resize(480, 320);
    connectSignals();
}

--\-- window.cpp (C++ CVS-1.3) --L1--Top-----
```

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C  Help

#ifndef BALLFIELD_H_
#define BALLFIELD_H_
#include <QtGui>
class BallField: public QWidget
{
public:
    BallField();
    virtual ~BallField();
protected:
    void mouseMoveEvent (QMouseEvent * event);
    void paintEvent (QPaintEvent * event);
    // slot not needed, use mouseMoveEvent
private:
    float ballX;
    float ballY;
    float ballVX;
    float ballVY; // (VX, VY) is a unit vector
    int ballSize;
    int paddleWidth;
    int paddleHeight;
    int paddleY;
};
#endif /*BALLFIELD_H_*/
--\-- BallField.h (C CVS:1.1)--L23--All-----
(No changes need to be saved)
```

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

#include "BallField.h"
#include <stdlib.h>
#include <math.h>
BallField::BallField() {
    ballX = width() / 2;
    ballY = height() / 2;
    ballSize = 5;
    paddleWidth = 5;
    paddleY = 0;
    do {
        ballVX = 0.001 * ((rand() % 1000) - 500);
    } while ((ballVX > -0.01) && (ballVX < 0.01));
    ballVY = sqrt(1 - ballVX * ballVX);
    paddleHeight = 10;
    setMouseTracking(true);
}

BallField::~~BallField() {
}

void BallField::mouseMoveEvent(QMouseEvent * event) {
    paddleY = event -> y();
    // keep the paddle inside
}

--\-- BallField.cpp      (C++ CVS-1.1)--L1--Top-----
Fontifying BallField.cpp... (regexps.....)
```

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

BallField::~~BallField() {
}

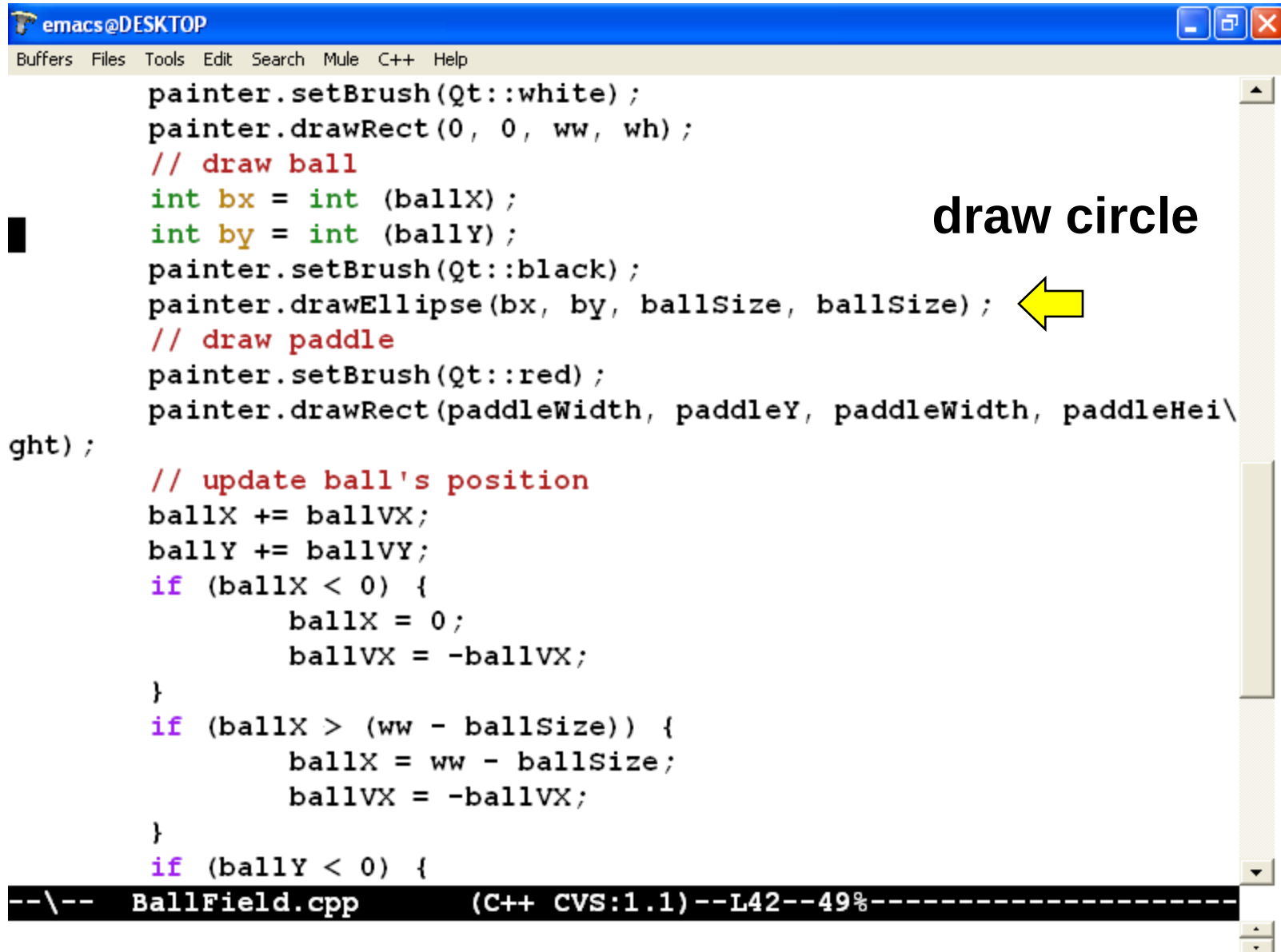
void BallField::mouseMoveEvent(QMouseEvent * event) {
    paddleY = event -> y();
    // keep the paddle inside
    if (paddleY < 0) {
        paddleY = 0;
    }
    if (paddleY > (height() - paddleHeight)) {
        paddleY = height() - paddleHeight;
    }
    update(); // tell qt to call paintEvent
}

void BallField::paintEvent(QPaintEvent * event) {
    int ww = width();
    int wh = height();
    // draw background and erase the previous scene
    QPainter painter(this);
    painter.setBrush(Qt::white);
    painter.drawRect(0, 0, ww, wh);
    // draw ball
}

--\-- BallField.cpp      (C++ CVS:1.1)--L23--22%-----
```



draw rectangle



```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

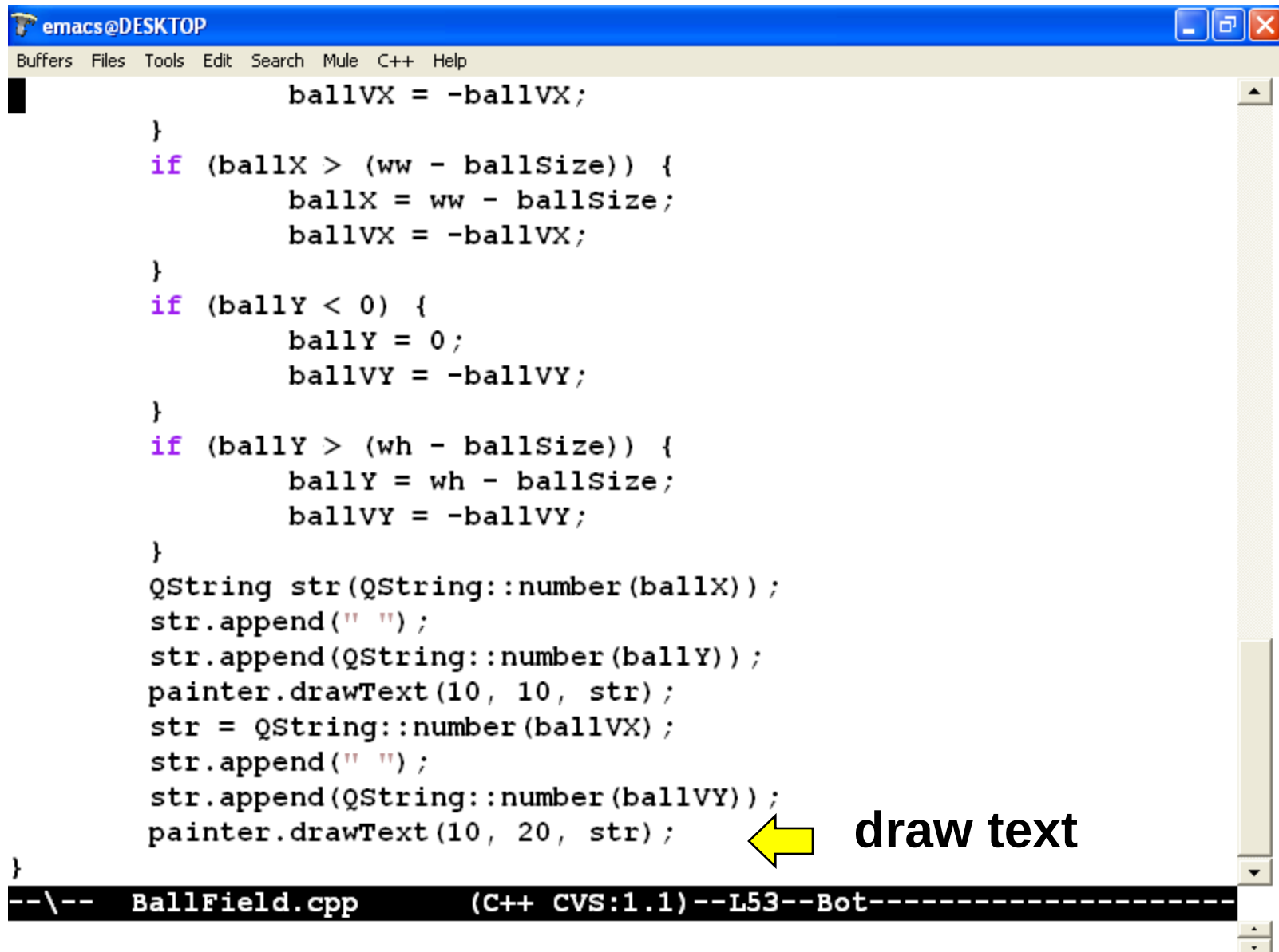
painter.setBrush(Qt::white);
painter.drawRect(0, 0, ww, wh);
// draw ball
int bx = int (ballX);
int by = int (ballY);
painter.setBrush(Qt::black);
painter.drawEllipse(bx, by, ballSize, ballSize);
// draw paddle
painter.setBrush(Qt::red);
painter.drawRect(paddleWidth, paddleY, paddleWidth, paddleHeight);

// update ball's position
ballX += ballVX;
ballY += ballVY;
if (ballX < 0) {
    ballX = 0;
    ballVX = -ballVX;
}
if (ballX > (ww - ballSize)) {
    ballX = ww - ballSize;
    ballVX = -ballVX;
}
if (ballY < 0) {
    ballY = 0;
    ballVY = -ballVY;
}

--\-- BallField.cpp      (C++ CVS:1.1)--L42--49%-----
```

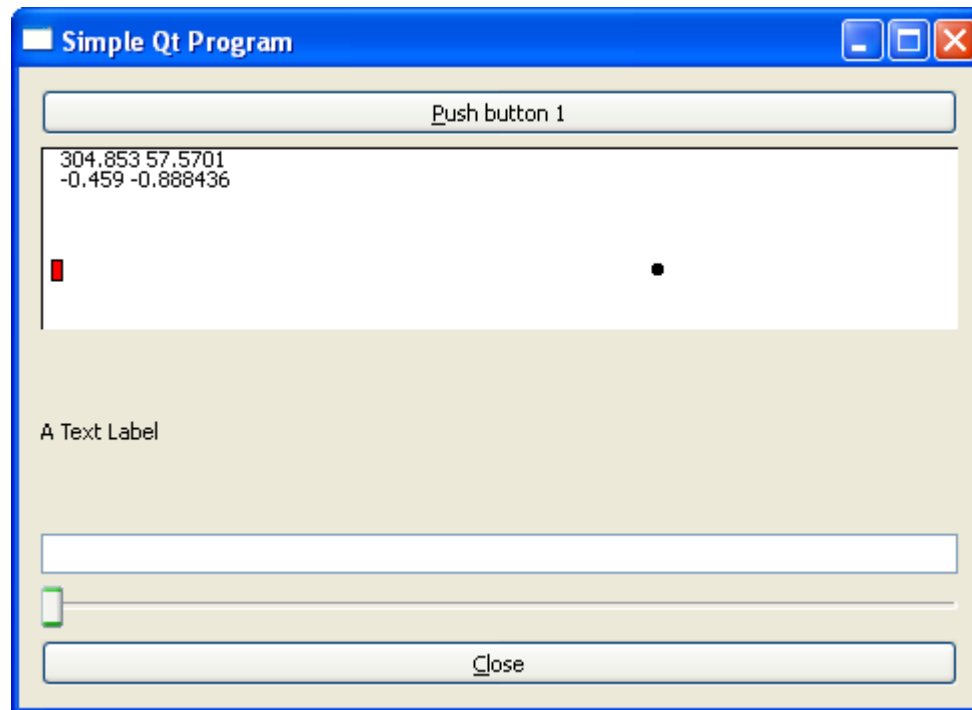
draw circle





```
ballVX = -ballVX;
}
if (ballX > (ww - ballSize)) {
    ballX = ww - ballSize;
    ballVX = -ballVX;
}
if (ballY < 0) {
    ballY = 0;
    ballVY = -ballVY;
}
if (ballY > (wh - ballSize)) {
    ballY = wh - ballSize;
    ballVY = -ballVY;
}
QString str(QString::number(ballX));
str.append(" ");
str.append(QString::number(ballY));
painter.drawText(10, 10, str);
str = QString::number(ballVX);
str.append(" ");
str.append(QString::number(ballVY));
painter.drawText(10, 20, str);
}

--\-- BallField.cpp (C++ CVS:1.1)--L53--Bot-----
```



Handle Mouse Event

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

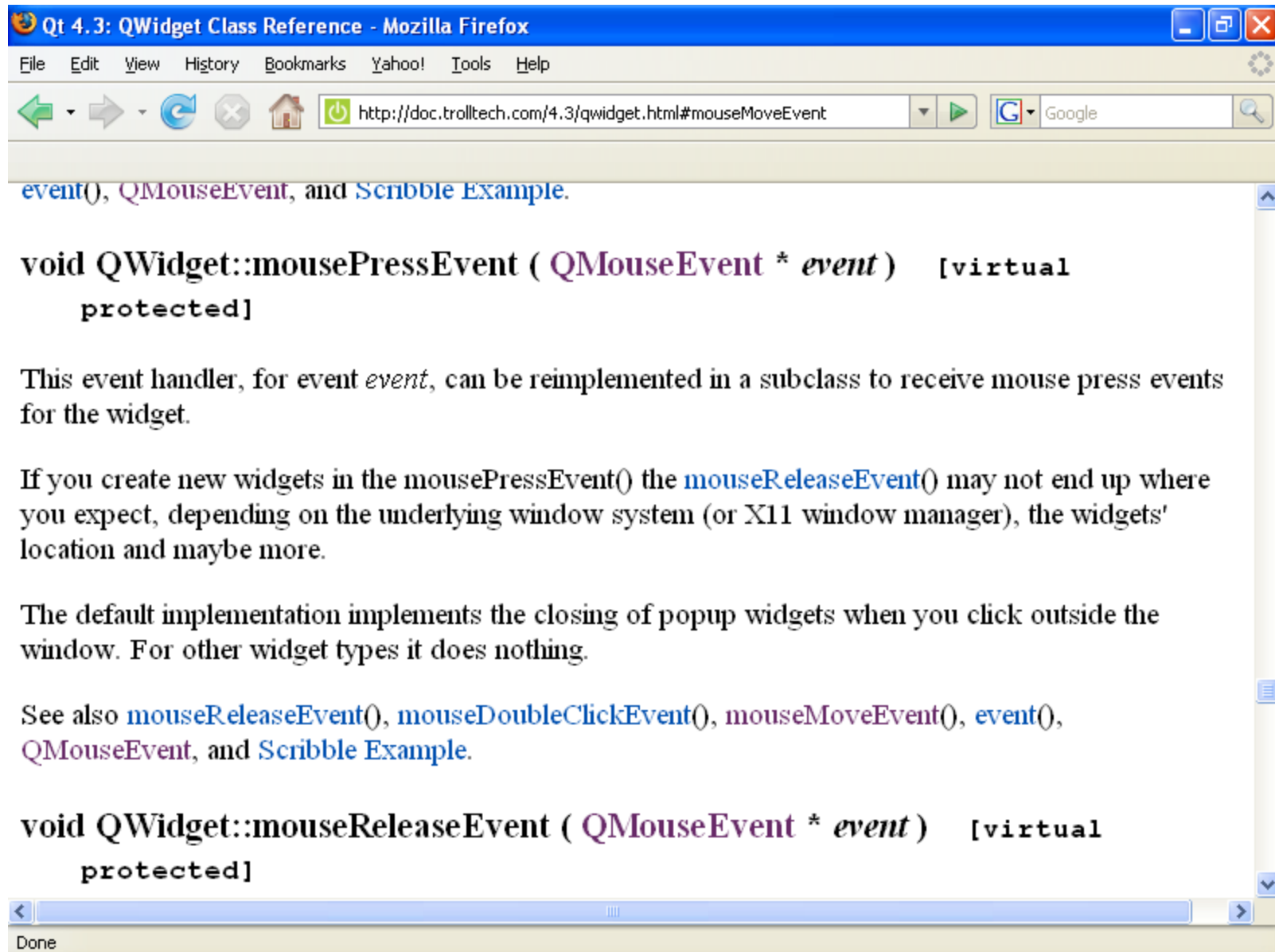
BallField::~BallField() {
}

void BallField::mouseMoveEvent(QMouseEvent * event) {
    paddleY = event -> y();
    // keep the paddle inside
    if (paddleY < 0) {
        paddleY = 0;
    }
    if (paddleY > (height() - paddleHeight)) {
        paddleY = height() - paddleHeight;
    }
    update(); // tell qt to call paintEvent
}

void BallField::paintEvent(QPaintEvent * event) {
    int ww = width();
    int wh = height();
    // draw background and erase the previous scene
    QPainter painter(this);
    painter.setBrush(Qt::white);
    painter.drawRect(0, 0, ww, wh);
    // draw ball
}

--\-- BallField.cpp      (C++ CVS:1.1)--L36--22%-----
```

called when mouse moves
does not have to use
connect function



```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

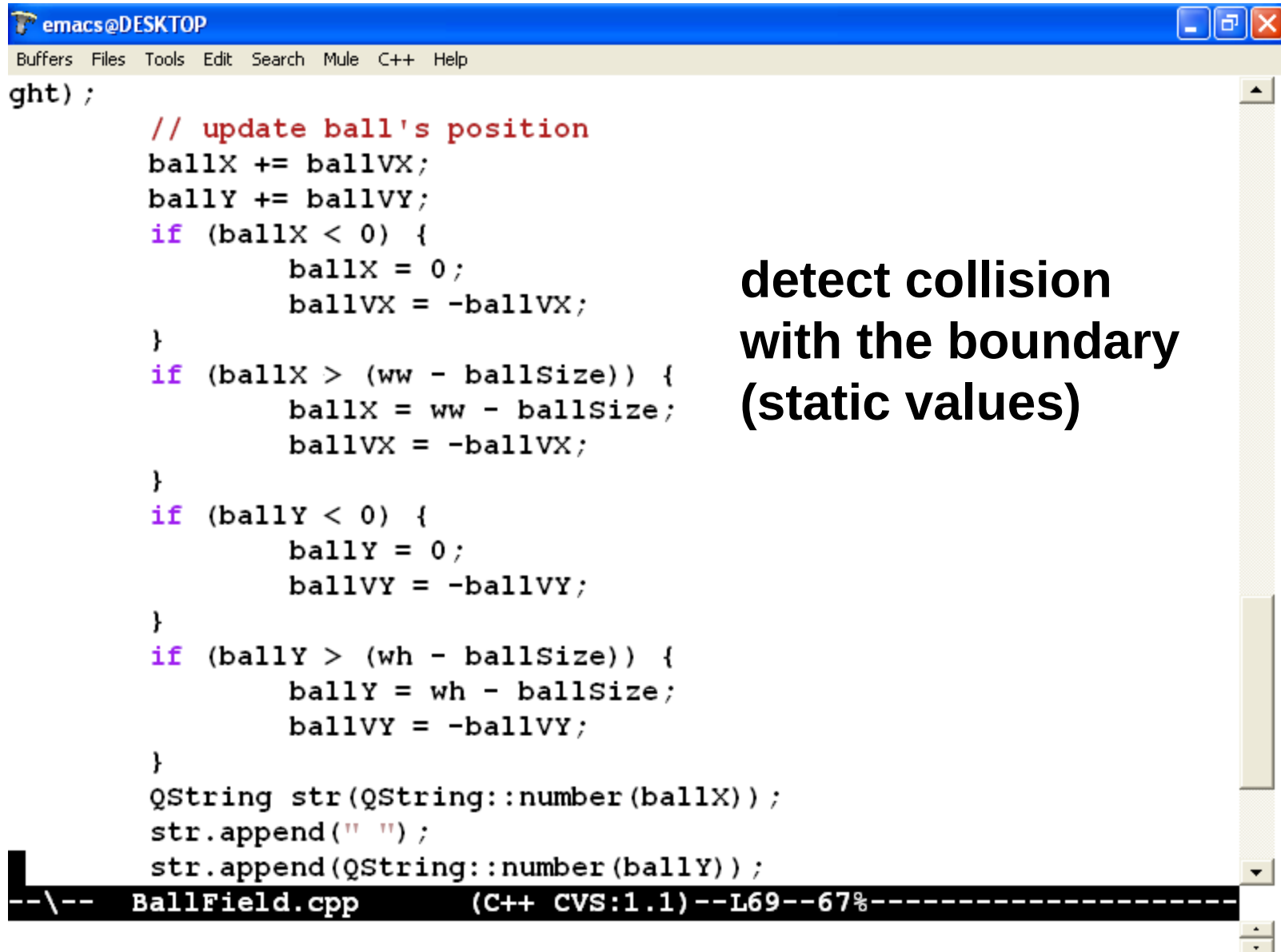
#include "BallField.h"
#include <stdlib.h>
#include <math.h>
BallField::BallField() {
    ballX = width() / 2;
    ballY = height() / 2;
    ballSize = 5;
    paddleWidth = 5;
    paddleY = 0;
    do {
        ballVX = 0.001 * ((rand() % 1000) - 500);
    } while ((ballVX > -0.01) && (ballVX < 0.01));
    ballVY = sqrt(1 - ballVX * ballVX);
    paddleHeight = 10;
    setMouseTracking(true);
}

BallField::~~BallField() {
}

void BallField::mouseMoveEvent(QMouseEvent * event) {
    paddleY = event -> y();
    // keep the paddle inside
}

--\-- BallField.cpp      (C++ CVS:1.1)--L23--Top-----
```

Collision Detection



```
emac@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

ght) ;

    // update ball's position
    ballX += ballVX;
    ballY += ballVY;
    if (ballX < 0) {
        ballX = 0;
        ballVX = -ballVX;
    }
    if (ballX > (ww - ballSize)) {
        ballX = ww - ballSize;
        ballVX = -ballVX;
    }
    if (ballY < 0) {
        ballY = 0;
        ballVY = -ballVY;
    }
    if (ballY > (wh - ballSize)) {
        ballY = wh - ballSize;
        ballVY = -ballVY;
    }
    QString str(QString::number(ballX));
    str.append(" ");
    str.append(QString::number(ballY));
--\-- BallField.cpp      (C++  CVS:1.1)--L69--67%-----
```

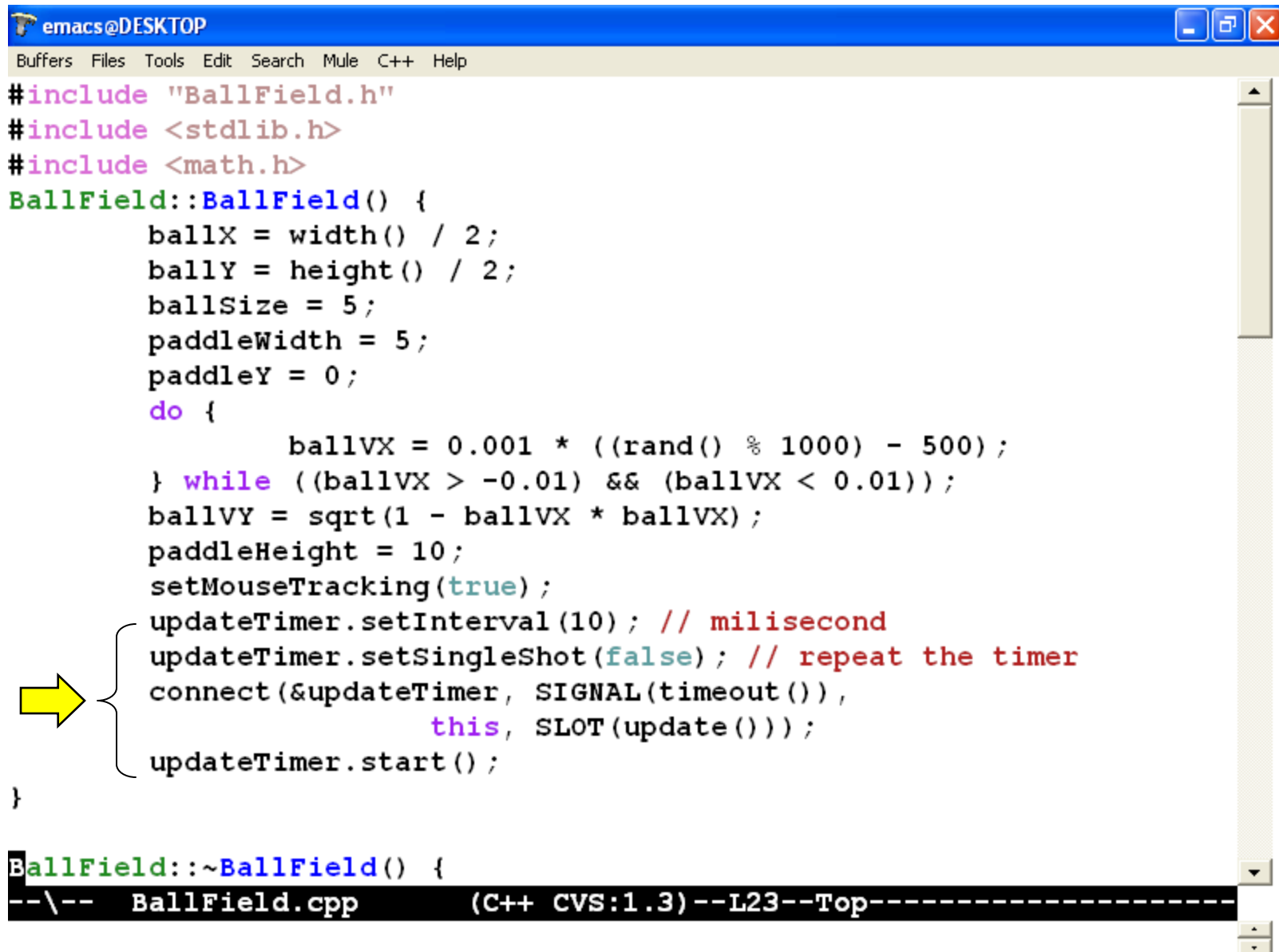
**detect collision
with the boundary
(static values)**

Timer Update

