ECE 462
Object-Oriented Programming using C++ and Java

Object and Class

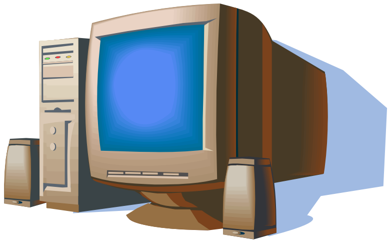
Yung-Hsiang Lu

yunglu@purdue.edu

What is an Object?

An object can be a “concrete and tangible” entity that can be separated with **unique properties**:

- you
- your book
- your car
- my computer
- Tom
- Amy’s computer
- your phone
- Sam’s digital camera
- Jennifer’s cat ...



What is an object?

- An object **can be abstract** and does not have to be tangible:
  - Purdue ECE's student database
  - the email sent by Mark at 9:07AM on 2008/03/22
  - the web page of Purdue ECE 462
  - the song played in WBAA at 7:02PM last night
- An object can **contain** other objects:
  - a car = wheels + engine + door + windshield + ...
  - a house = kitchen + bedrooms + living room + ...
  - a laptop = keyboard + display + processor + ...

Objects' Three Properties

- Each object is unique and can be **identified** using name, serial number, relationship with another object ...
- Each object has a set of **attributes**, such as location, speed, size, address, phone number, on/off ...
- Each object has unique **behaviors**, such as ring (phone), accelerate and move (car), take picture (camera), send email (computer), display caller (pager)
- Each object has **three** important properties:
  - **unique identity**
  - **attributes**, noun
  - **behavior** (action), verb

Objects’ Interactions

- You (object) press (action) the pedal (object) of your car (object). As a result, your car accelerate (action).
- When your phone (object) rings (action) and alerts (action) you (object) of an incoming call (state), you answer (action) the call (state).
- You submit (action) homework (object) and it is graded (action) with a score (state).

Object as a Special Case

- A person is an object. A student is also an object. A student is a **special case** of a person

⇒ A student has **all attributes** of a person:

name, home address, parents ...

⇒ A student has **all behavior** of a person:

eat, sleep, talk ...

⇒ A student has something that a person may not have:

– attributes: student ID, list of courses, classmates ...

– behavior: submit homework, take exam ...

What is a Class?

- A class **describes the commonalities** of similar objects:
  - Person: you, David, Mary, Tom, Amy ...
  - Car: your Toyota Camry, his Ford Explorer, Jennifer's Mercedes C300 ...
  - Classroom: EE170, EE117, EE129 ...
  - Building: EE, MSEE, Purdue Bell Tower, Hovde Hall...
- A class describes both the attributes and the behavior:
  - Person: name, home ... + sleep, eat, speak ...
  - Car: engine size, year ... + accelerate, brake, turn ...

Relationship among Classes

- A class can be a **special case** of another class:
  - Student is a special case of Person
  - Sedan is a special case of Car
  - Laptop is a special case of Computer
  - Computer is a special case of ElectronicMachine

⇒ This is called a **"is a"** relationship.

- any Student object is a Person object
- any Sedan object is a Car object
- any Laptop object is a Computer object
- any Computer object is an ElectronicMachine object

Class and Object

- An object is an **instantiation** (i.e. concrete example) of a class:
  - an object is unique
  - a class describes the common properties of many objects
- An object may contain an object. This must be described in the former object's class. We can say that one class **"has a"** class.

Encapsulation

- An object can hide information (**attributes**) from being manipulated by or even visible to other objects:

  A person's name is given once when the object is created. This attribute is visible but cannot be changed.

- An attribute may be modified by only restricted channels to keep consistency.

  A person's address and phone number must be change together when this person moves.

Self Test

ECE 462
Object-Oriented Programming using C++ and Java

Inheritance and Polymorphism

Yung-Hsiang Lu

yunglu@purdue.edu

Inheritance = "Is A"

- Any Student object **is a** Person object. Student class is a **derived class** of Person. Person is the **base class**.