```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C  Help

{
    Q_OBJECT
public:
    BallField();
    virtual ~BallField();
protected:
    void mouseMoveEvent (QMouseEvent * event);
    void paintEvent (QPaintEvent * event);
    // slot not needed, use mouseMoveEvent
private:
    float ballX;
    float ballY;
    float ballVX;
    float ballVY; // (VX, VY) is a unit vector
    int ballSize;
    int paddleWidth;
    int paddleHeight;
    int paddleY;
    QTimer updateTimer;
};
#endif /*BALLFIELD_H_*/

--\-- BallField.h      (C CVS-1.3)--L13--Bot-----
```



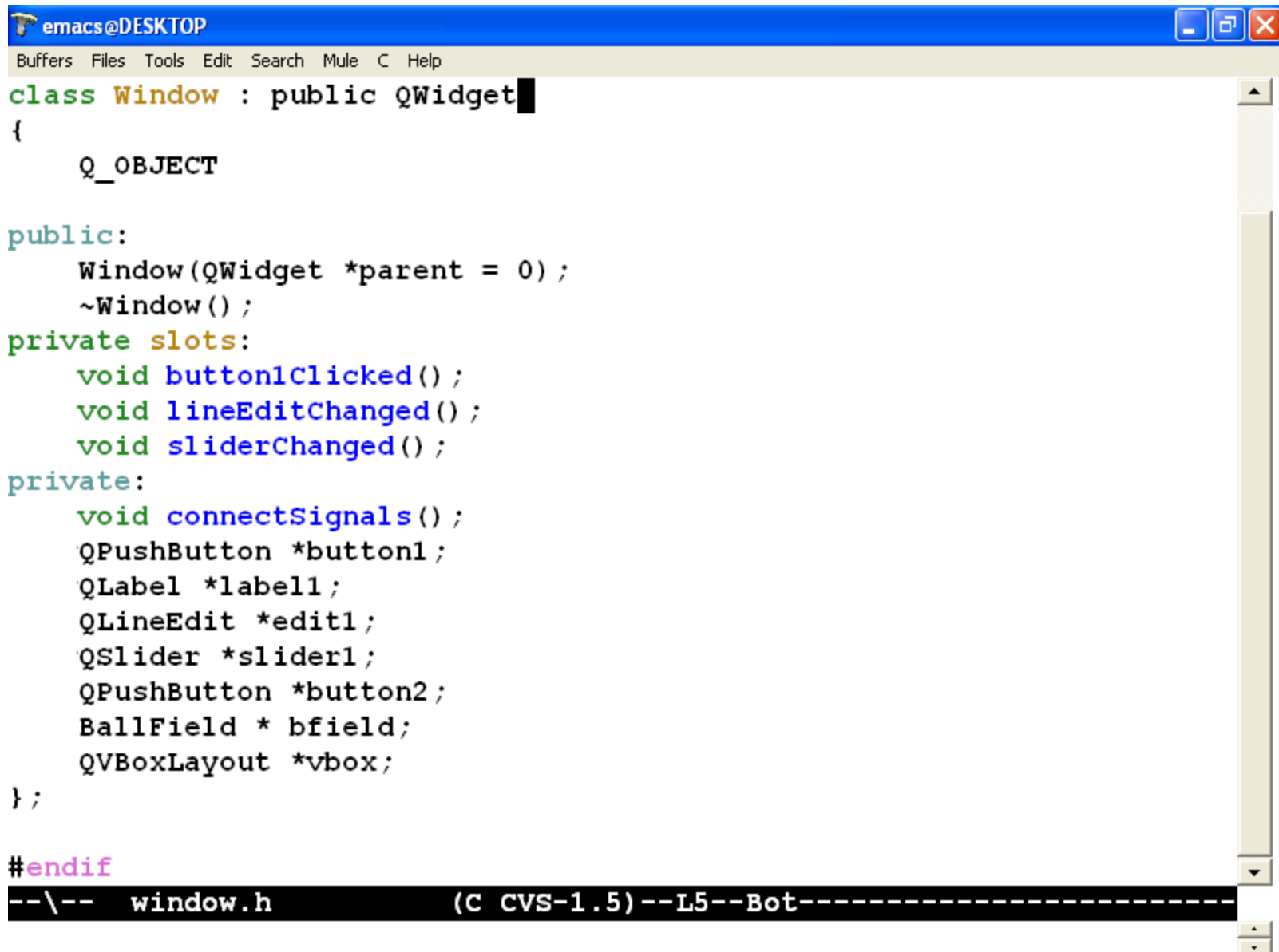
```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

#include "BallField.h"
#include <stdlib.h>
#include <math.h>
BallField::BallField() {
    ballX = width() / 2;
    ballY = height() / 2;
    ballSize = 5;
    paddleWidth = 5;
    paddleY = 0;
    do {
        ballVX = 0.001 * ((rand() % 1000) - 500);
    } while ((ballVX > -0.01) && (ballVX < 0.01));
    ballVY = sqrt(1 - ballVX * ballVX);
    paddleHeight = 10;
    setMouseTracking(true);
    updateTimer.setInterval(10); // milisecond
    updateTimer.setSingleShot(false); // repeat the timer
    connect(&updateTimer, SIGNAL(timeout()),
            this, SLOT(update()));
    updateTimer.start();
}

BallField::~~BallField() {
--\-- BallField.cpp      (C++ CVS:1.3)--L23--Top-----
```

Memory Management

What is created by **new** should be destroyed
by **delete**.



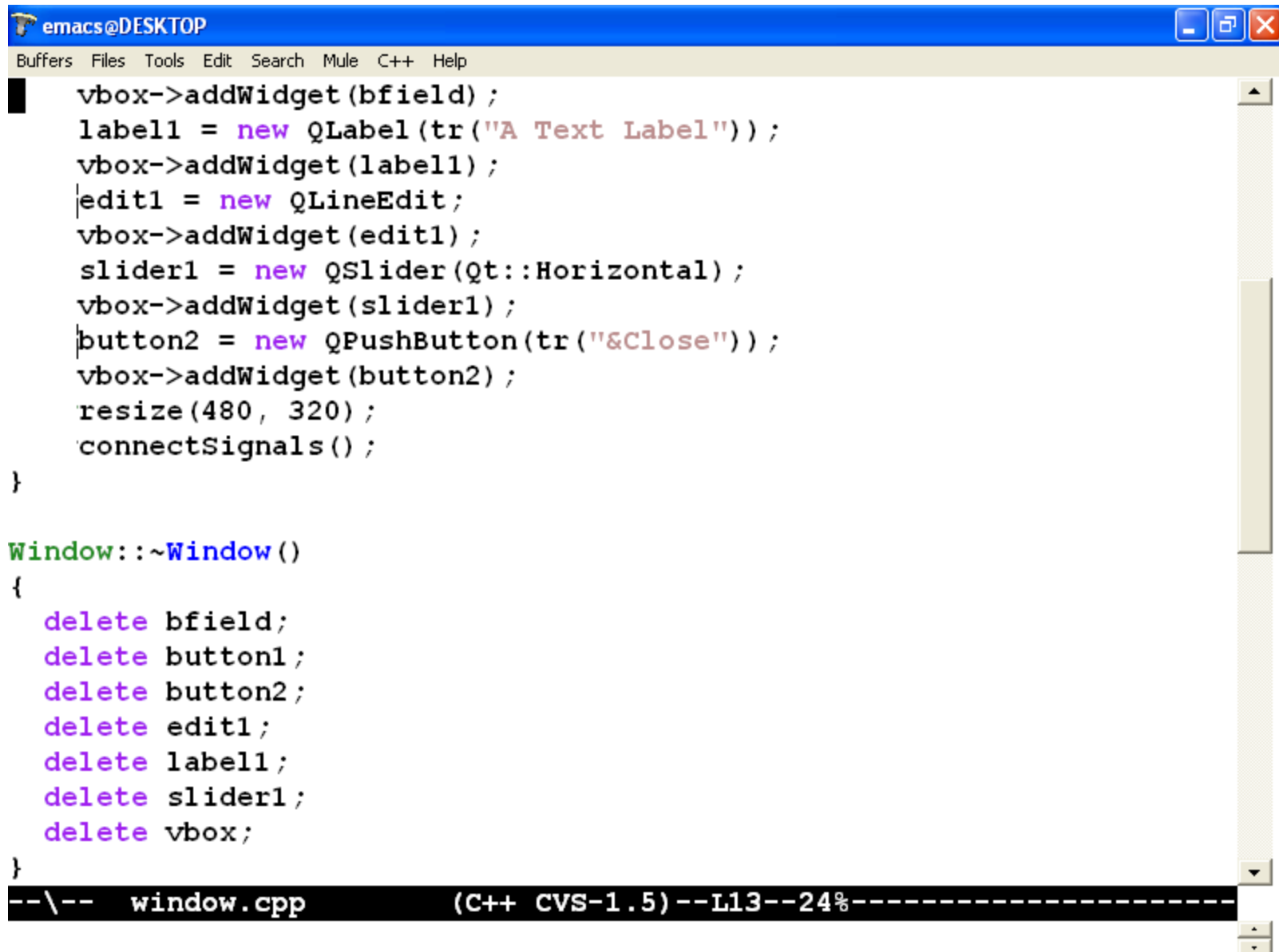
```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C  Help

class Window : public QWidget
{
    Q_OBJECT

public:
    Window(QWidget *parent = 0);
    ~Window();
private slots:
    void button1Clicked();
    void lineEditChanged();
    void sliderChanged();
private:
    void connectSignals();
    QPushButton *button1;
    QLabel *label1;
    QLineEdit *edit1;
    QSlider *slider1;
    QPushButton *button2;
    BallField * bfield;
    QVBoxLayout *vbox;
};

#endif

--\-- window.h      (C CVS-1.5)--L5--Bot-----
```



```
emacs@DESKTOP
Buffers  Files  Tools  Edit  Search  Mule  C++  Help

vbox->addWidget(bfield);
label1 = new QLabel(tr("A Text Label"));
vbox->addWidget(label1);
edit1 = new QLineEdit;
vbox->addWidget(edit1);
slider1 = new QSlider(Qt::Horizontal);
vbox->addWidget(slider1);
button2 = new QPushButton(tr("&Close"));
vbox->addWidget(button2);
resize(480, 320);
connectSignals();
}

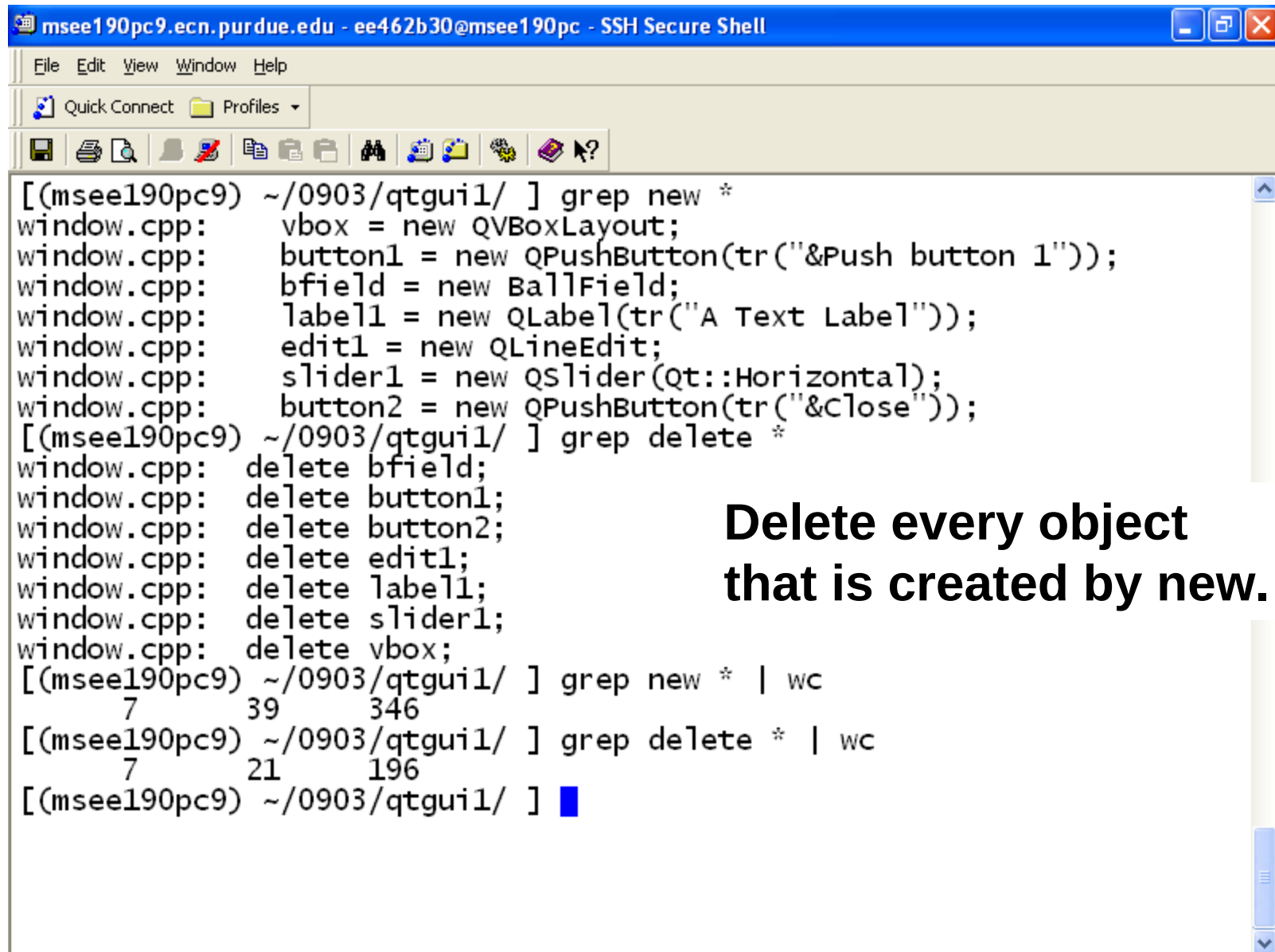
Window::~Window()
{
    delete bfield;
    delete button1;
    delete button2;
    delete edit1;
    delete label1;
    delete slider1;
    delete vbox;
}

--\--  window.cpp  (C++  CVS-1.5) --L13--24%-----
```

Qt and Valgrind

Valgrind and Qt do not work well together.

Fortunately, we can follow a simple rule for memory management.



The screenshot shows a terminal window titled "msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell". The window has a menu bar (File, Edit, View, Window, Help) and a toolbar with icons for file operations. The terminal content is as follows:

```
[(msee190pc9) ~/0903/qtgui1/ ] grep new *
window.cpp:      vbox = new QVBoxLayout;
window.cpp:      button1 = new QPushButton(tr("&Push button 1"));
window.cpp:      bfield = new BallField;
window.cpp:      label1 = new QLabel(tr("A Text Label"));
window.cpp:      edit1 = new QLineEdit;
window.cpp:      slider1 = new QSlider(Qt::Horizontal);
window.cpp:      button2 = new QPushButton(tr("&Close"));
[(msee190pc9) ~/0903/qtgui1/ ] grep delete *
window.cpp:      delete bfield;
window.cpp:      delete button1;
window.cpp:      delete button2;
window.cpp:      delete edit1;
window.cpp:      delete label1;
window.cpp:      delete slider1;
window.cpp:      delete vbox;
[(msee190pc9) ~/0903/qtgui1/ ] grep new * | wc
      7      39      346
[(msee190pc9) ~/0903/qtgui1/ ] grep delete * | wc
      7      21      196
[(msee190pc9) ~/0903/qtgui1/ ] █
```

**Delete every object
that is created by new.**

Self Test

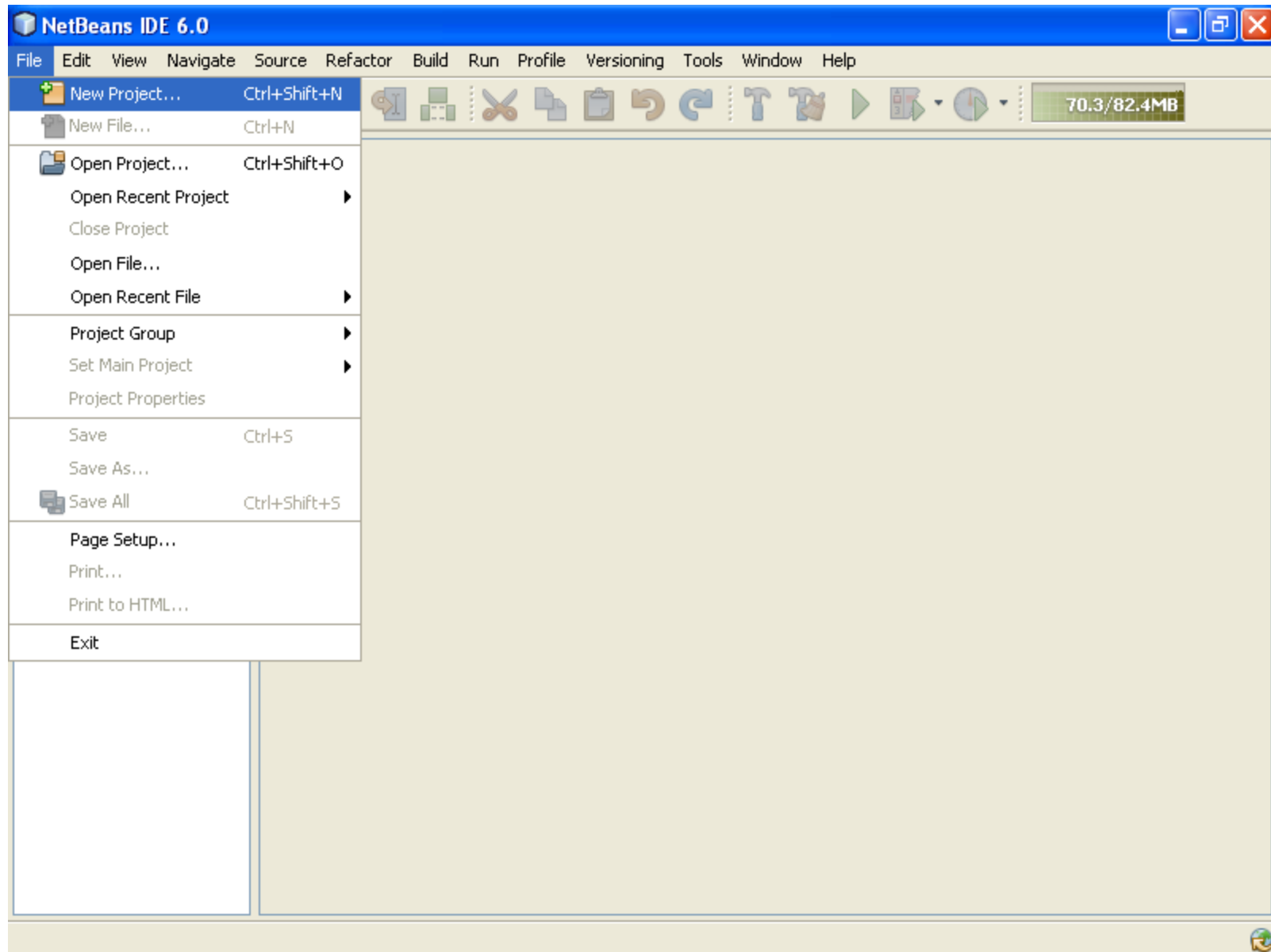
ECE 462

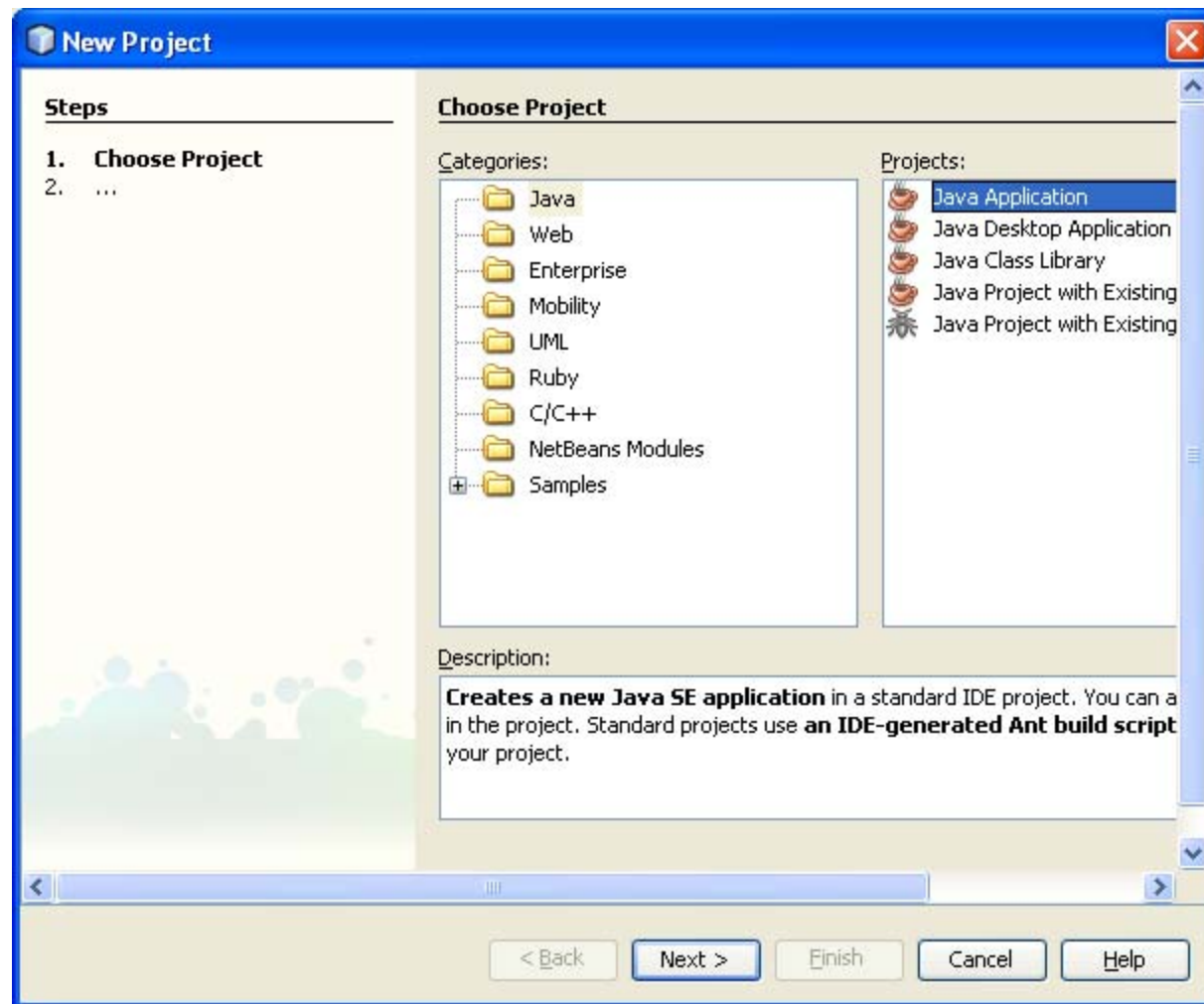
Object-Oriented Programming