⇒ Person is more general, with fewer attributes and behaviors.

⇒ Student is more specific, with more attributes (school, major) and behaviors (submit homework, take exam).

- Any TabletPC object **is a** Computer object. TabletPC class is a **derived class** of Computer.

⇒ Computer is more general.

⇒ TabletPC is more specific, with more attributes (battery lifetime) and behavior (close or turn the screen)

Derived Class

- A class may have **multiple** derived classes:
  - Car: Sedan, Truck, Sport Utility Vehicle, Sport Car ...
  - Computer: Laptop, Desktop, Server
  - Person: Student, Teacher, Father, Mother ...
- A derived class may also have derived classes:
  - Vehicle: Car, Bike ... Car: Sedan, Truck ...
  - Animal: Bird, Mammal ... Mammal: Dog, Cat ...
- Use "base" and "derived" classes. **Do not** use "super" and "sub" classes. A base class or a superclass is "smaller" (fewer attributes and behaviors)

  ⇒ too confusing

Why Object-Oriented?

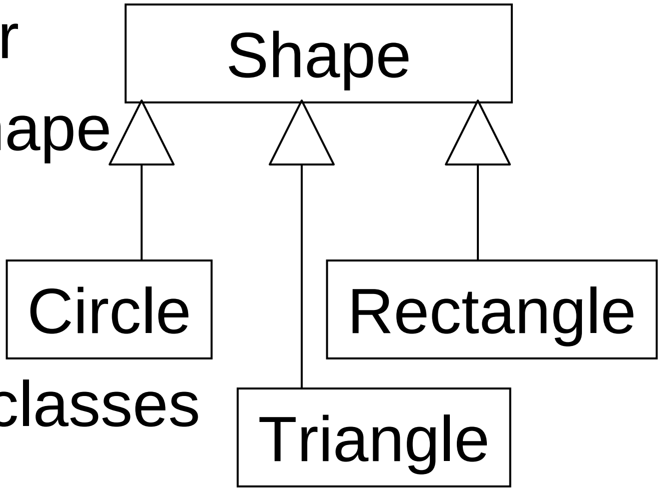
- Object-oriented programming (**OOP**) is more **natural** to describe the interactions between "things" (i.e. objects).
- OOP provides better **code reuse:**
  - commonalities among objects described by a class
  - commonalities among classes described by a base class (inheritance)
- Objects know what to do using their attributes:

  Each object responds differently to "What is your name?"
- OOP provides **encapsulation**: hide data that do not have to be visible to the other objects or protect data from unintentional, inconsistent changes.

Interface ≠ Implementation

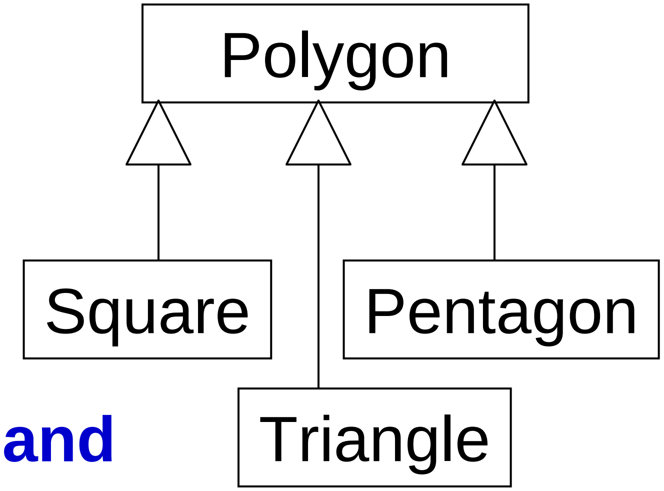
If a behavior is common among classes, the behavior should be **available** in their base class. However, this behavior may need additional information from derived classes and must be **handled** in derived classes.