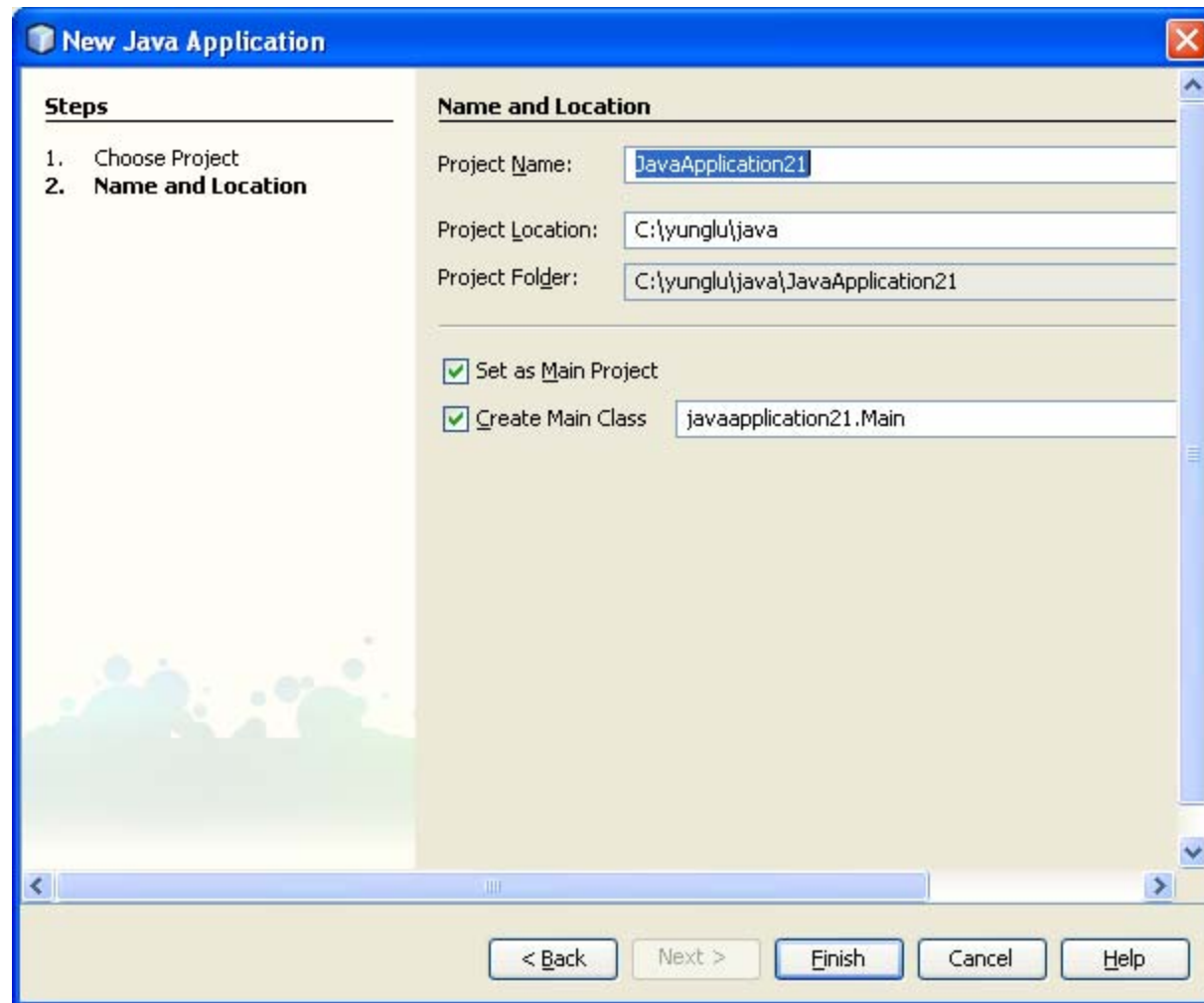
using C++ and Java

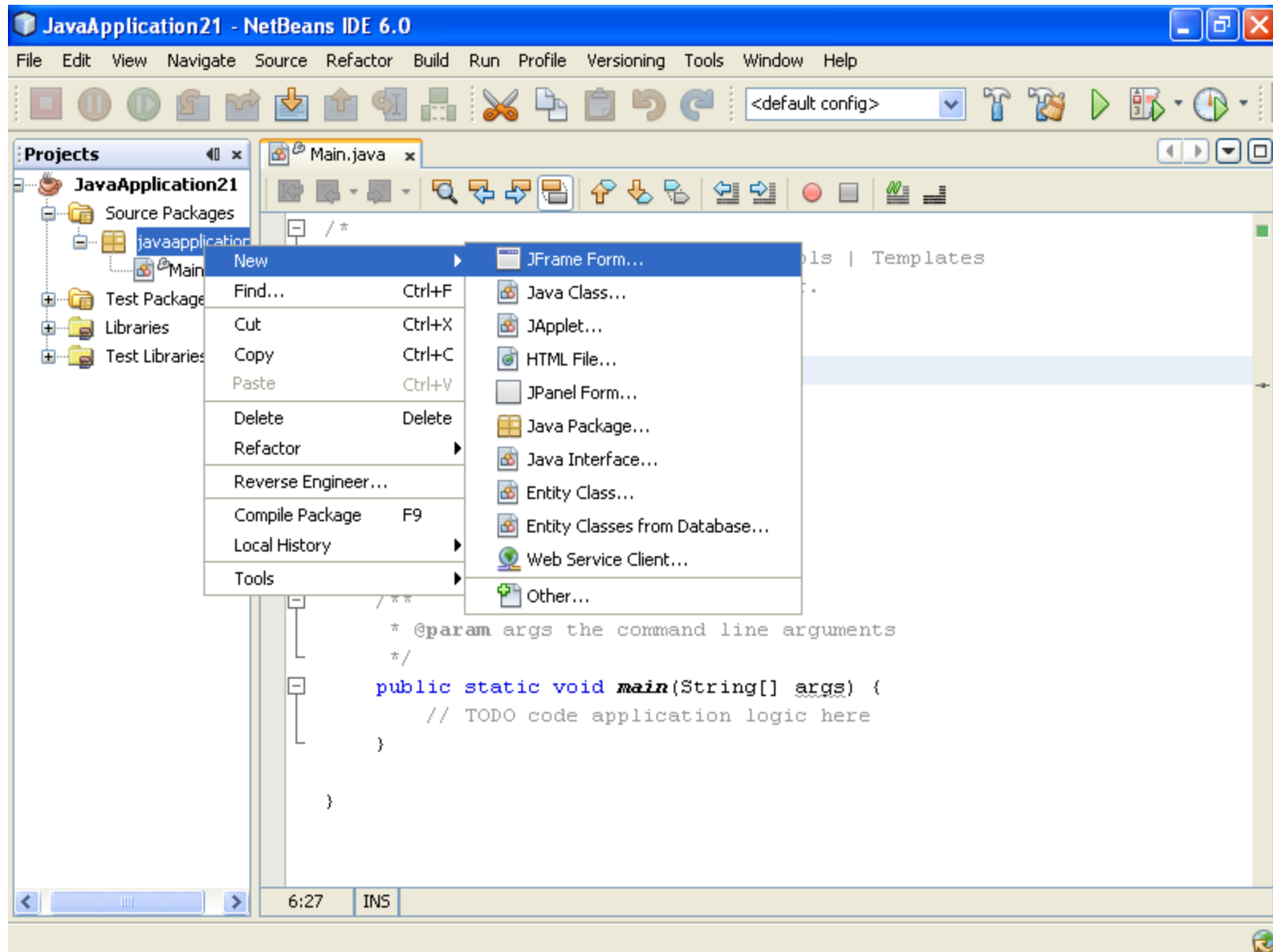
Java Events and Handlers

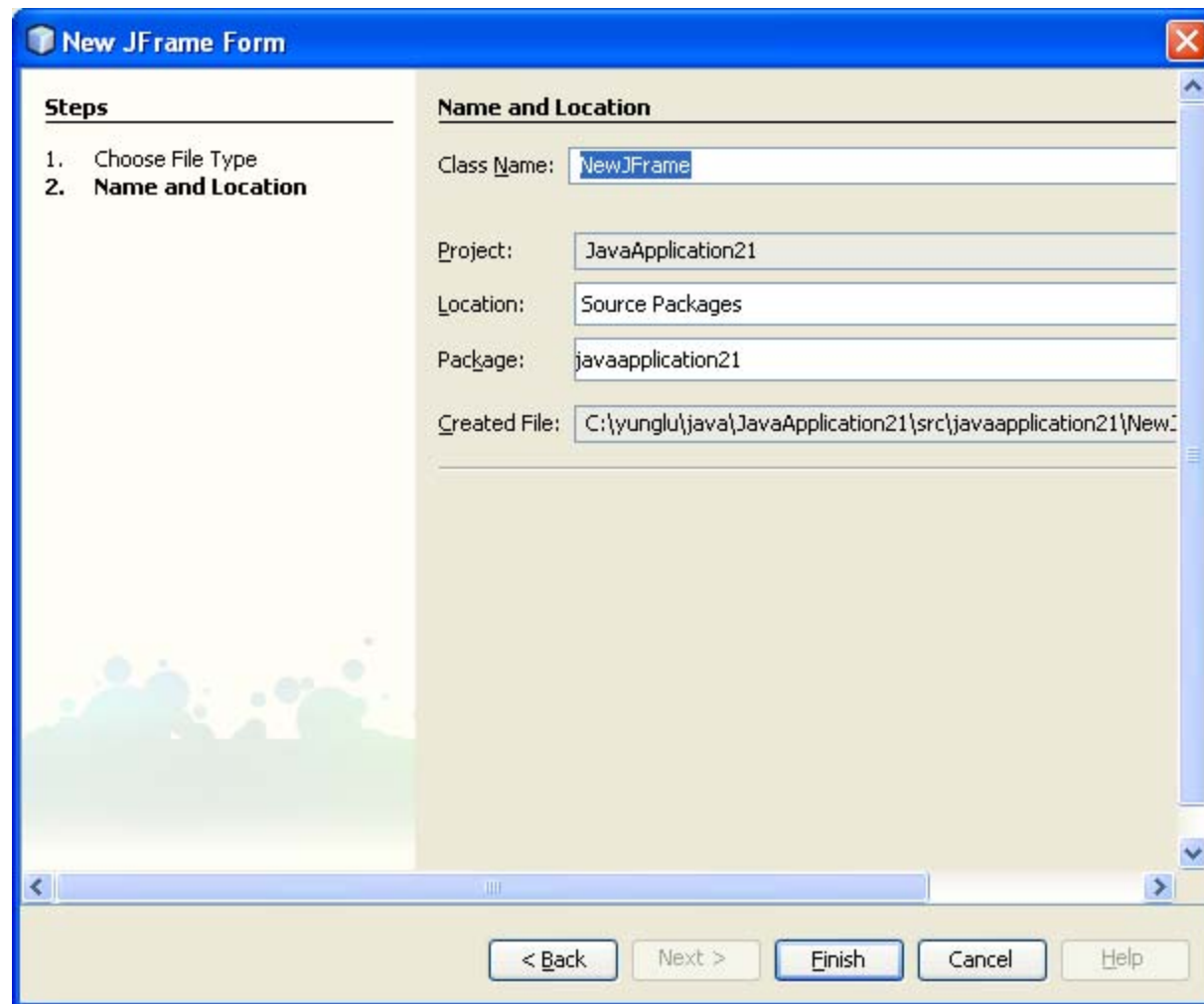
Yung-Hsiang Lu
yunглу@purdue.edu











The image shows a 'New JFrame Form' dialog box with a blue title bar and a close button. It is divided into two main sections: 'Steps' on the left and 'Name and Location' on the right. The 'Steps' section lists two steps: '1. Choose File Type' and '2. Name and Location', with the second step being the active one. The 'Name and Location' section contains several text input fields: 'Class Name' (containing 'NewJFrame'), 'Project' (containing 'JavaApplication21'), 'Location' (containing 'Source Packages'), 'Package' (containing 'javaapplication21'), and 'Created File' (containing a full path). At the bottom, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

New JFrame Form

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

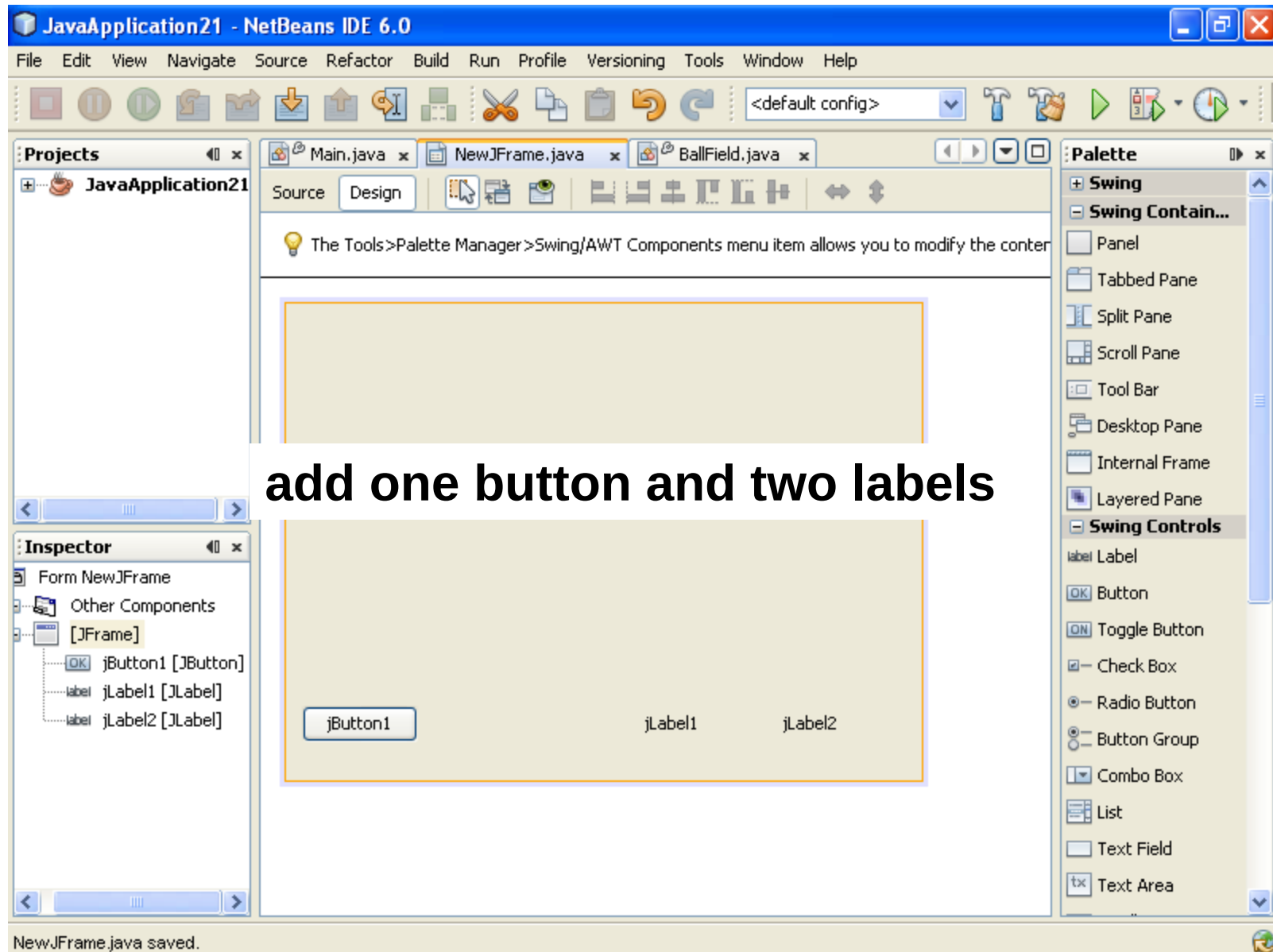
Project:

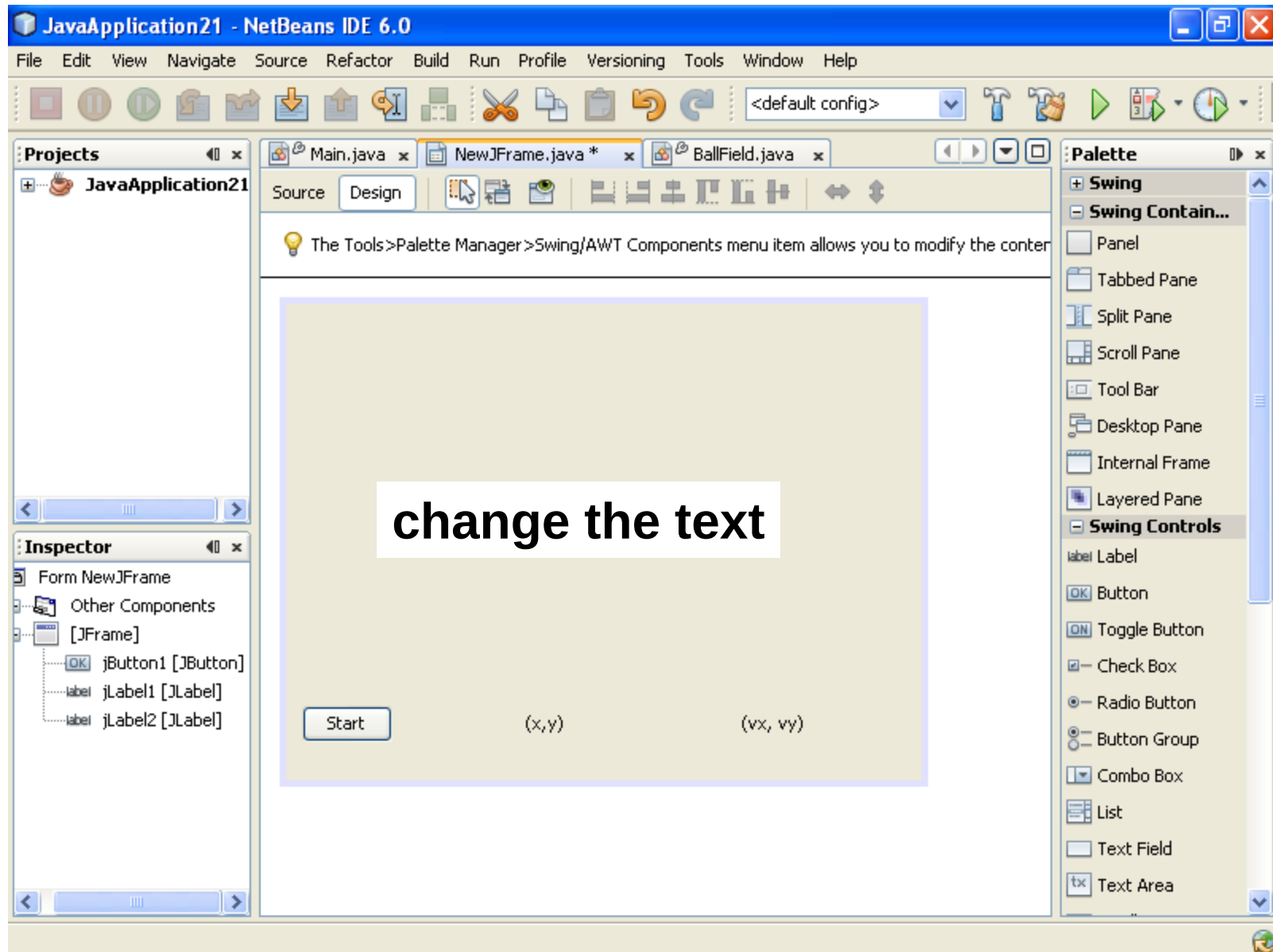