- Shape: contains color, lineStyle ... attributes
- Shape **supports** getArea behavior
- getArea **cannot** be handled by Shape
- getArea must be handled by individual derived classes
- getArea **implemented** in derived classes



Override Behavior

- Polygon can support getArea.
- Derived classes (such as Triangle, Square, and Pentagon) can have better (faster) ways to getArea.



⇒ getArea is implemented in Polygon **and** the derived classes.

- A Polygon object calls getArea in Polygon
- A Square object calls getArea in Square if getArea is implemented in Square.
- A Pentagon object calls getArea in Polygon if getArea is ***not*** implemented in Pentagon.

Overriding

Base	Derived	Object	Execute
Y	Y	Base	Base
Y	Y	Derived	Derived
Y	N	B	B
Y	N	D	B
N	Y	Base	Error
N	Y	Derived	D
N	N	B	Error
N	N	D	Error

The behavior implemented in a sibling class (such as Square-Triangle) has no effect.

Class and Object

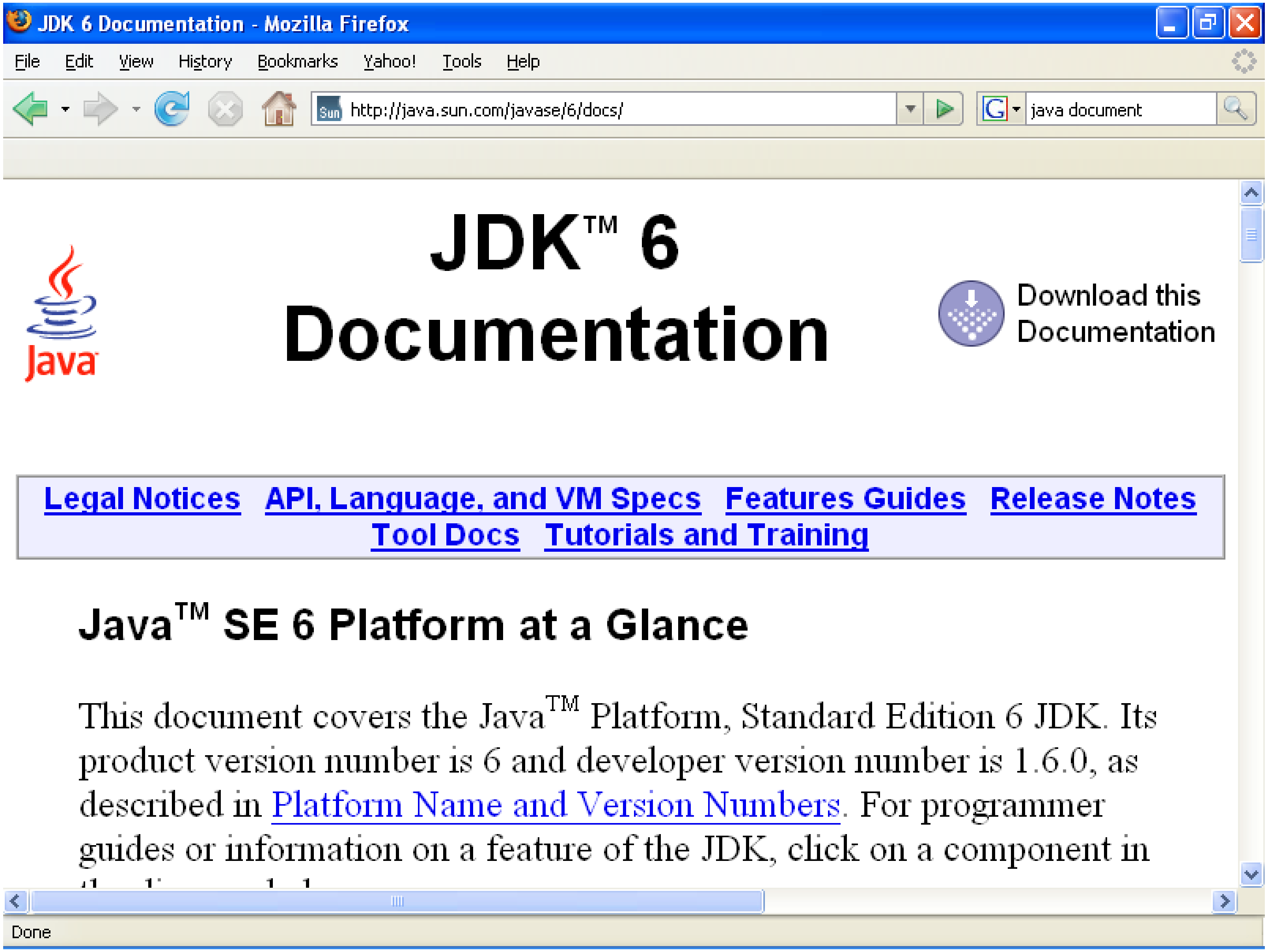
```
Polygon p1;   // this is a comment: p1 is a Polygon object
p1.getArea(); // call the implementation in Polygon
Square s2;    // s2 is a Square object
p1 = s2;      // p1 now behaves likes a square
// a Square object is always a Polygon object
p1.getArea(); // implementation in Square (if available)
// polymorphism
s2 = p1;      // error
// a Polygon object may not be a Square object
```

Fundamental Concepts in OOP

- **object and class**
- **encapsulation**
- **inheritance**
- **polymorphism**

Self Test

Java and Qt Documentations

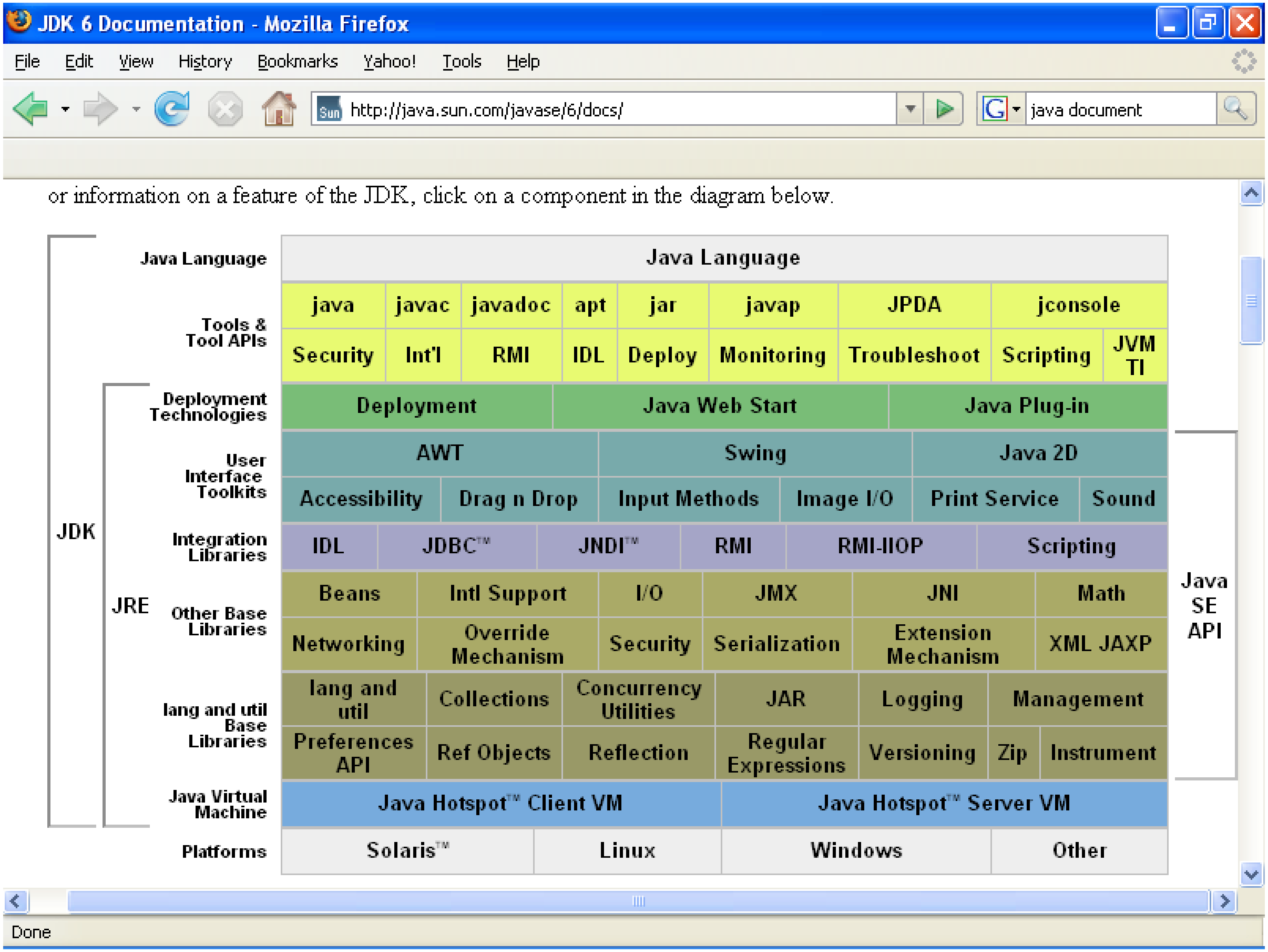

JDK 6 Documentation - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/javase/6/docs/
java document
Java
Download this Documentation
Done

JDK™ 6 Documentation

Legal Notices API, Language, and VM Specs Features Guides Release Notes Tool Docs Tutorials and Training

Java™ SE 6 Platform at a Glance