Location:

Package:

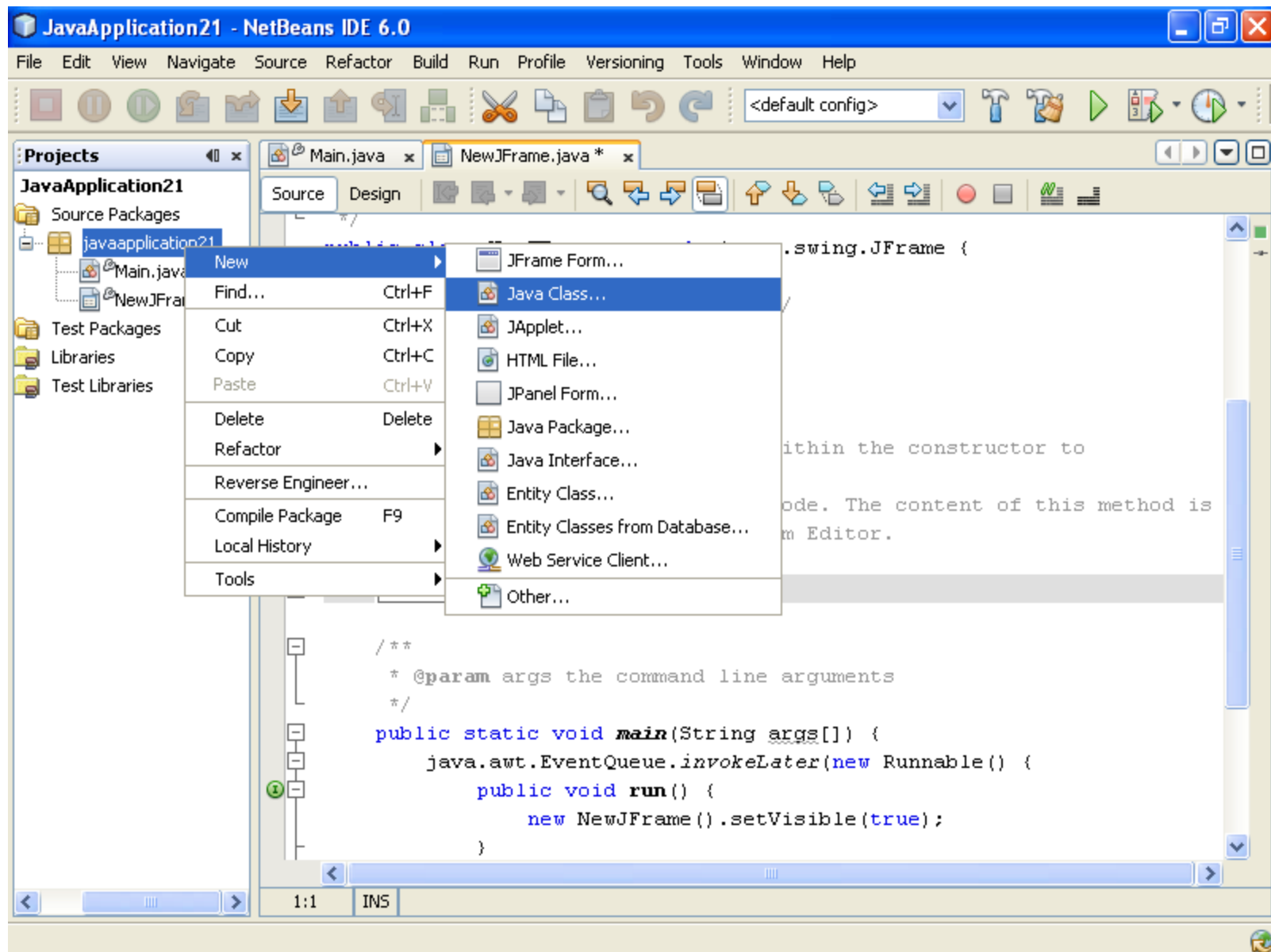
Created File:

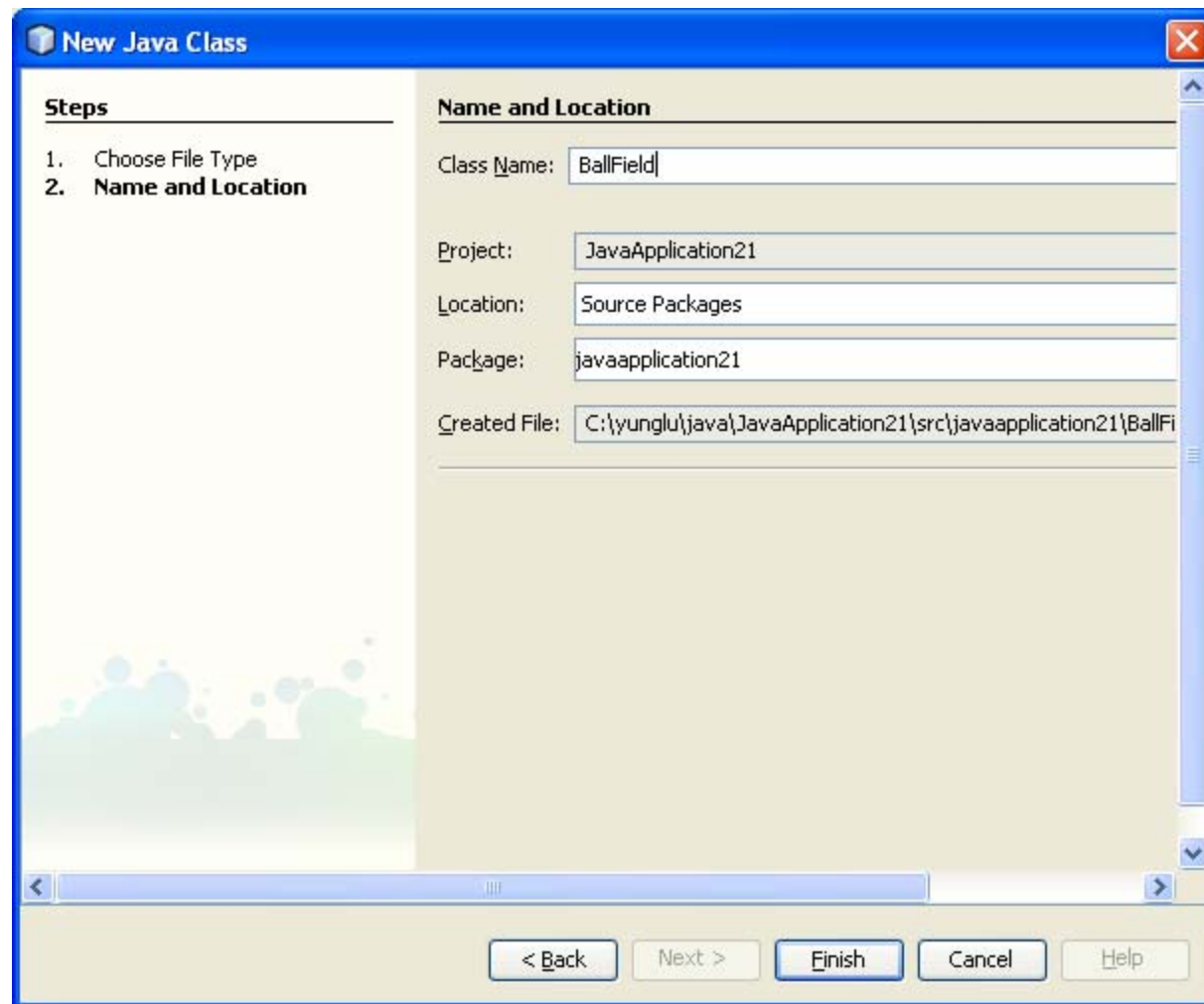
< Back Next > Finish Cancel Help

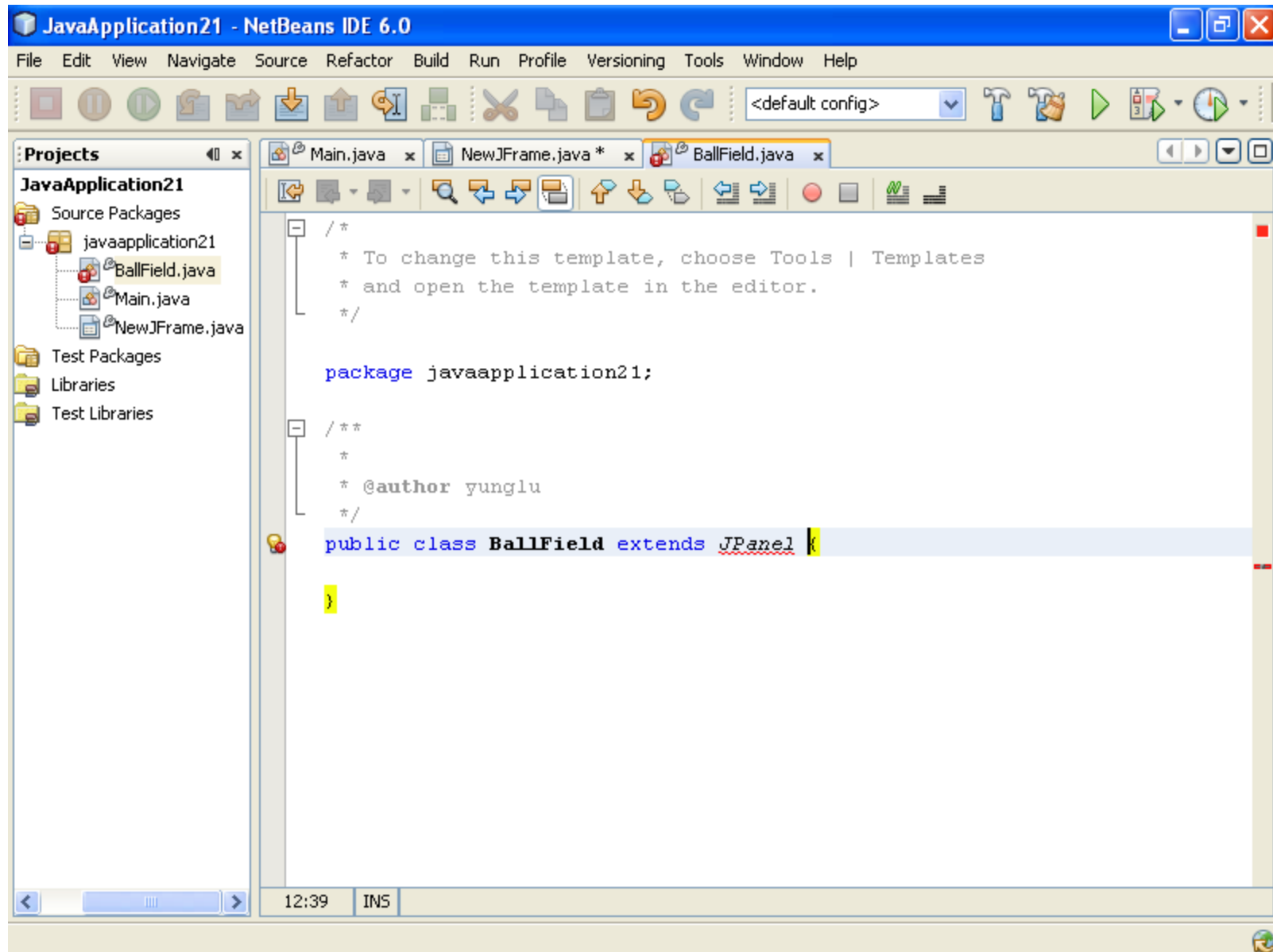


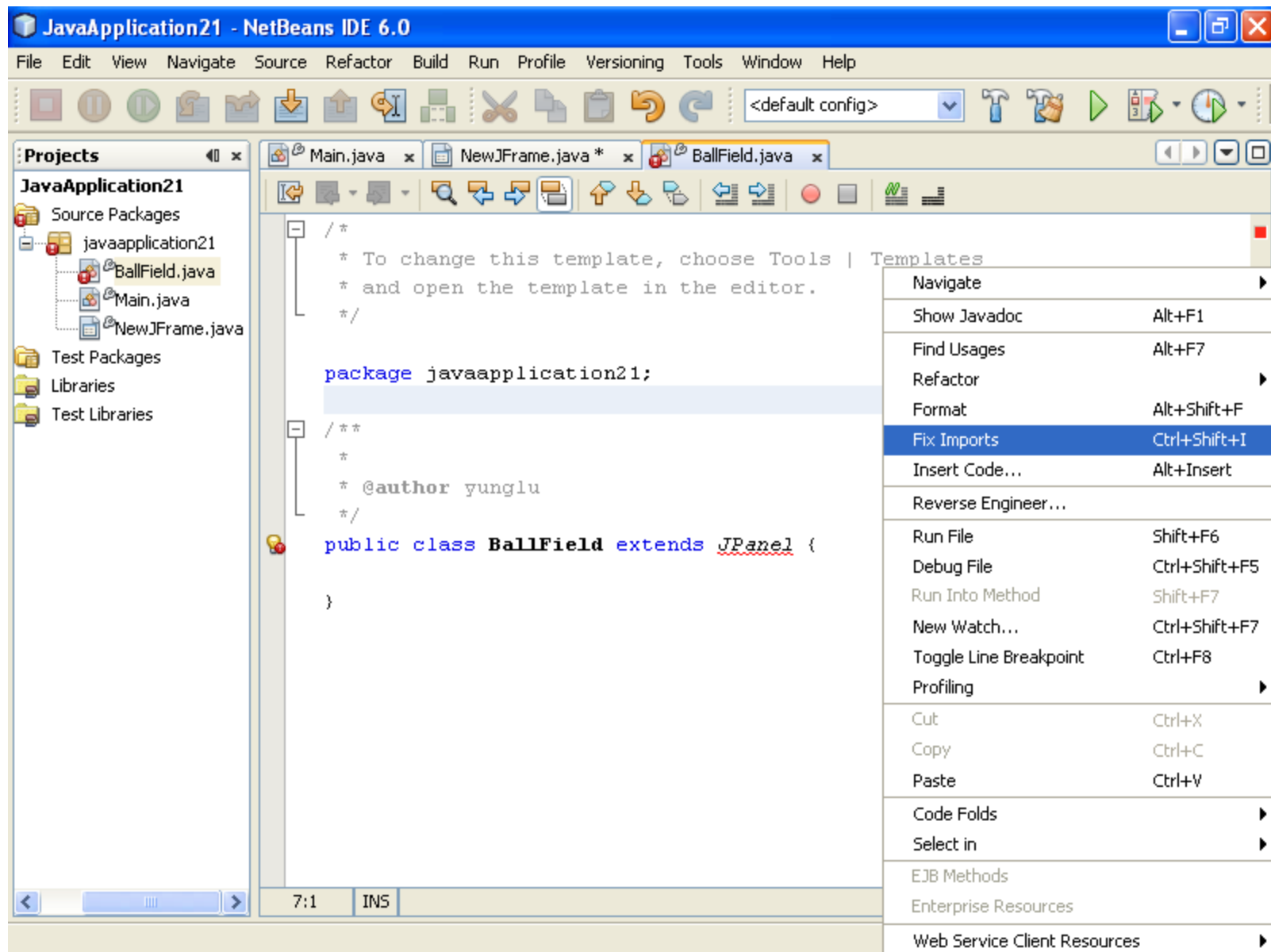


Add a Derived Class of JPanel

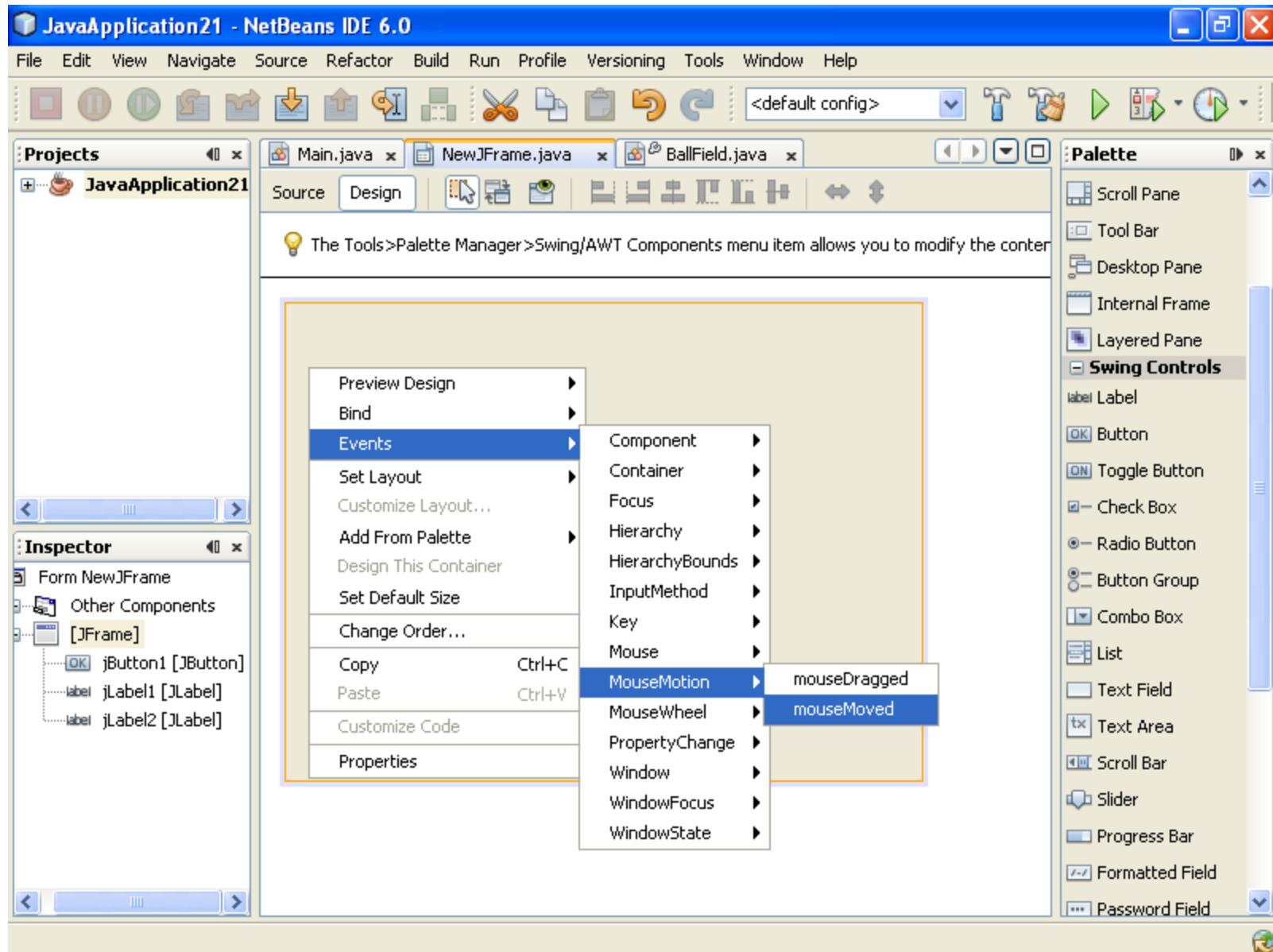


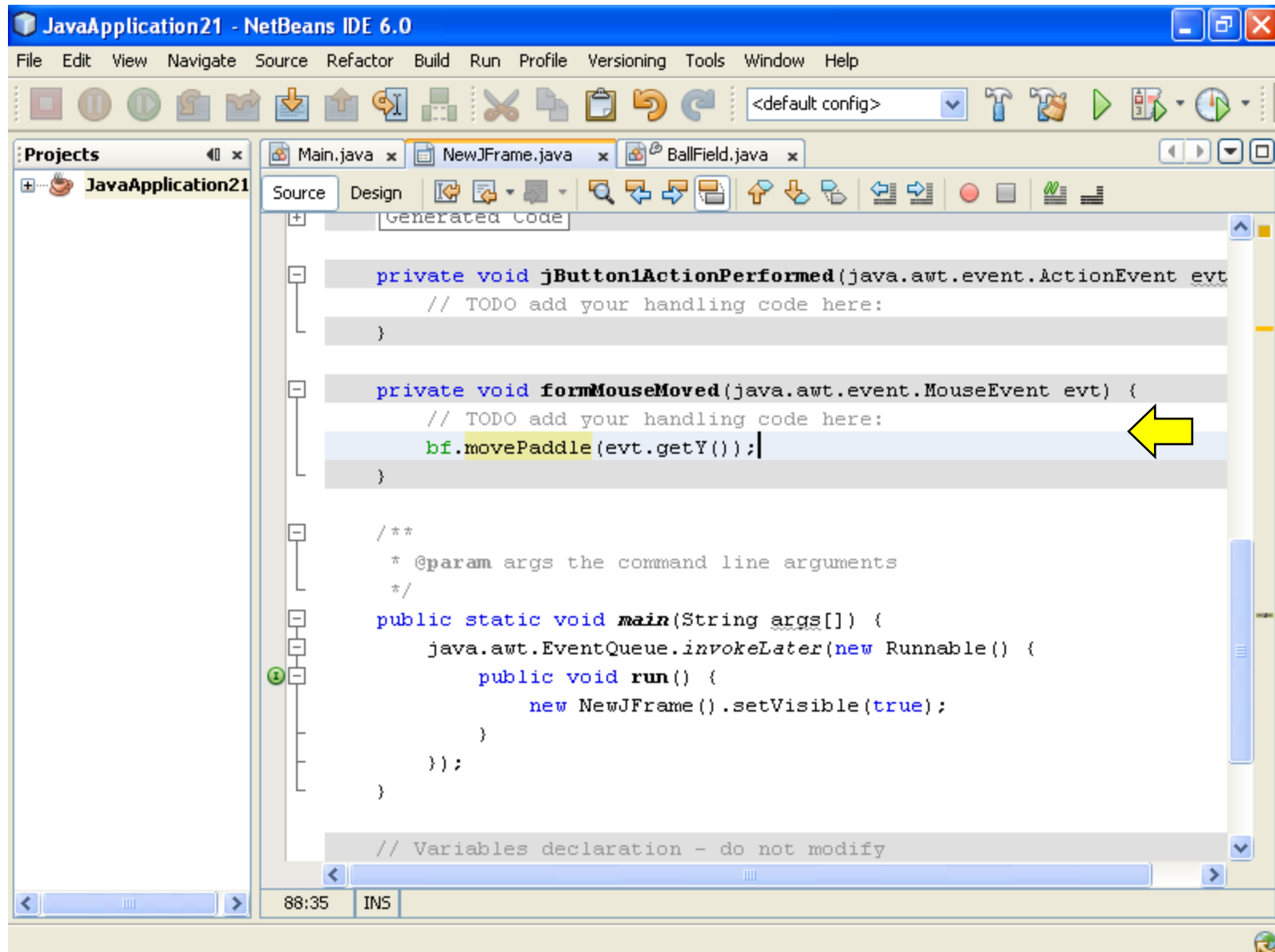


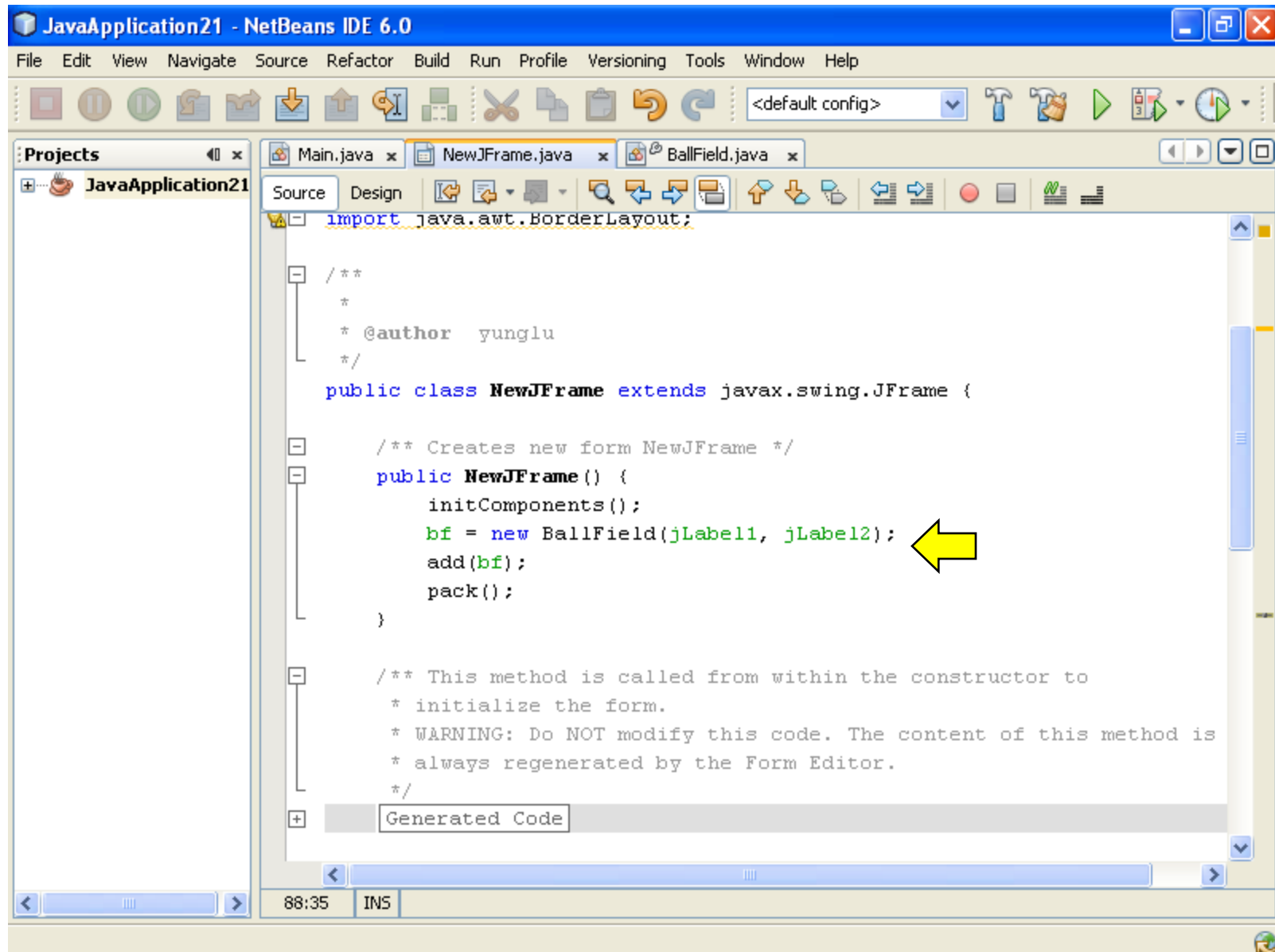


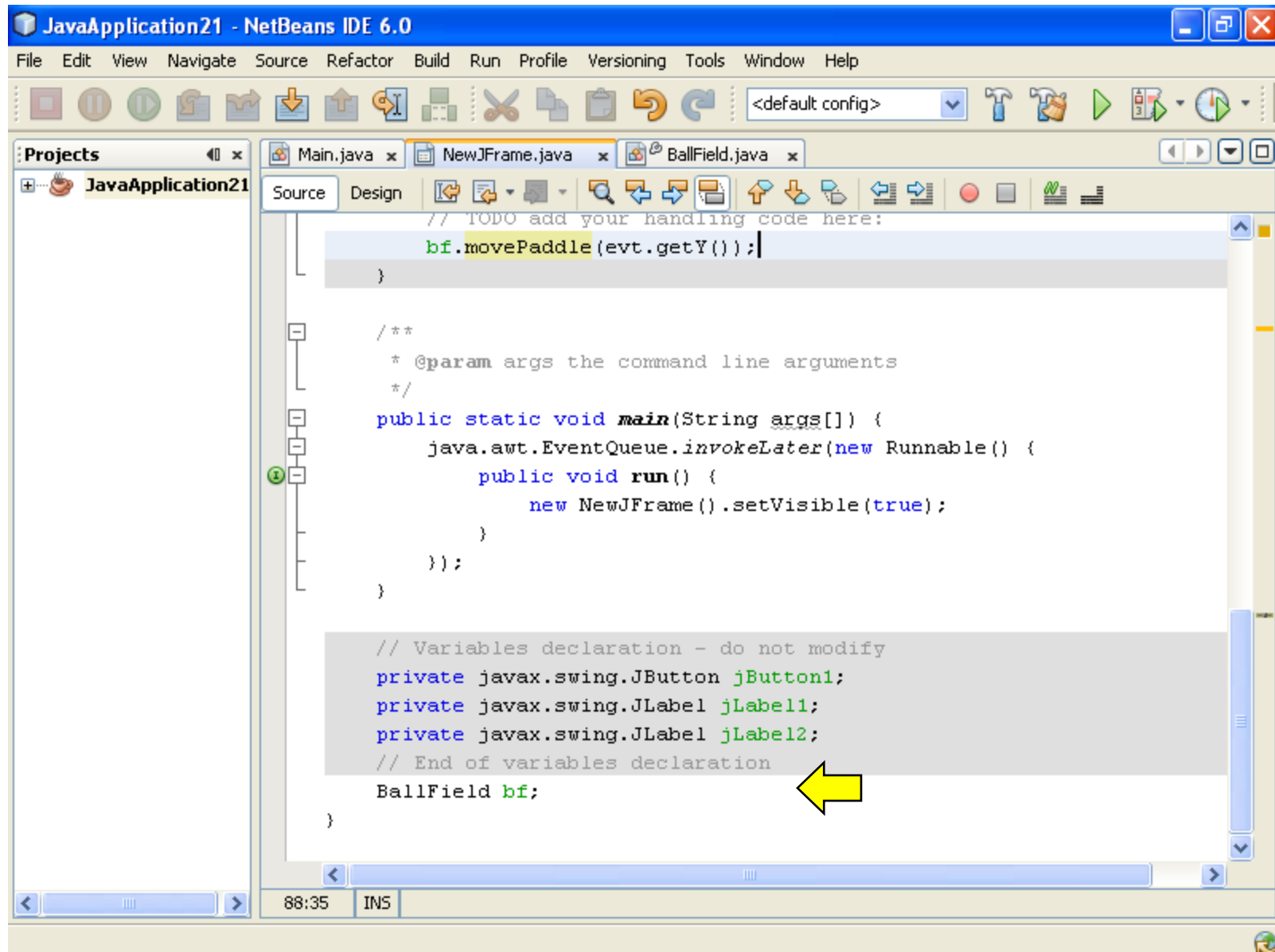


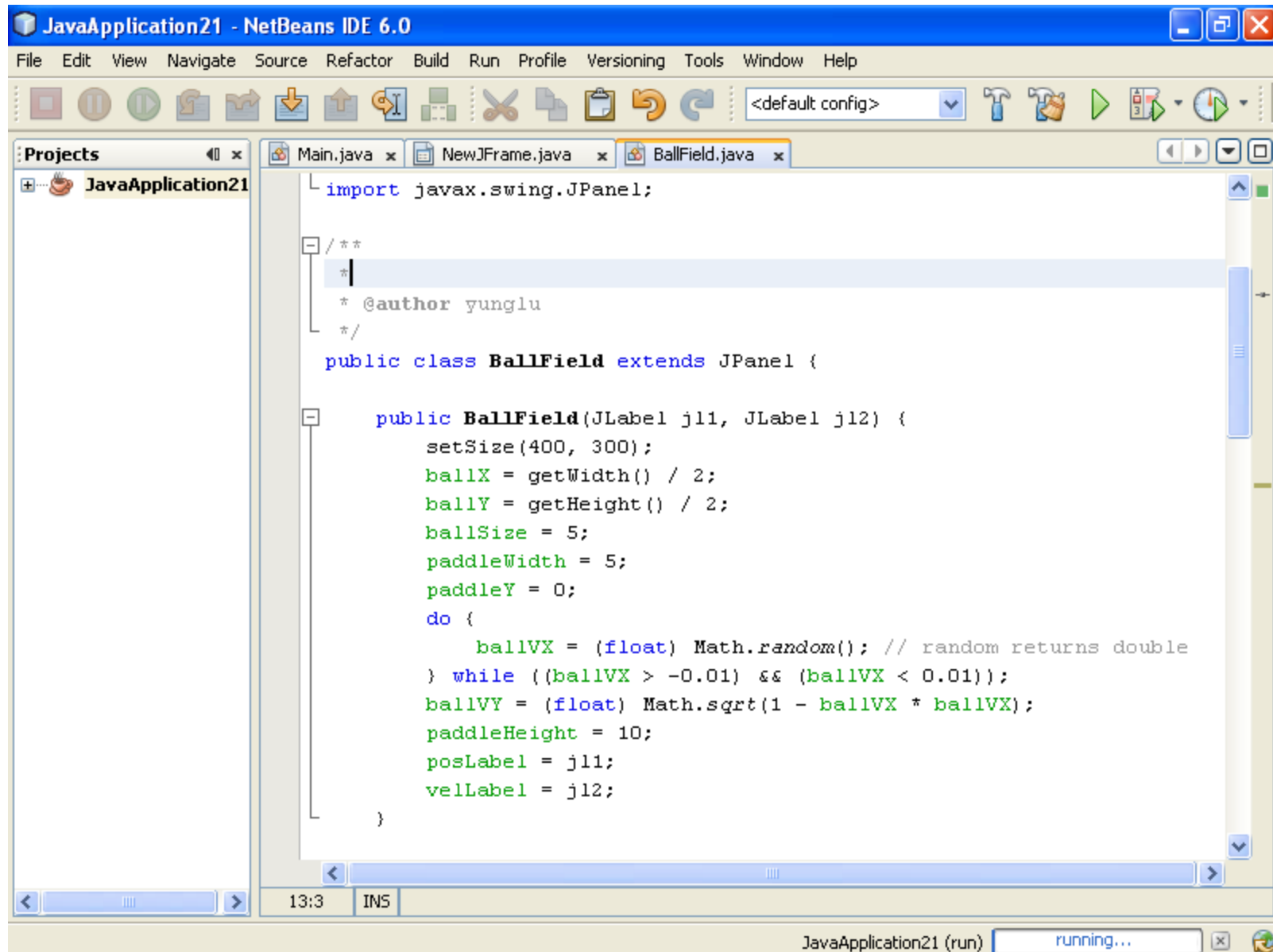
Handle Mouse Event

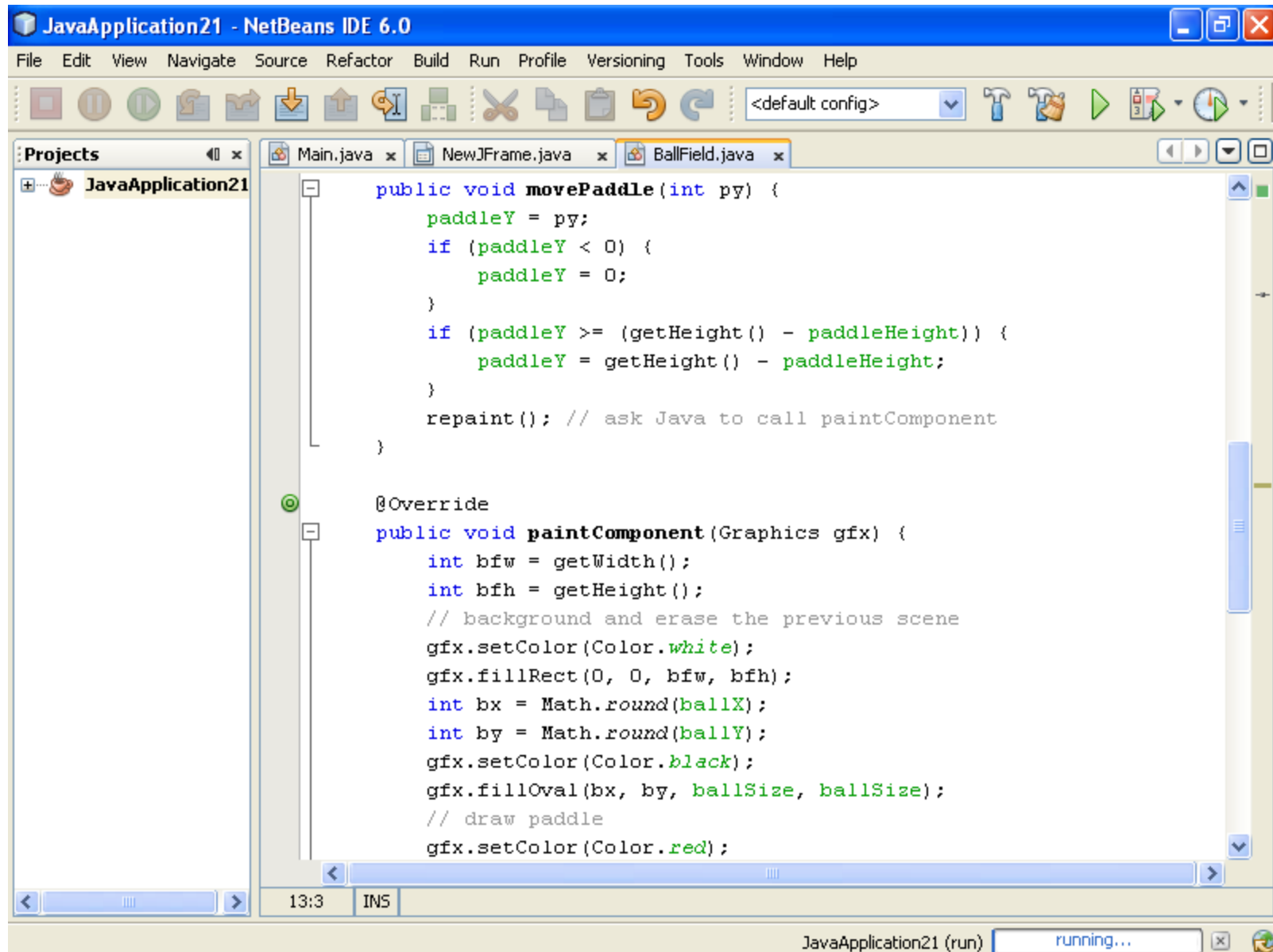


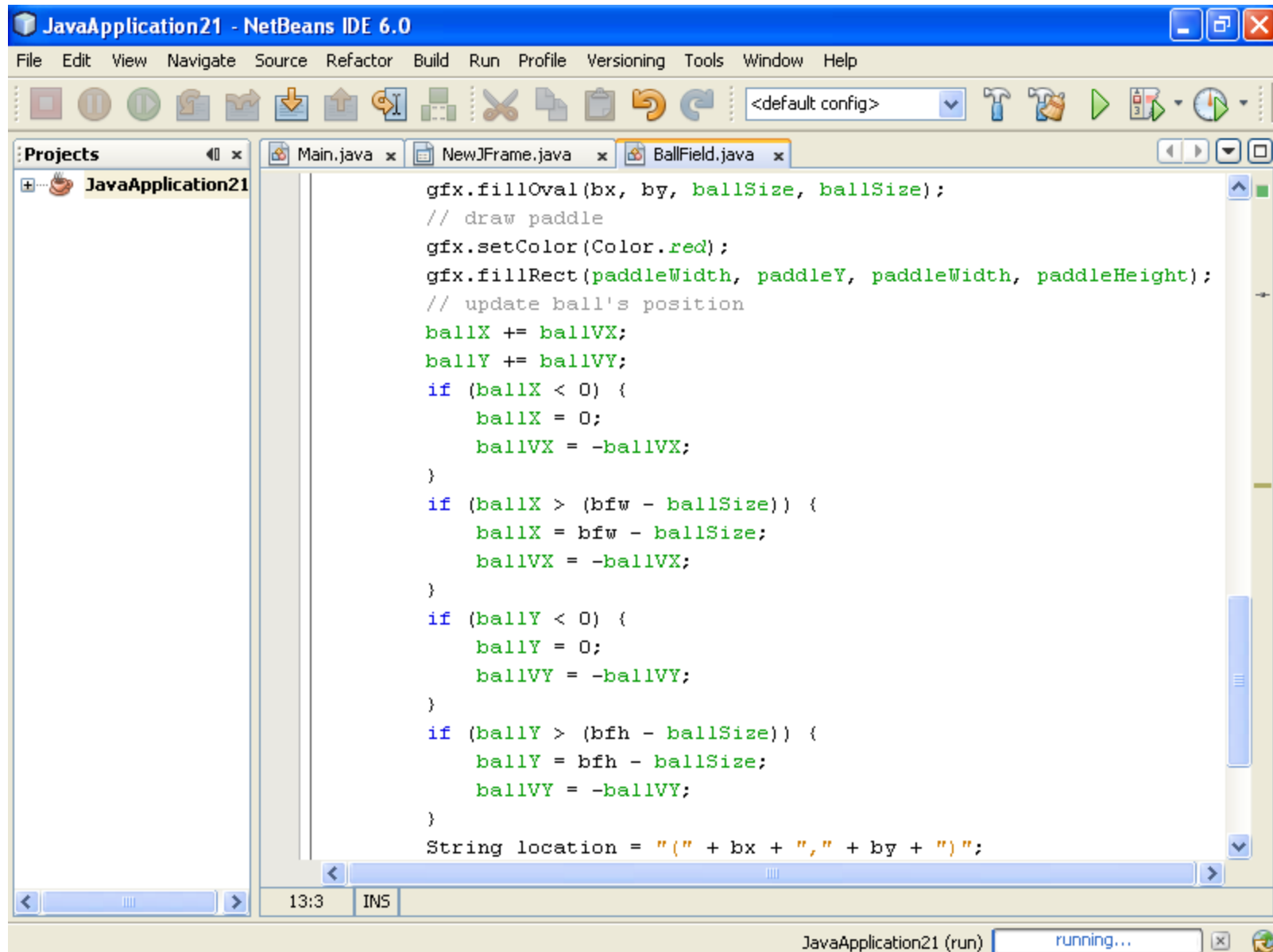


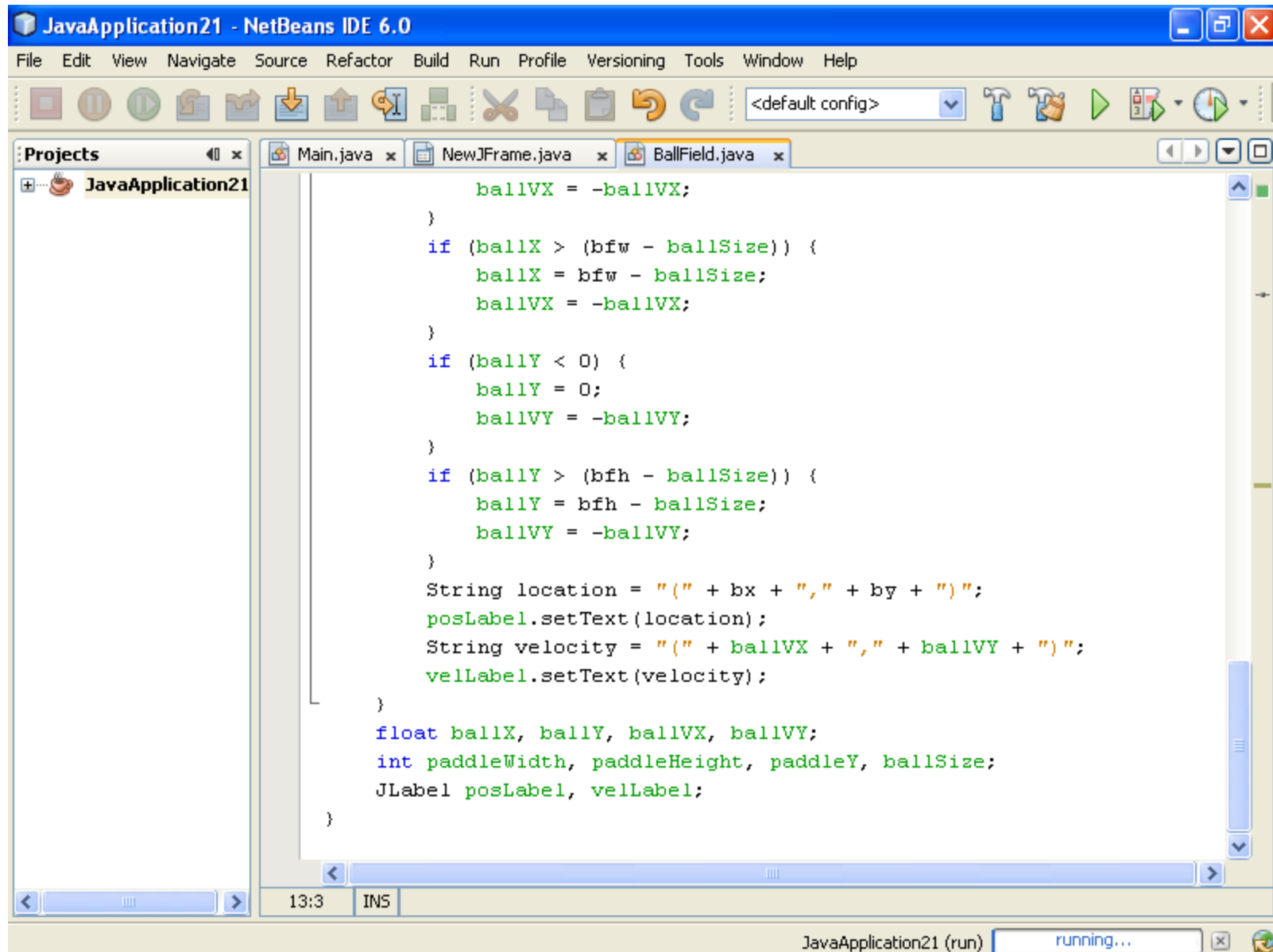


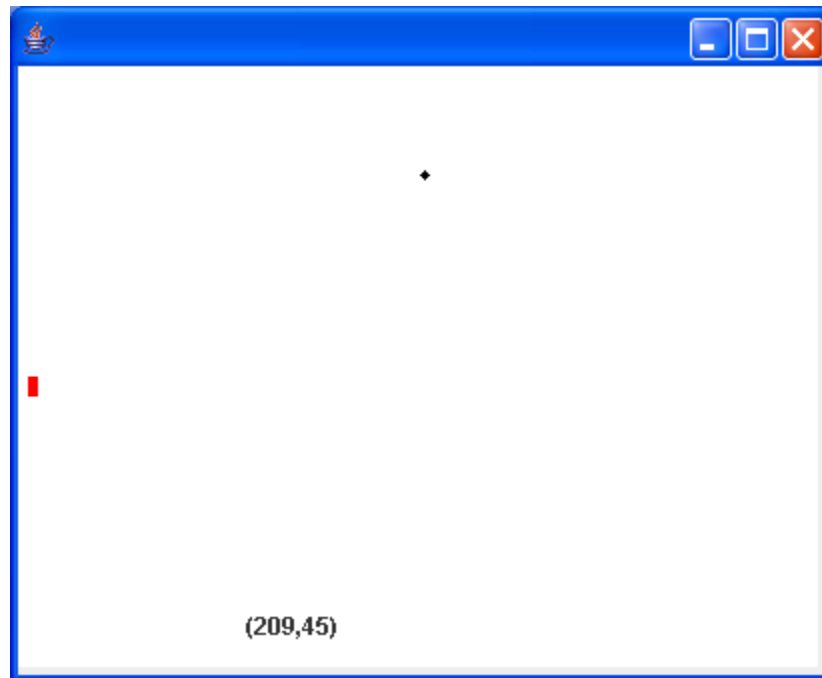


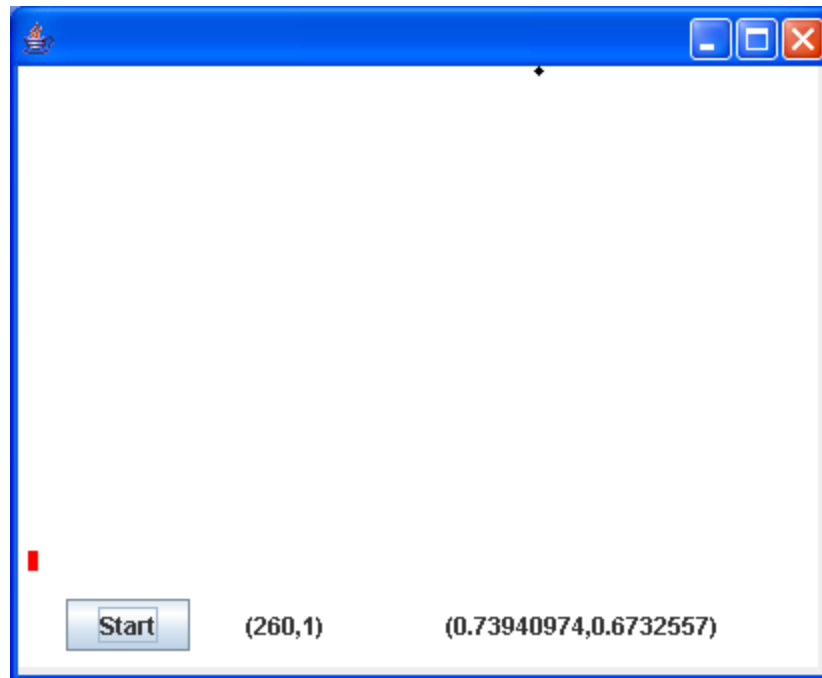




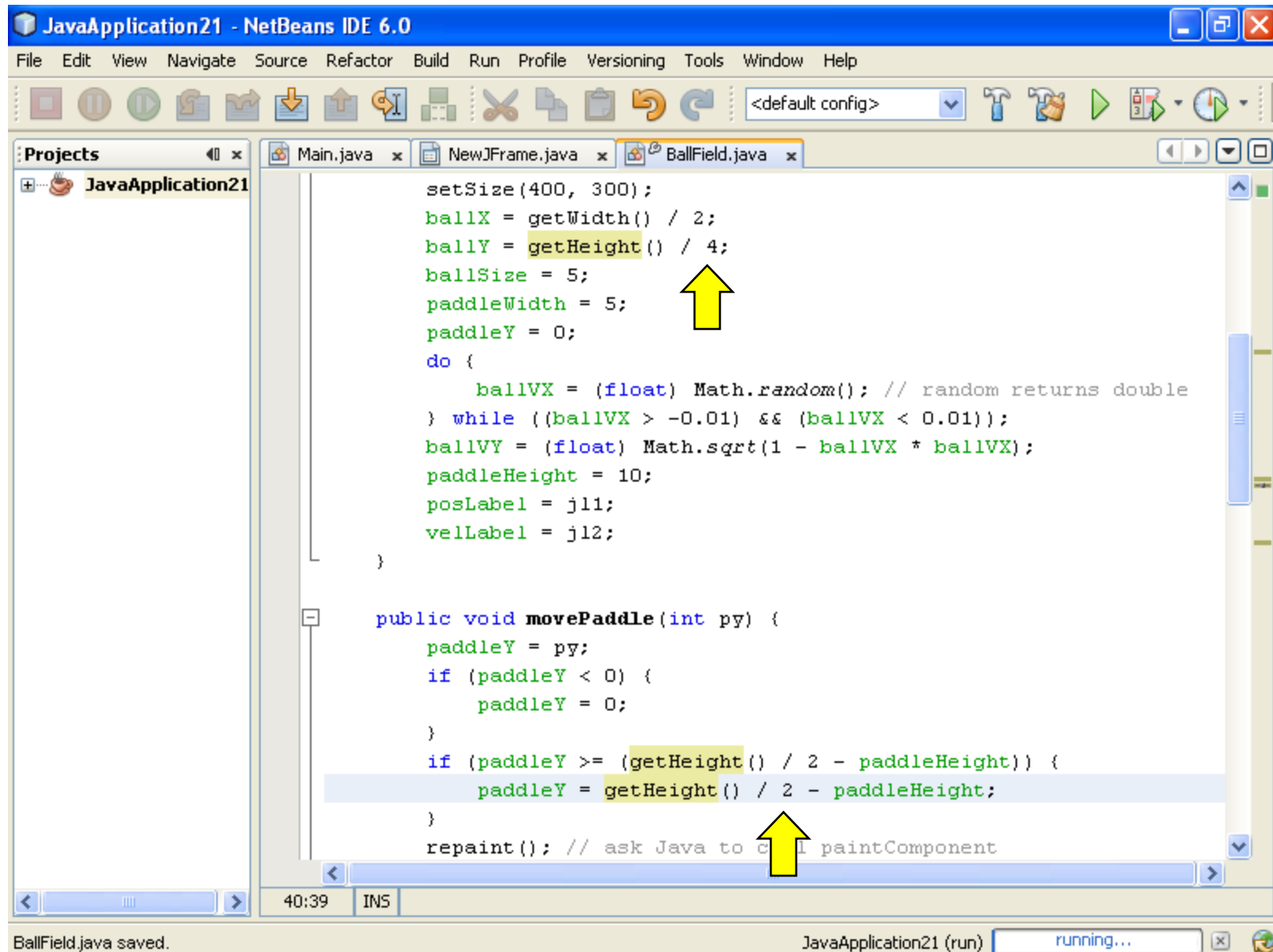


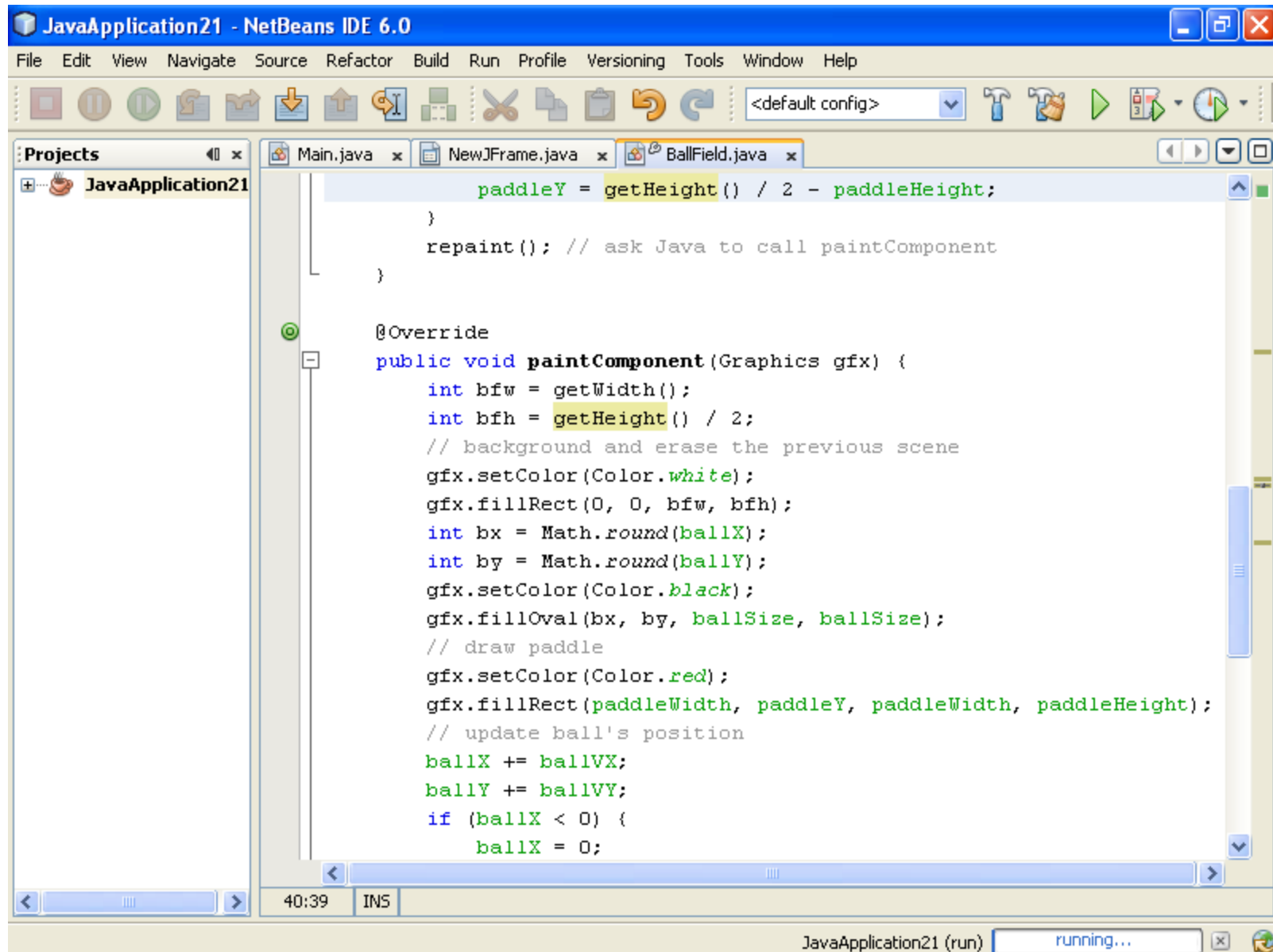






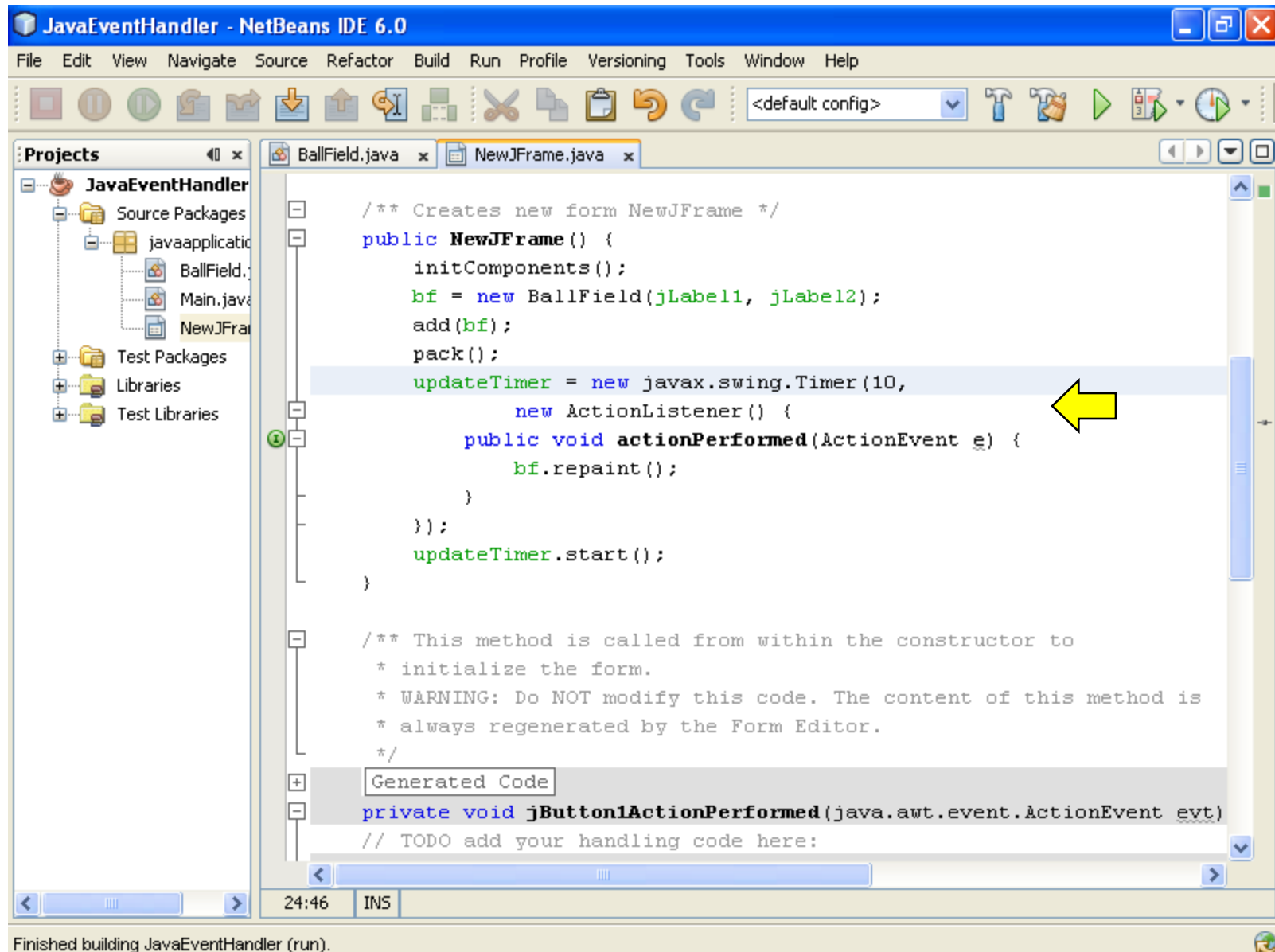
Shrink the Panel

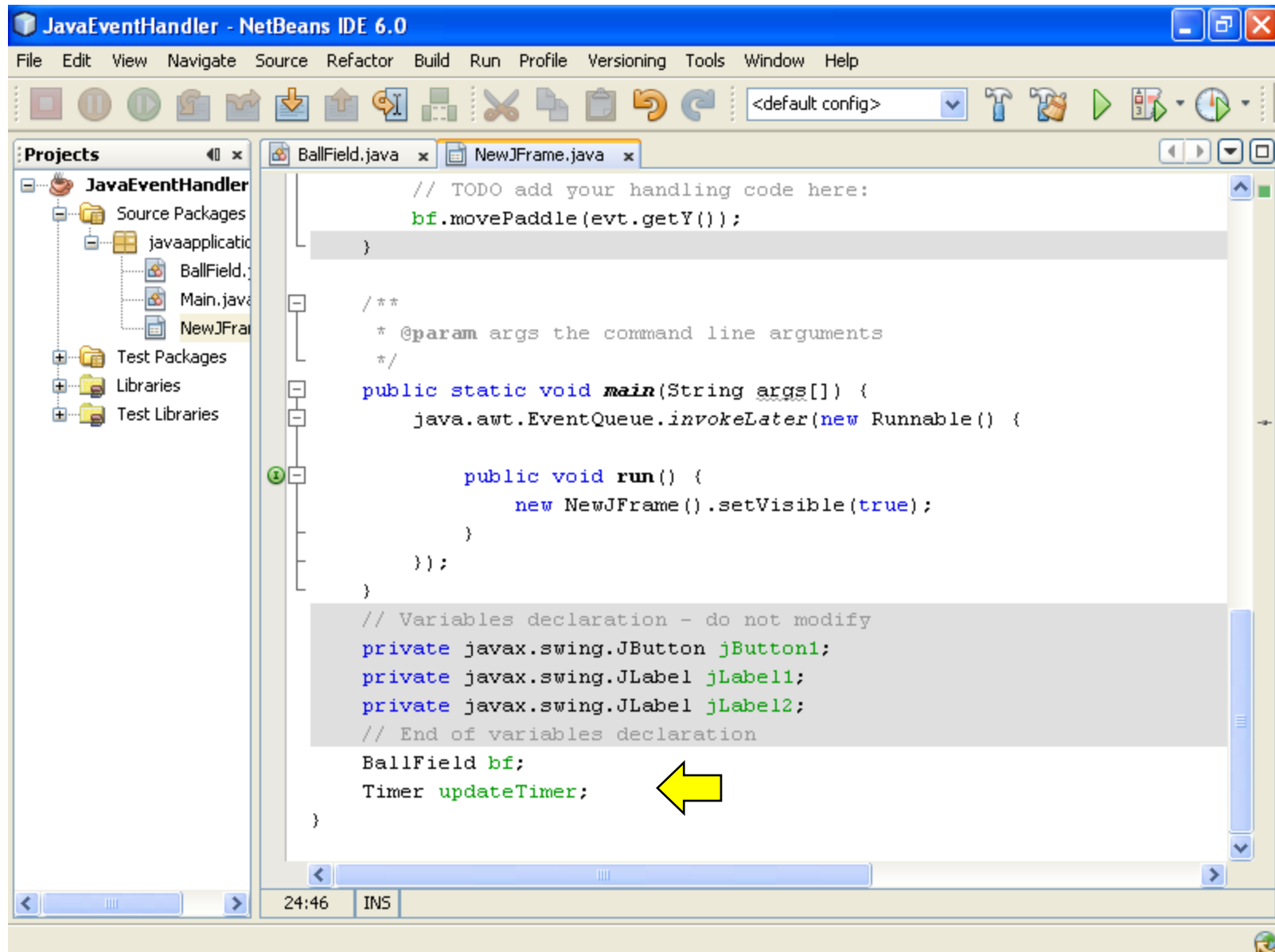


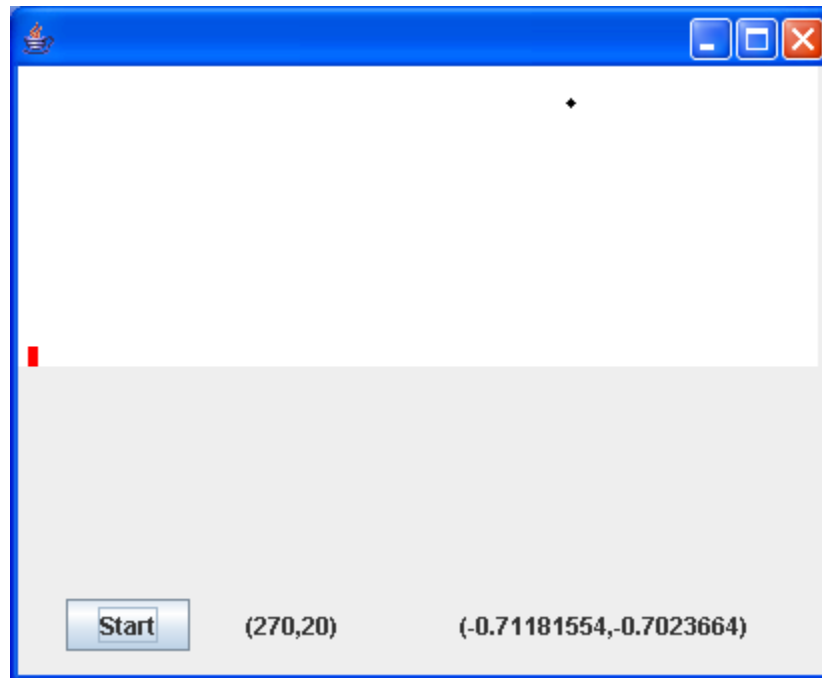




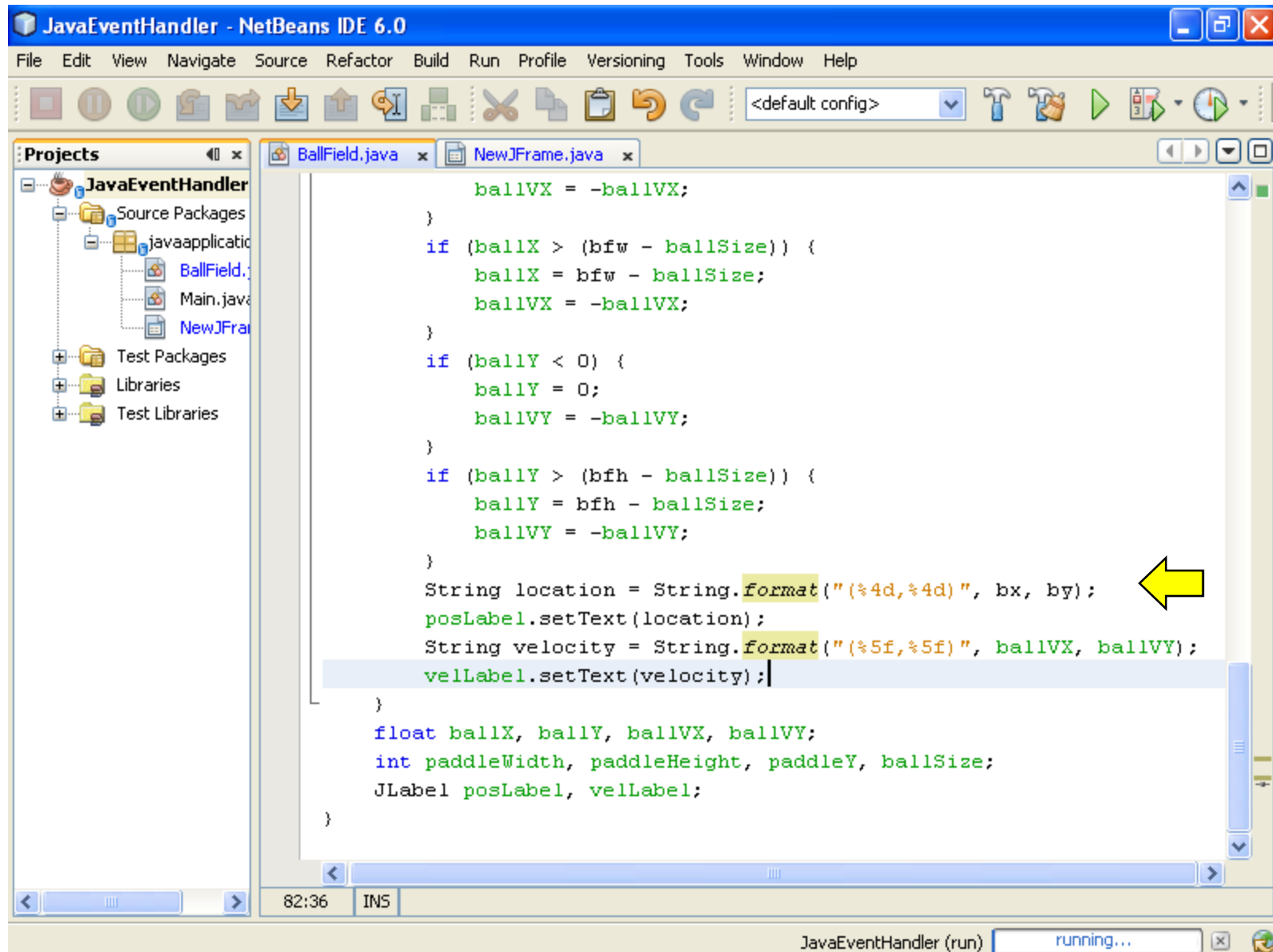
Add Timer

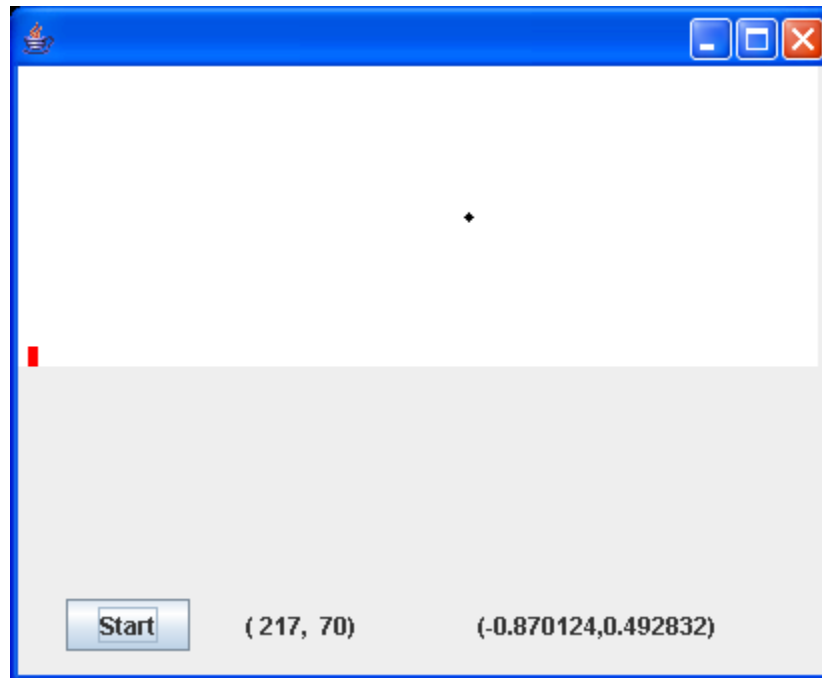




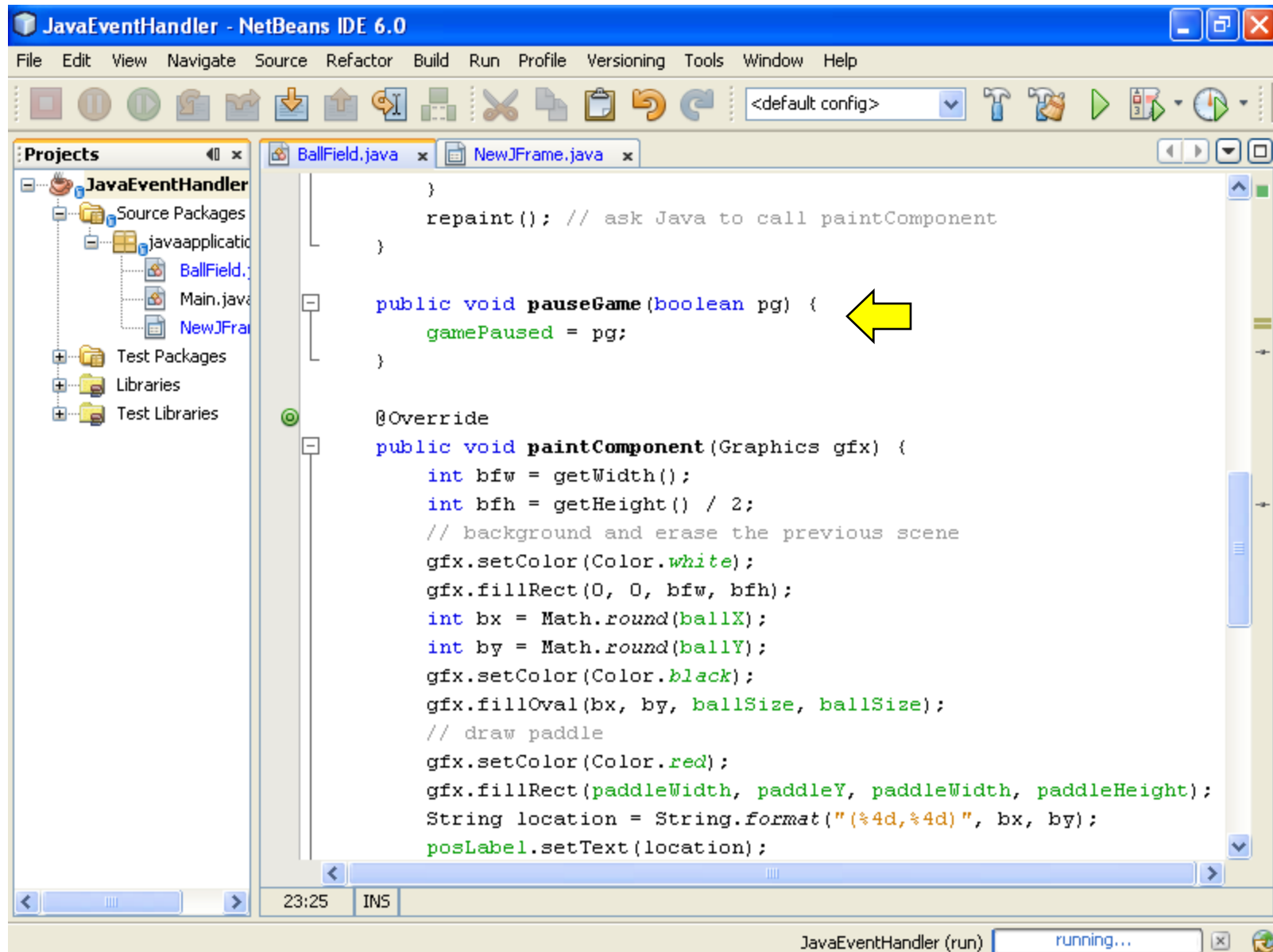


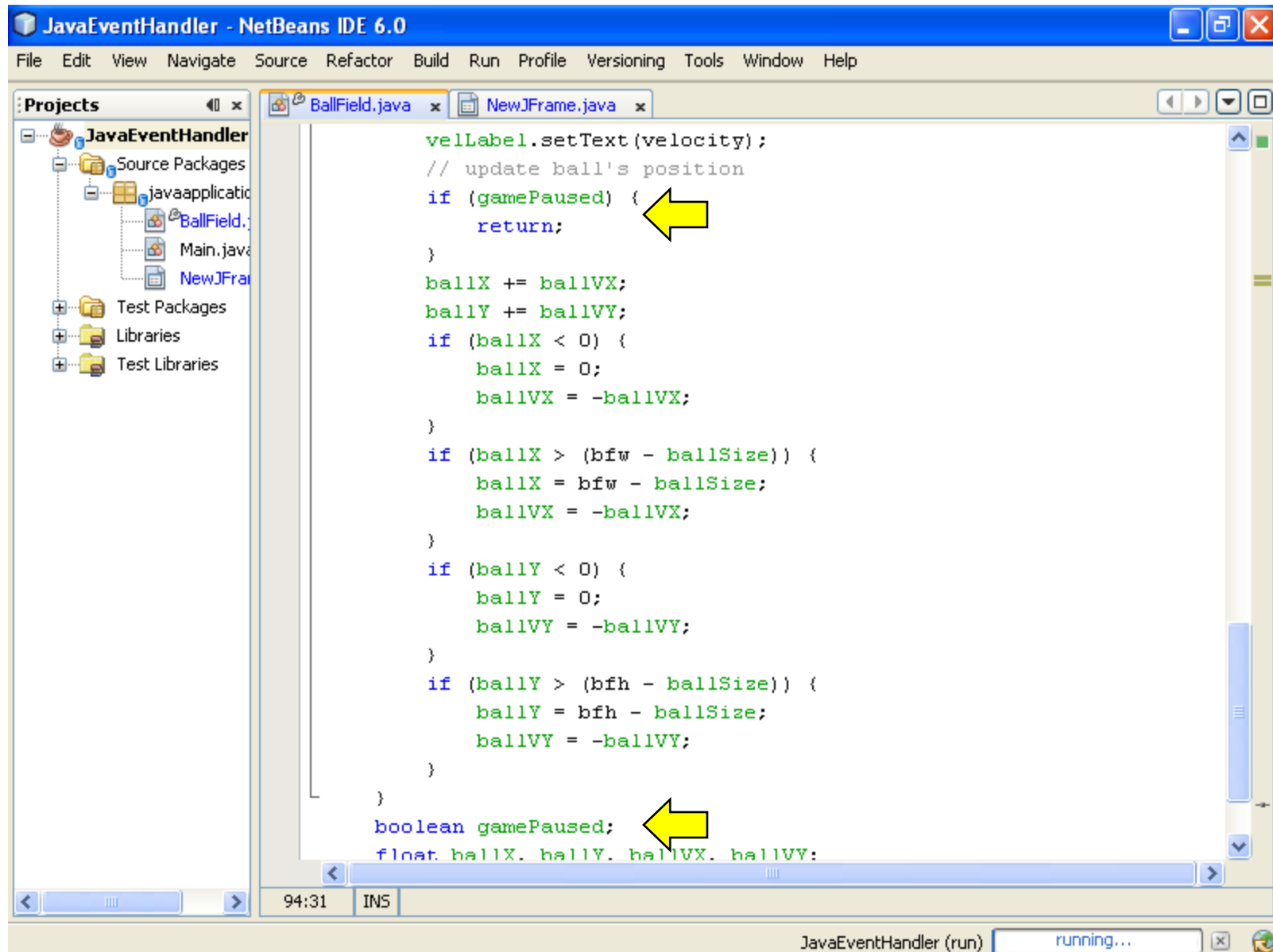
Format String

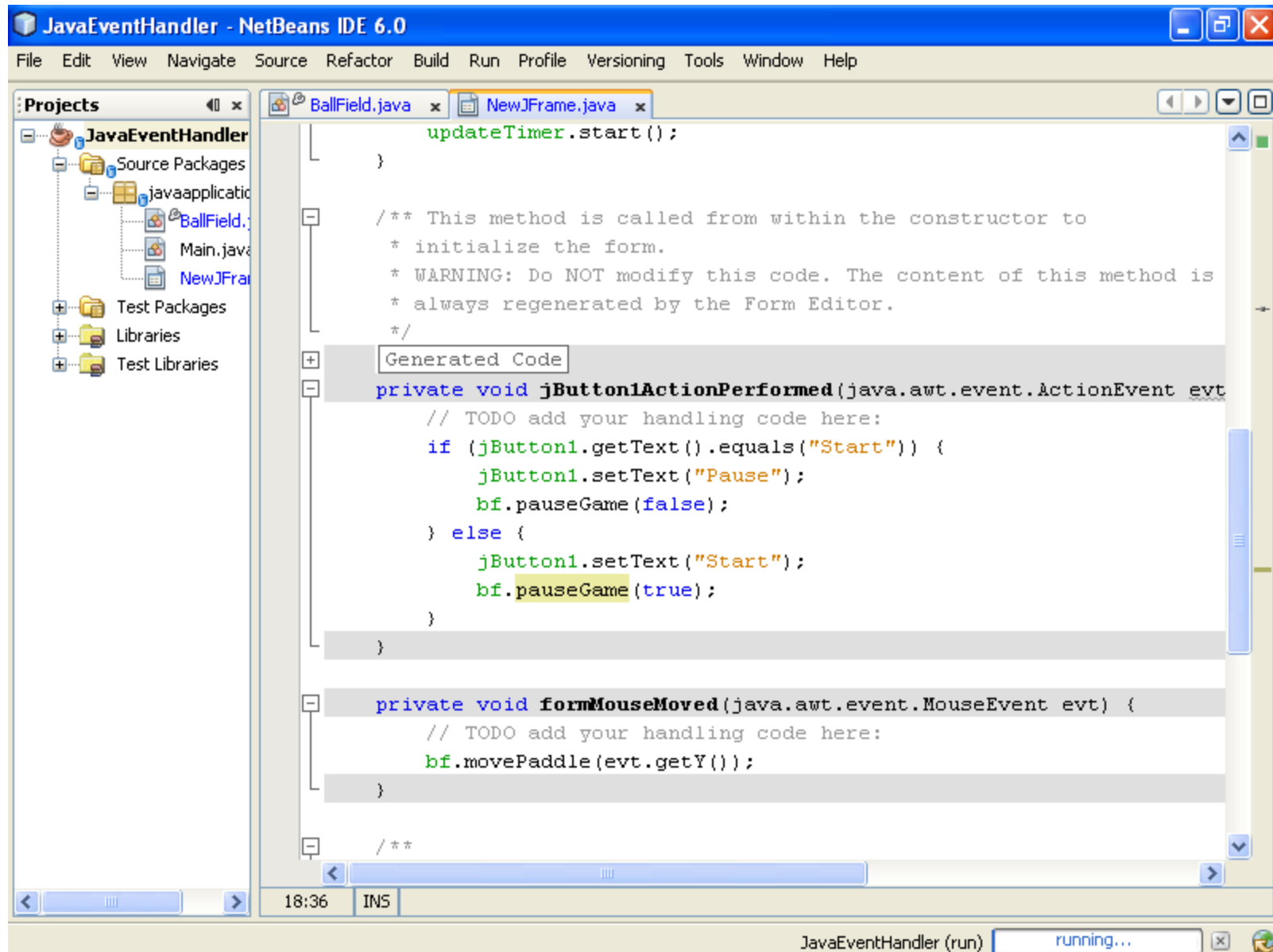




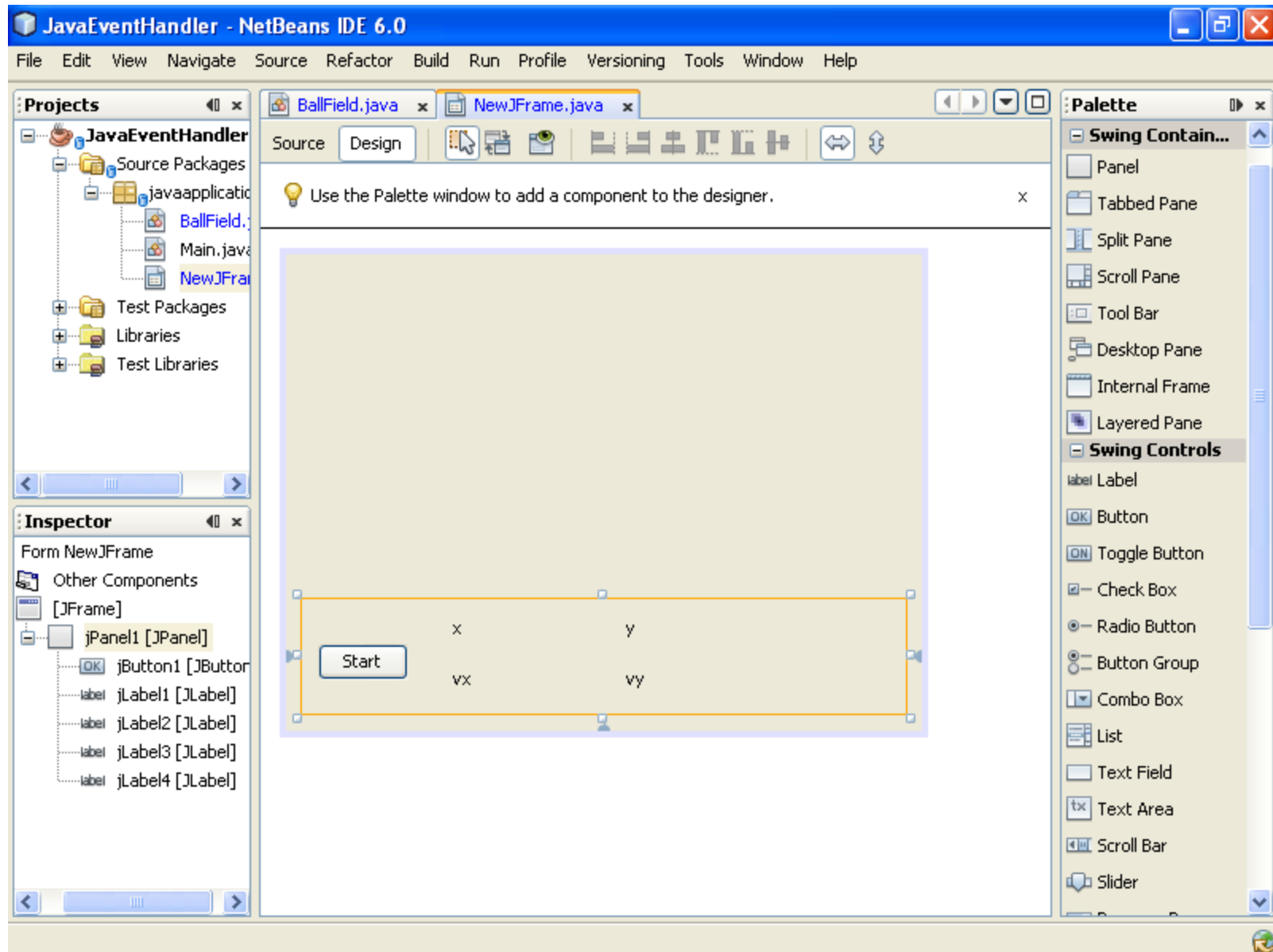
Start and Pause

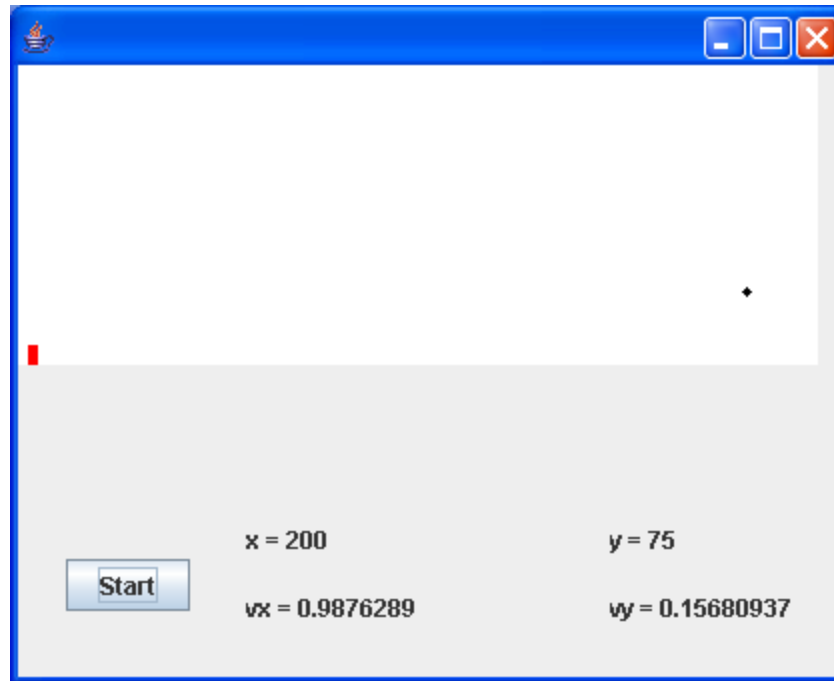






Add JPanel to Contain Button and Labels





Self Test

ECE 462

Object-Oriented Programming

using C++ and Java

Array

Yung-Hsiang Lu
yunglu@purdue.edu

Array

Java

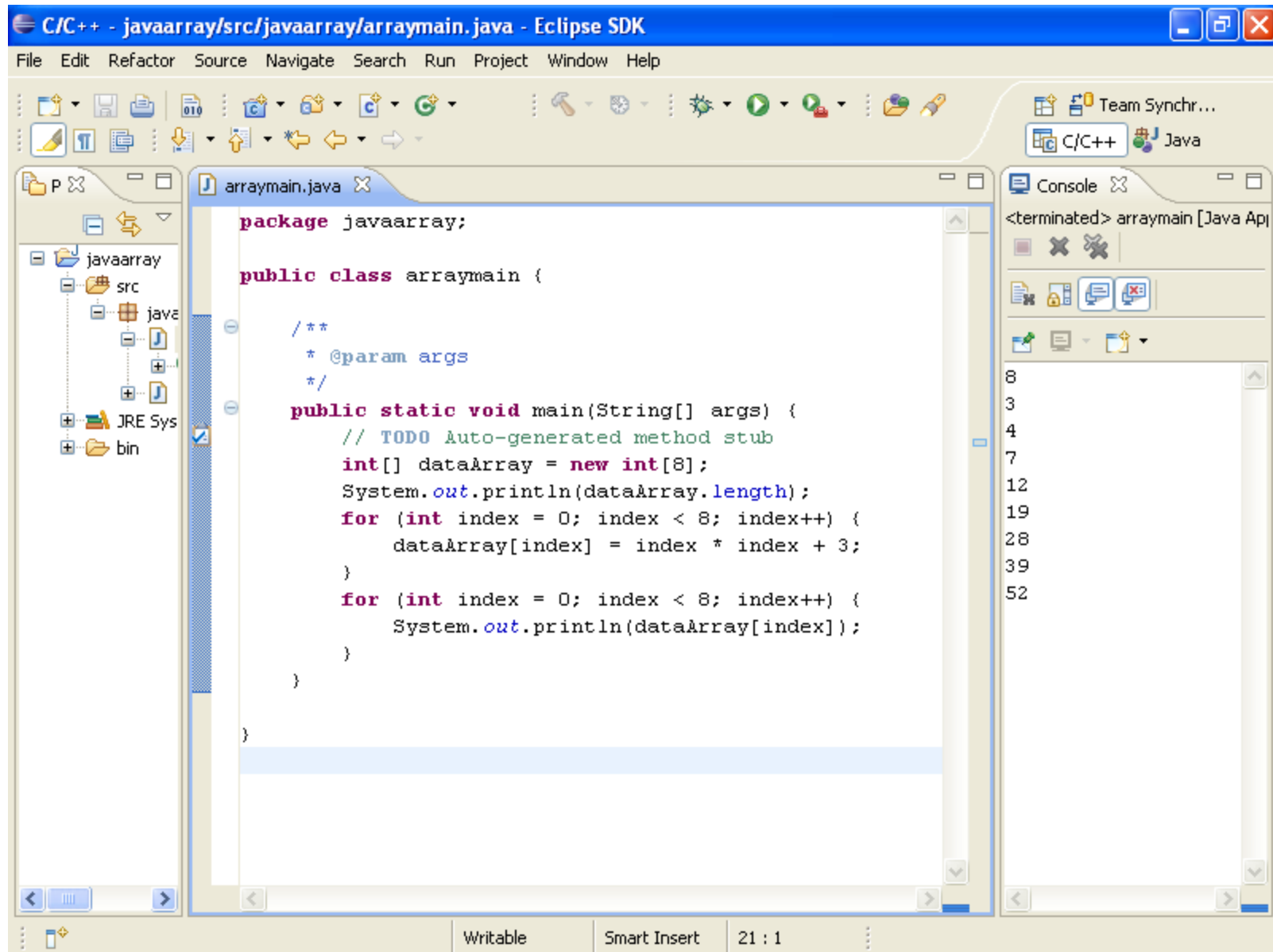
```
int [] dataArray = new int[4];  
dataArray[2] = 7;  
// Java uses garbage collection  
// start from [0]  
User [] usrList = new User[10];  
// int dataArray[8] ⇒ error  
// Java arrays know their lengths  
dataArray.length;
```

C++

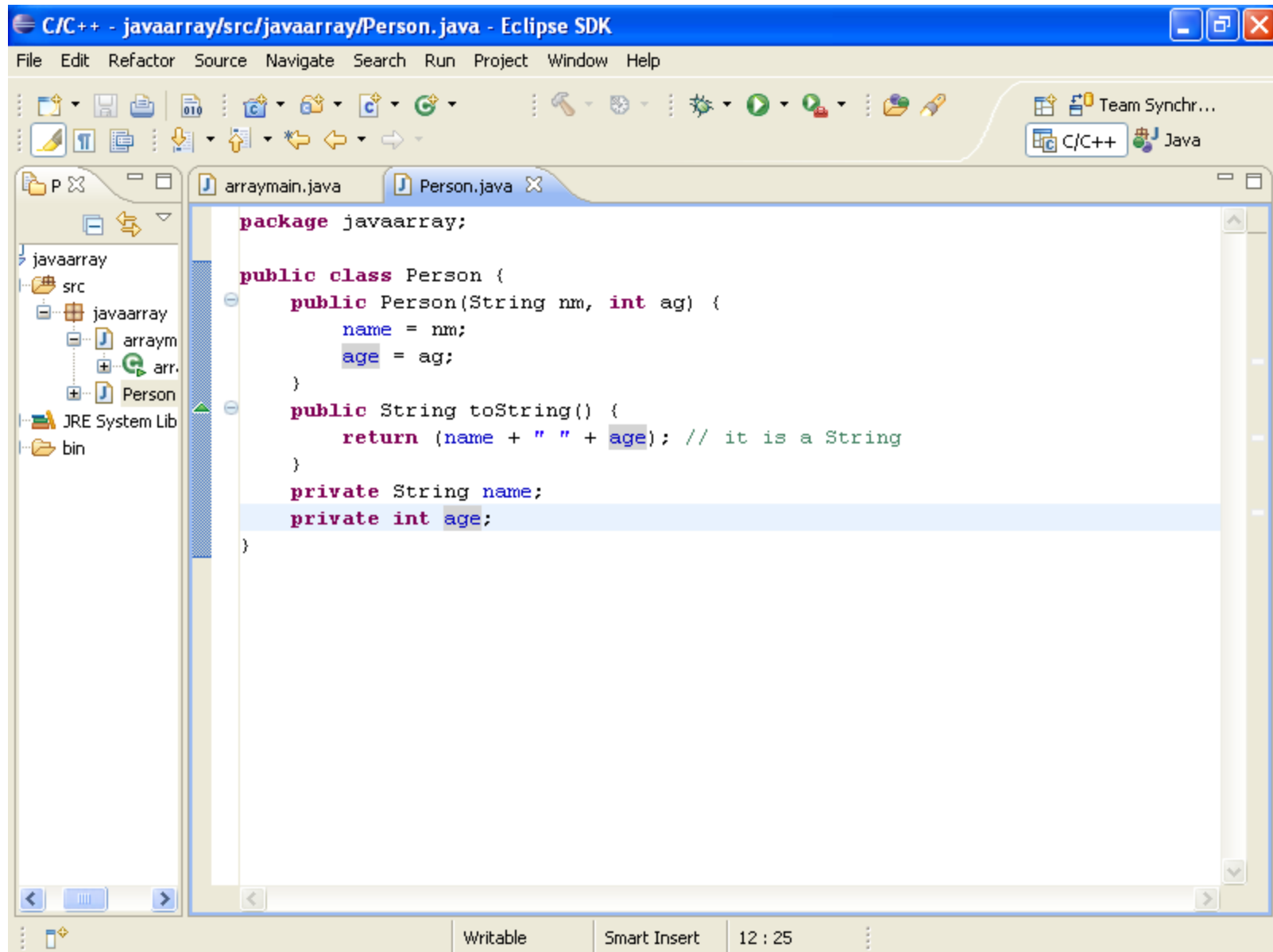
```
int * dataArray = new int[4];  
dataArray[2] = 7;  
delete [] dataArray; // use []  
// same  
User * usrList = new User[10];  
int dataArray[8];  
// not for C++
```

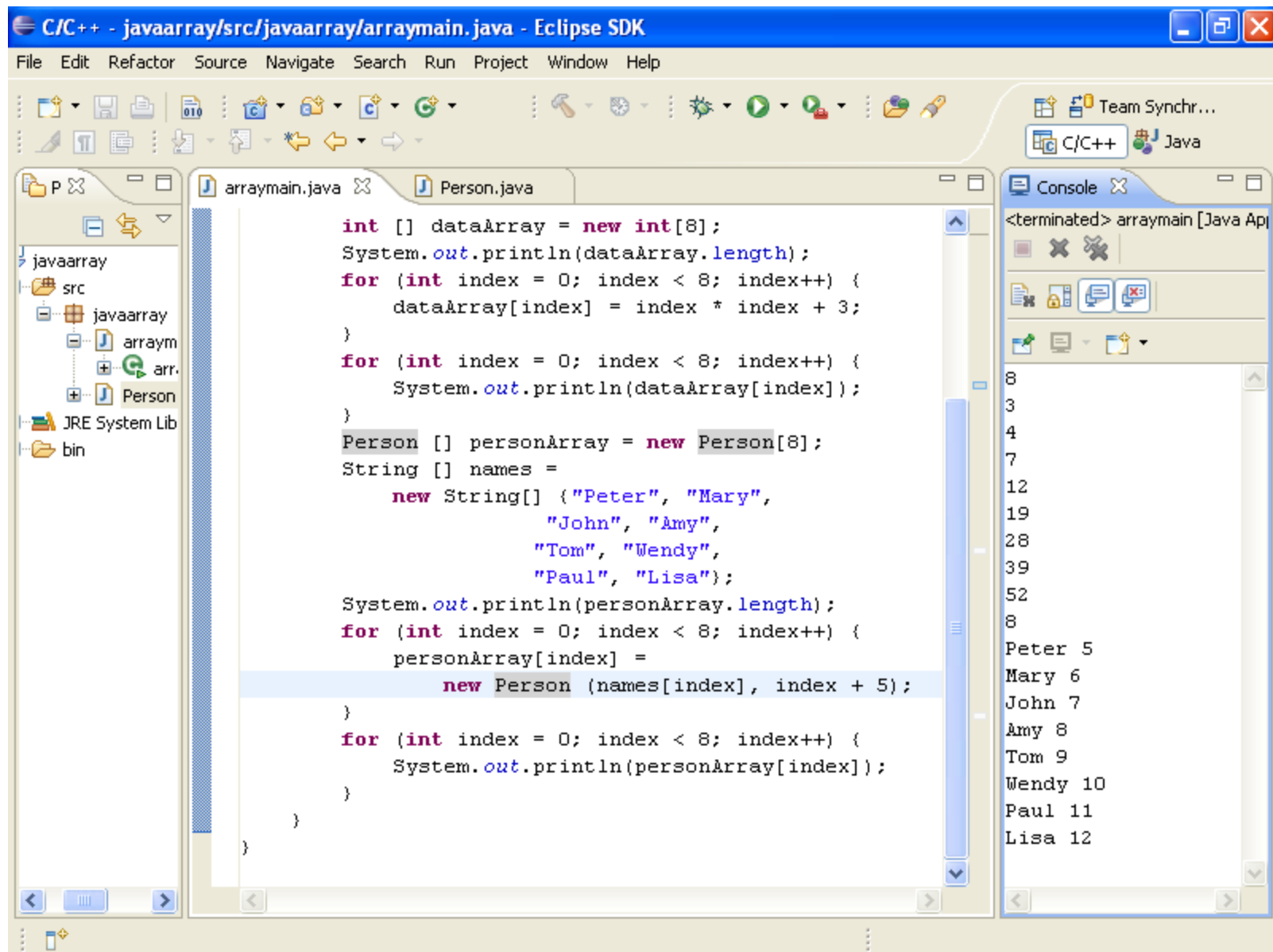
Java Array

Array of int

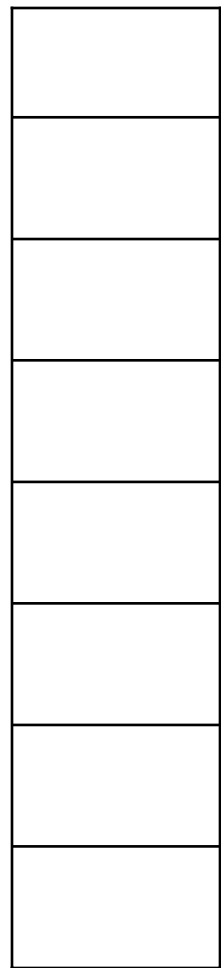


Array of Objects