This document covers the Java™ Platform, Standard Edition 6 JDK. Its product version number is 6 and developer version number is 1.6.0, as described in Platform Name and Version Numbers. For programmer guides or information on a feature of the JDK, click on a component in


JDK 6 Documentation - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/javase/6/docs/
java document
Done

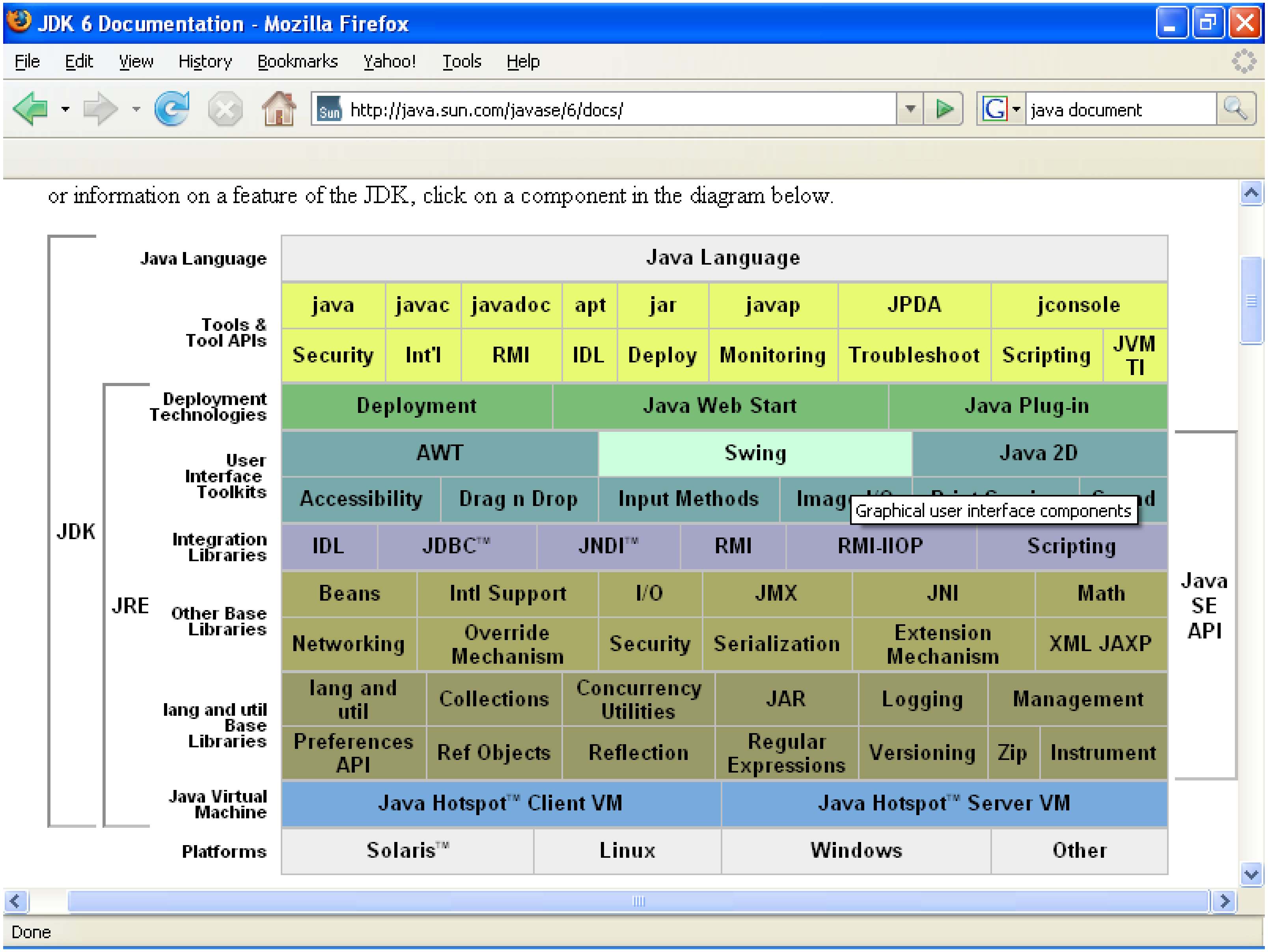
or information on a feature of the JDK, click on a component in the diagram below.