```
Person [] personArray = new Person[8];
```



```
personArray[index] = new Person (...)
```

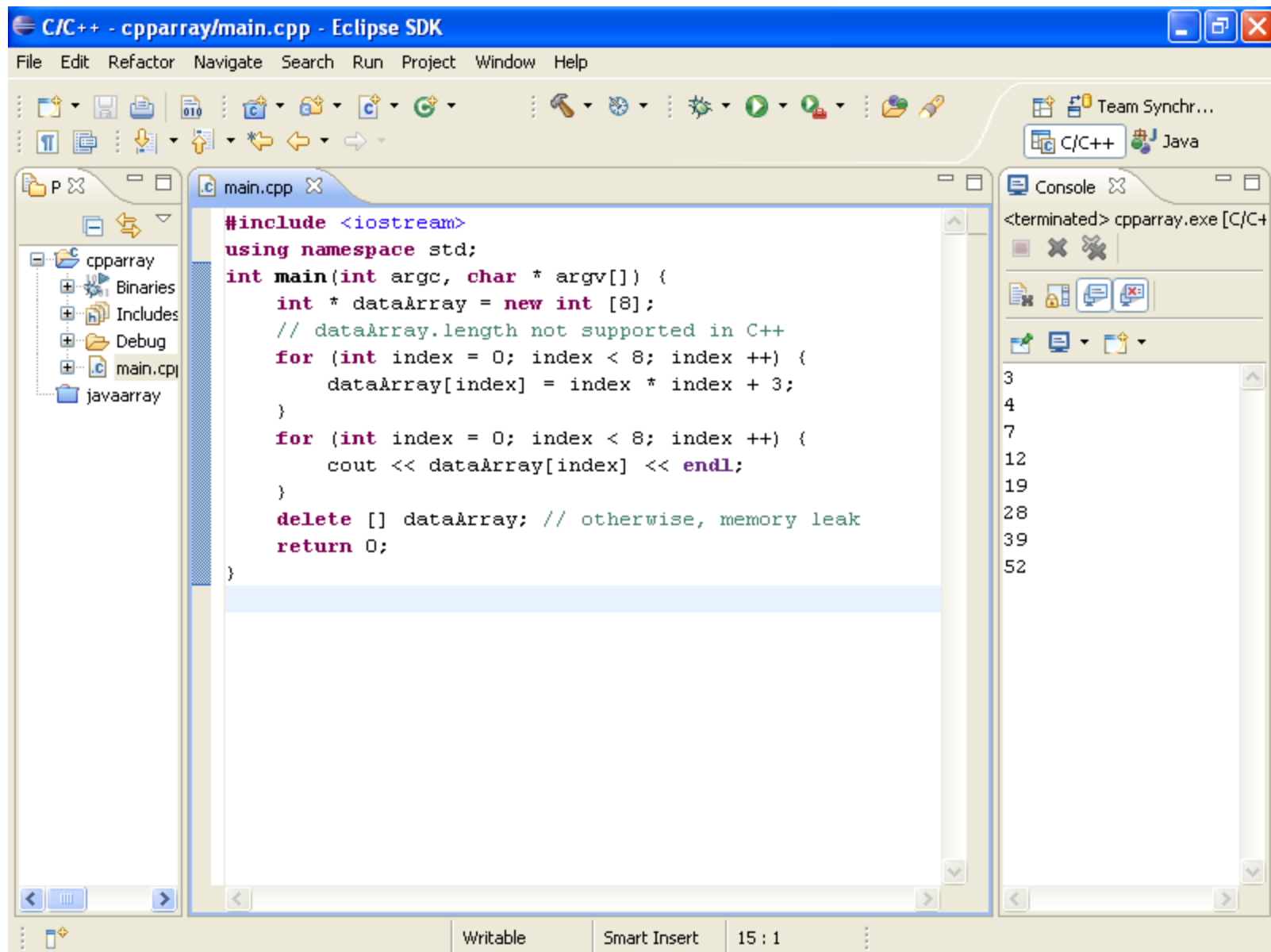
Peter, 5

```
personArray[index] = new Person (...)
```

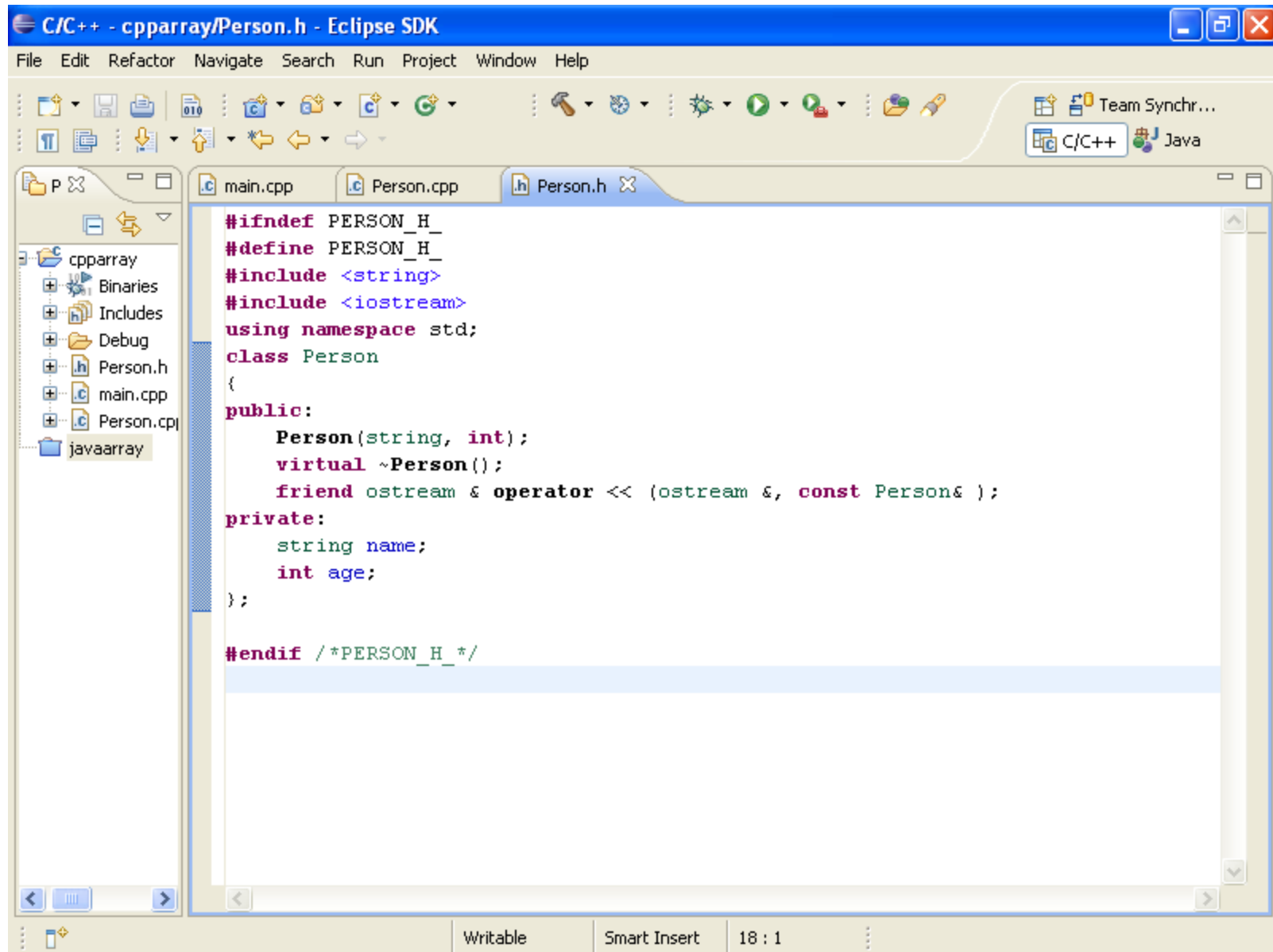
Lisa, 12

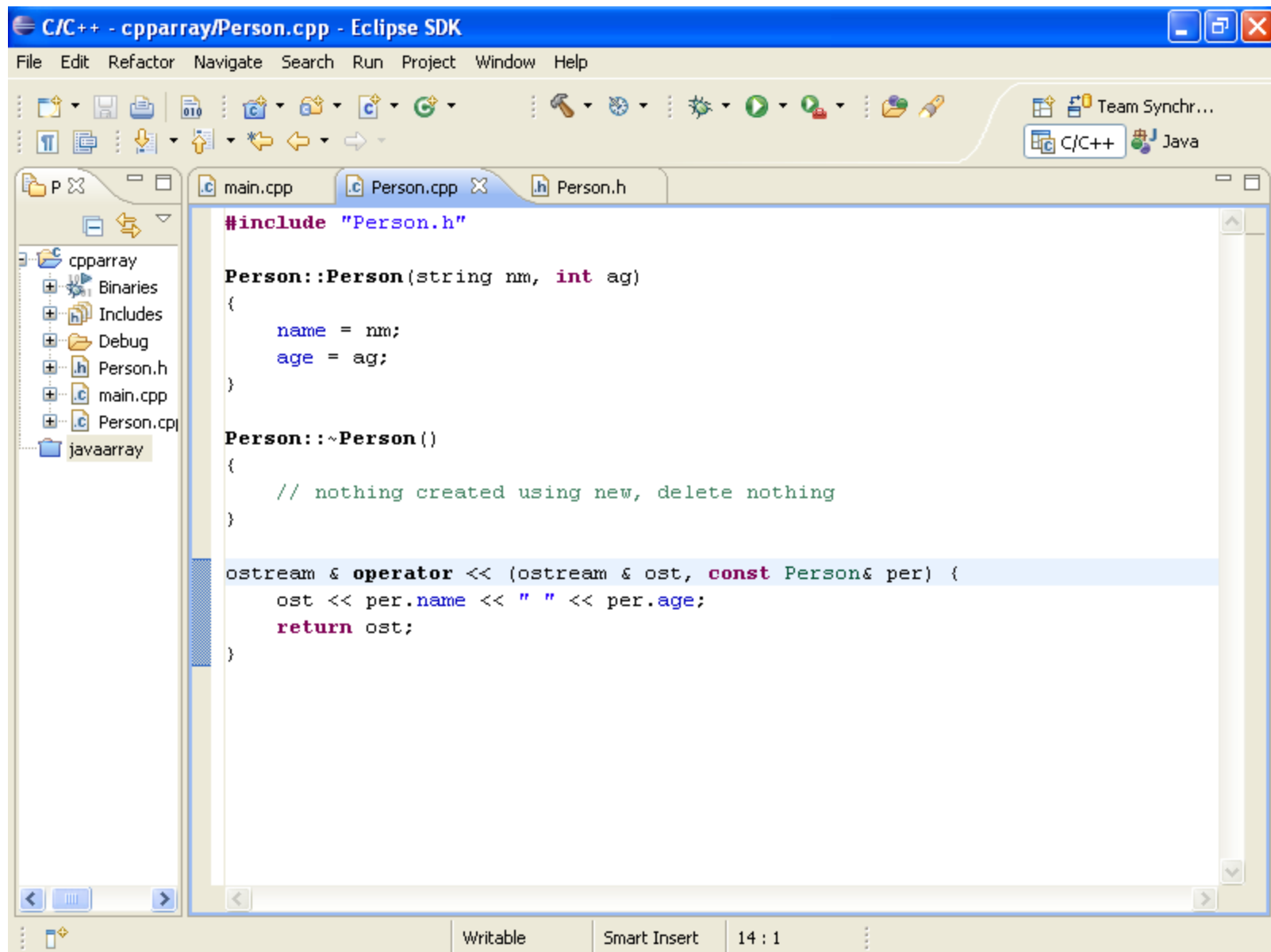
C++ Array

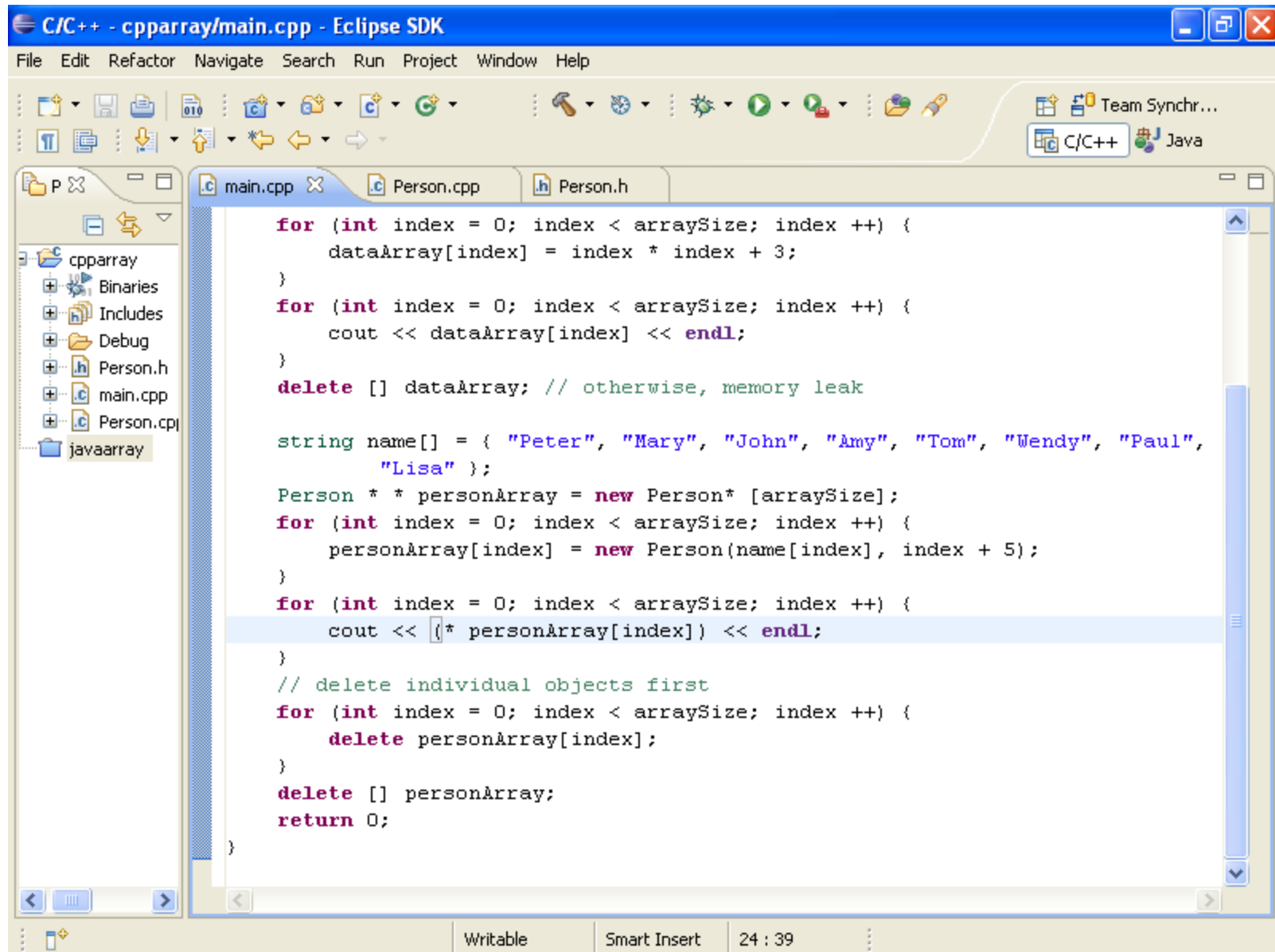
Array of int

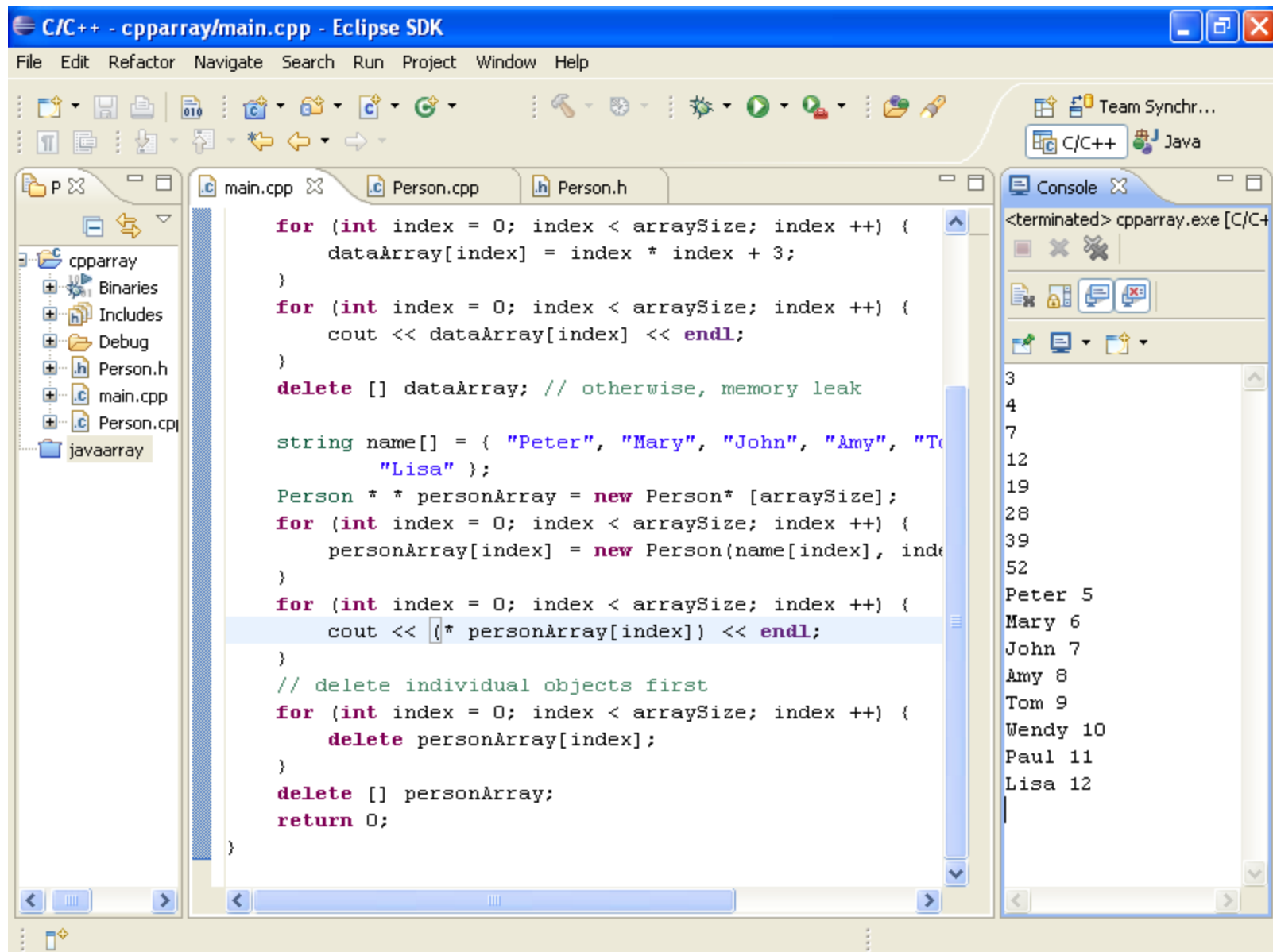


Array of Objects

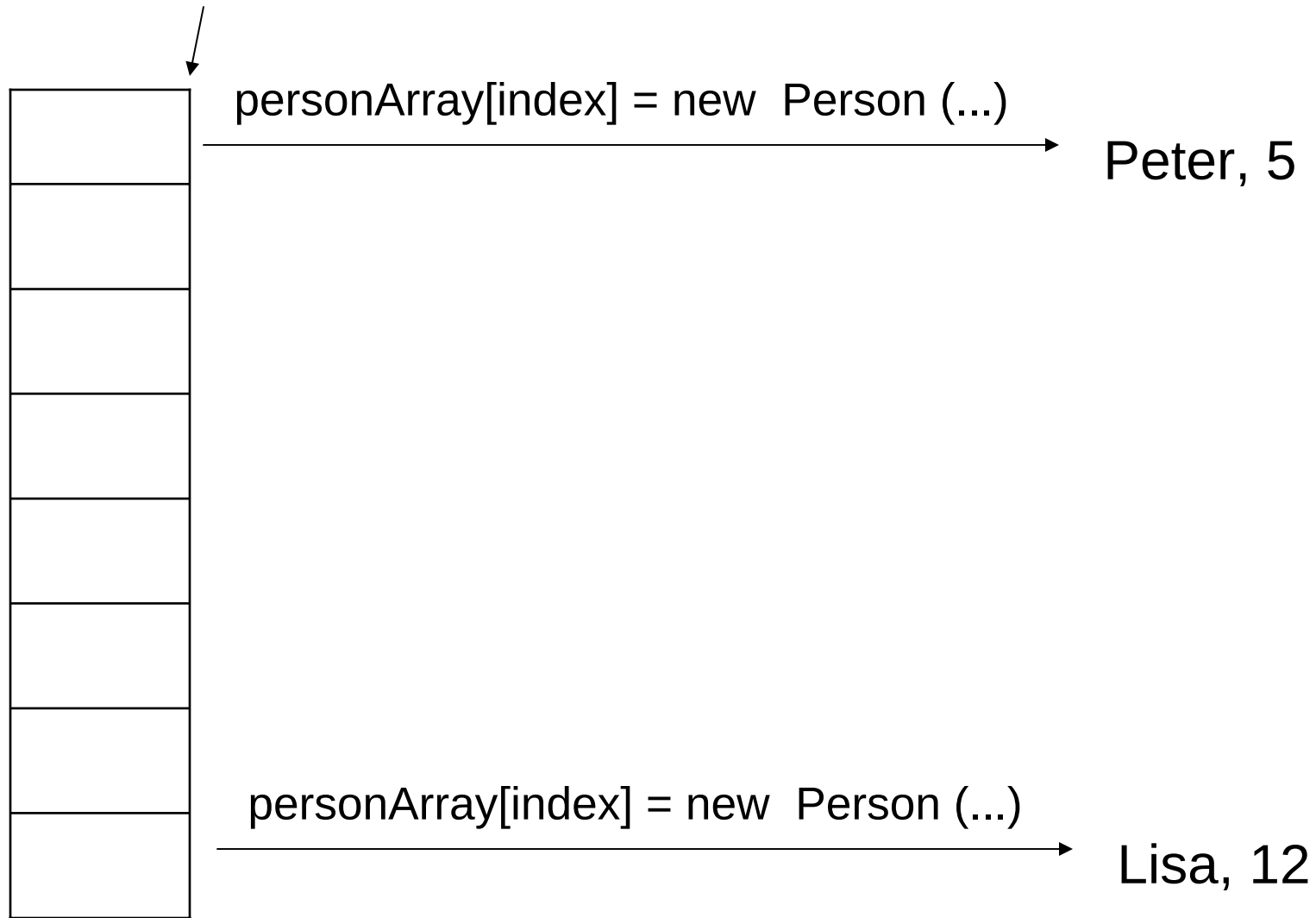






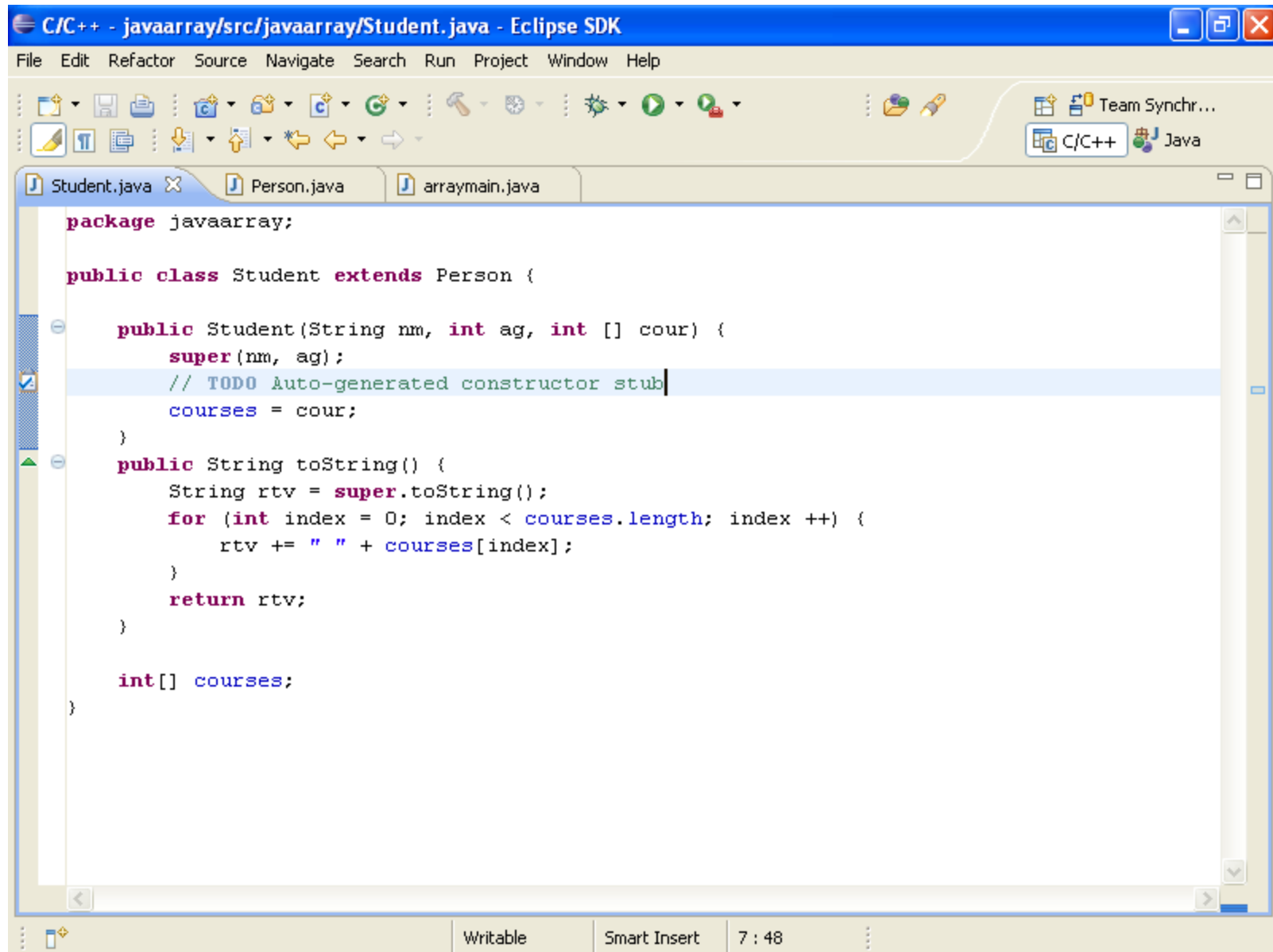


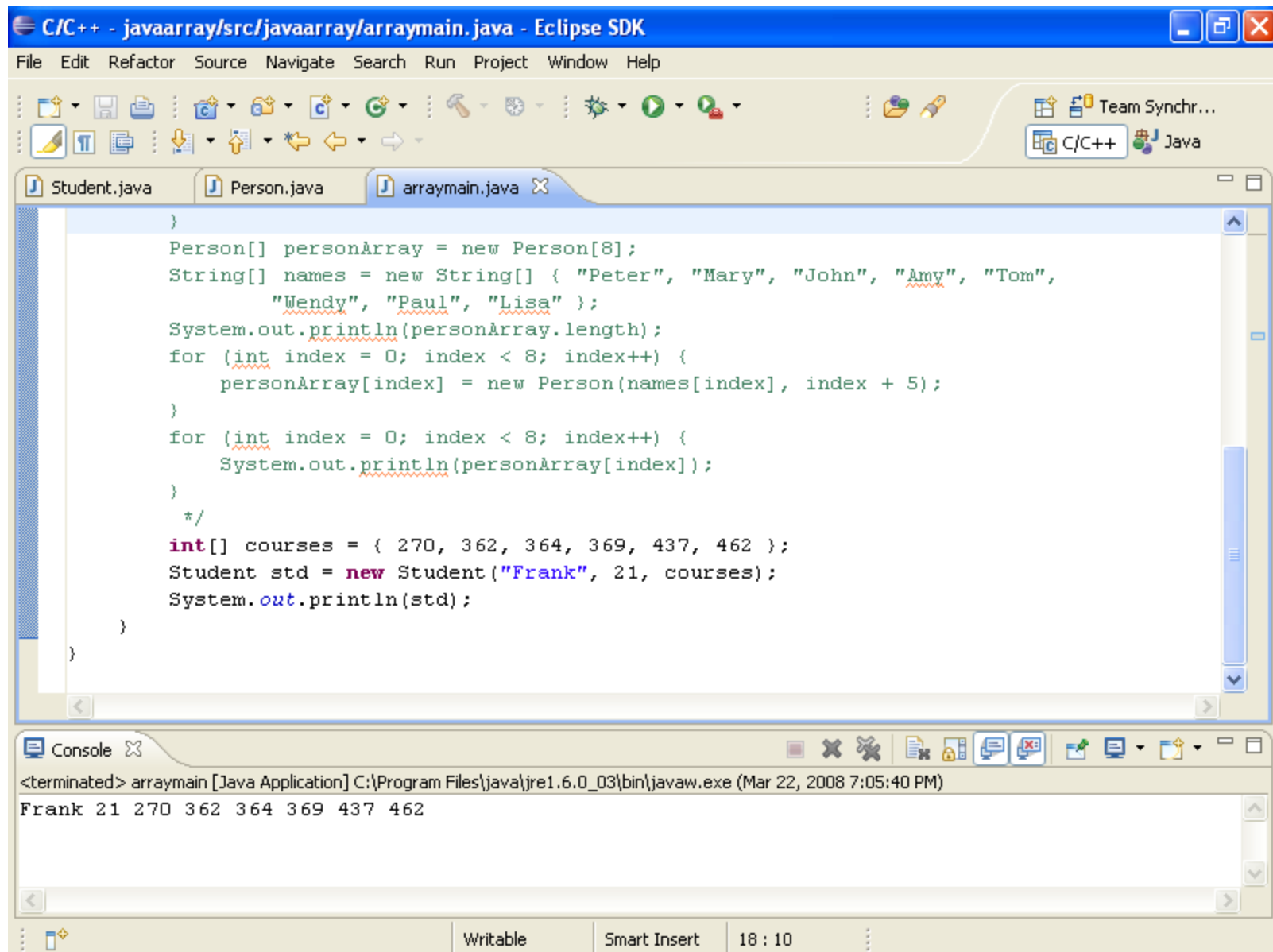
```
Person ** personArray = new Person* [arraySize];
```



Objects Contain Arrays

Java





C++

```
#ifndef STUDENT_H_
#define STUDENT_H_
#include "Person.h"
class Student : public Person
{
public:
    Student(string nm, int ag, int * cour, int numcour);
    virtual ~Student();
    friend ostream & operator << (ostream &, const Student & );
private:
    int * courses;
    int numberOfCourses;
};

#endif /*STUDENT_H_*/
```

**need another attribute to store the
number of elements in the array**

```
C/C++ - cpparray/Student.cpp - Eclipse SDK
File Edit Refactor Navigate Search Run Project Window Help

main.cpp Student.cpp Student.h Person.h Person.cpp

#include "Student.h"

Student::Student(string nm, int ag, int * cour, int numcour) :
    Person(nm, ag), numberCourses(numcour) {
    courses = new int[numberCourses];
    for (int index = 0; index < numcour; index++) {
        courses[index] = cour[index];
    }
}

Student::~Student() {
    delete [] courses;
}

ostream & operator <<(ostream & ost, const Student & std) {
    const Person * per = & std; // Chapter 15 Section 5
    ost << (* per);
    for (int index = 0; index < std.numberCourses; index++) {
        ost << " ";
        ost << std.courses[index];
    }
    return ost;
}

Writable Smart Insert 25 : 1
```

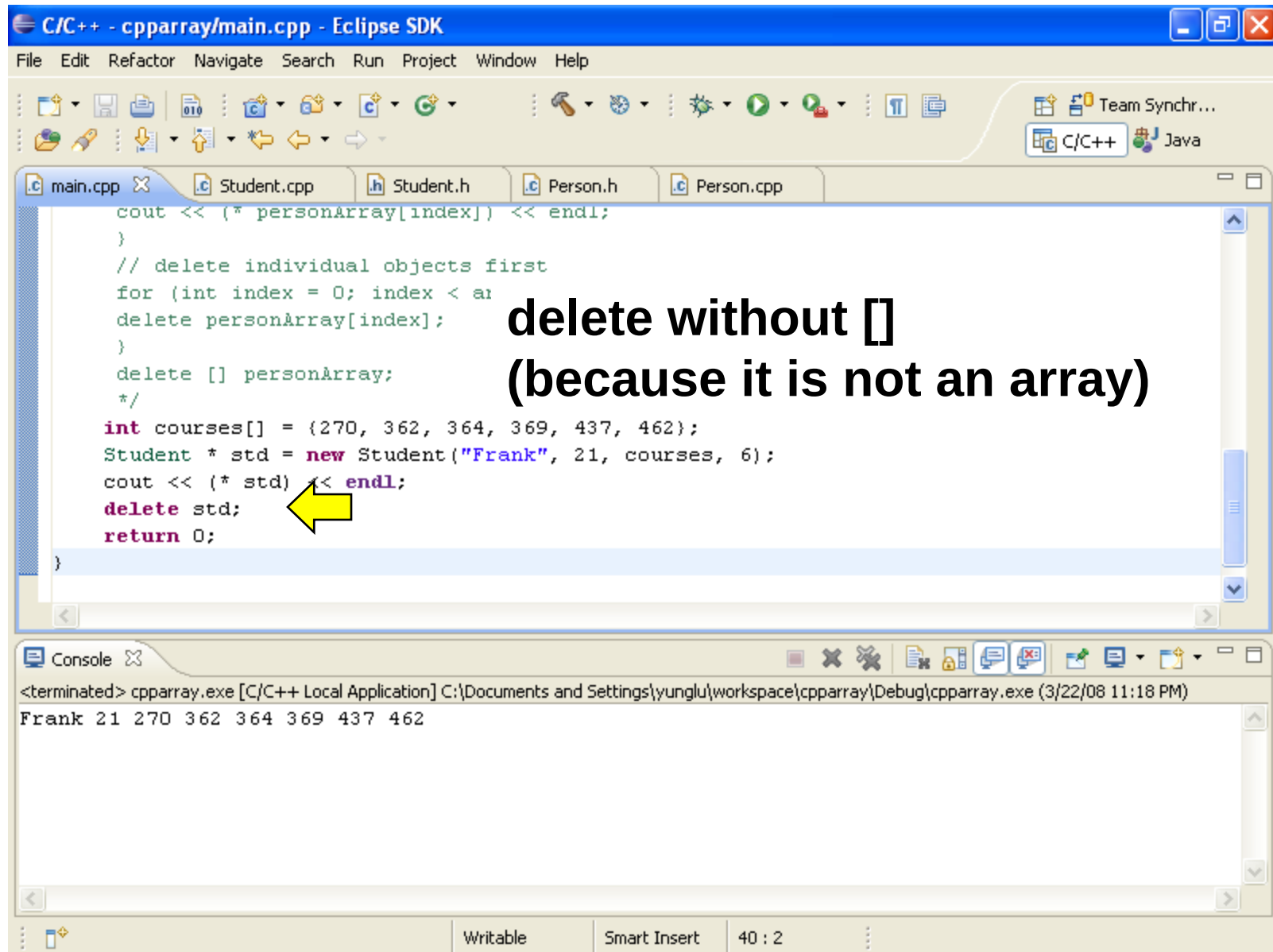
The screenshot shows the Eclipse IDE interface. The top window is titled "C/C++ - cpparray/main.cpp - Eclipse SDK". It contains the following C++ code:

```
personArray[index] = new Person(name[index], index + 5);
}
for (int index = 0; index < arraySize; index++) {
    cout << (* personArray[index]) << endl;
}
// delete individual objects first
for (int index = 0; index < arraySize; index++) {
    delete personArray[index];
}
delete [] personArray;
*/
int courses[] = {270, 362, 364, 369, 437, 462};
Student * std = new Student("Frank", 21, courses, 6);
cout << (* std) << endl;
return 0;
}
```

The bottom window is titled "Console" and shows the output of the program:

```
<terminated> cpparray.exe [C/C++ Local Application] C:\Documents and Settings\yunglu\workspace\cpparray\Debug\cpparray.exe (3/22/08 10:35 PM)
Frank 21 270 362 364 369 437 462
```

The status bar at the bottom indicates "Writable", "Smart Insert", and "40 : 1".



Self Test