Java Language	Java Language
Tools & Tool APIs	java, javac, javadoc, apt, jar, javap, JPDA, jconsole Security, Int'l, RMI, IDL, Deploy, Monitoring, Troubleshoot, Scripting, JVM TI
Deployment Technologies	Deployment, Java Web Start, Java Plug-in
User Interface Toolkits	AWT, Swing, Java 2D Accessibility, Drag n Drop, Input Methods, Image I/O, Print Service, Sound
Integration Libraries	IDL, JDBC™, JNDI™, RMI, RMI-IIOP, Scripting
Other Base Libraries	Beans, Intl Support, I/O, JMX, JNI, Math Networking, Override Mechanism, Security, Serialization, Extension Mechanism, XML JAXP
lang and util Base Libraries	lang and util, Collections, Concurrency Utilities, JAR, Logging, Management Preferences API, Ref Objects, Reflection, Regular Expressions, Versioning, Zip, Instrument
Java Virtual Machine	Java Hotspot™ Client VM, Java Hotspot™ Server VM
Platforms	Solaris™, Linux, Windows, Other

JDK

JRE

Java SE API

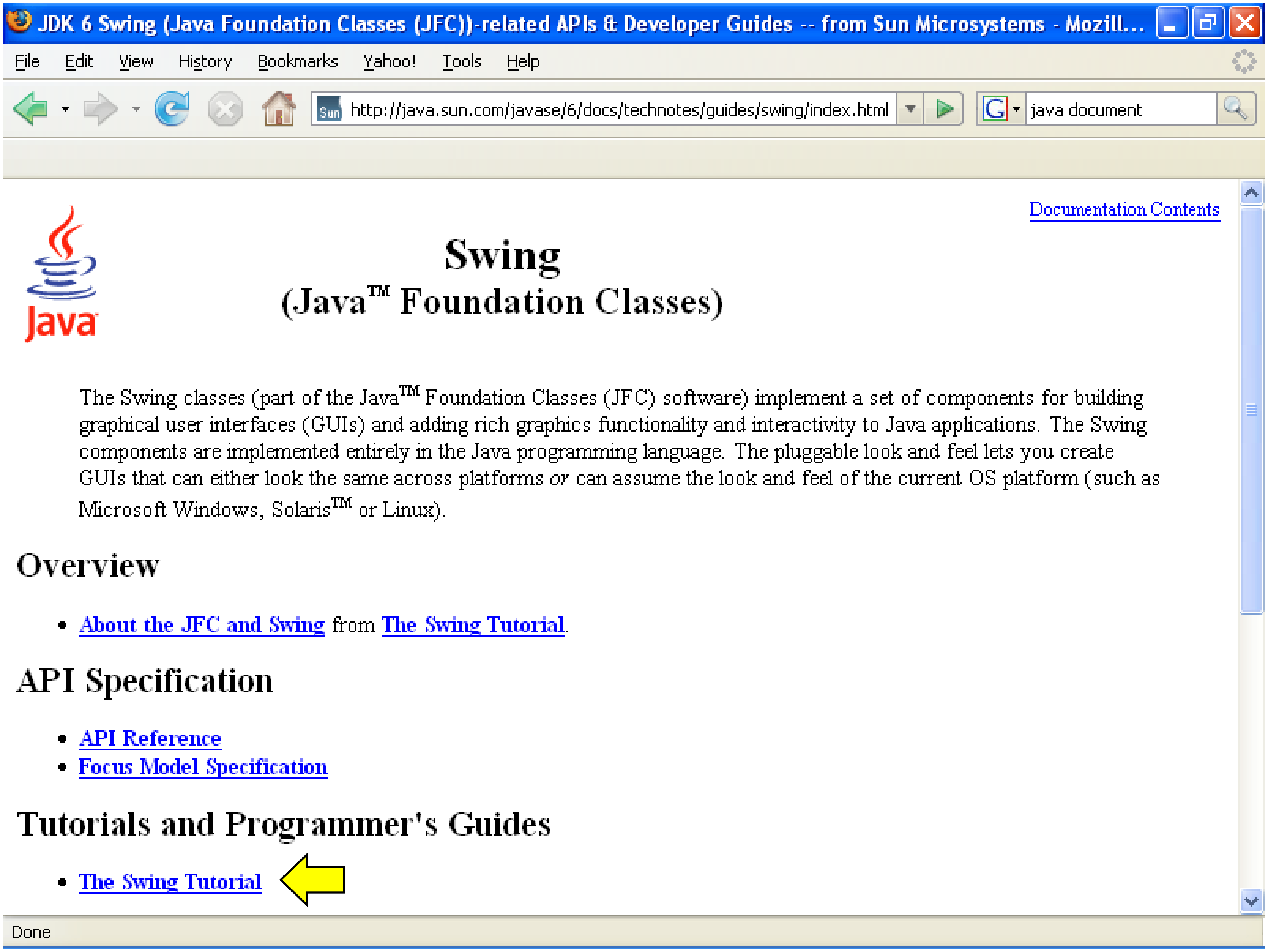


http://java.sun.com/javase/6/docs/

or information on a feature of the JDK, click on a component in the diagram below.

Java Language	Java Language
Tools & Tool APIs	java javac javadoc apt jar javap JPDA jconsole
	Security Int'l RMI IDL Deploy Monitoring Troubleshoot Scripting JVM TI
Deployment Technologies	Deployment Java Web Start Java Plug-in
User Interface Toolkits	AWT Swing Java 2D
	Accessibility Drag n Drop Input Methods Image I/O Print Service Sound
Integration Libraries	IDL JDBC™ JNDI™ RMI RMI-IIOP Scripting
Other Base Libraries	Beans Intl Support I/O JMX JNI Math
	Networking Override Mechanism Security Serialization Extension Mechanism XML JAXP
lang and util Base Libraries	lang and util Collections Concurrency Utilities JAR Logging Management
	Preferences API Ref Objects Reflection Regular Expressions Versioning Zip Instrument
Java Virtual Machine	Java Hotspot™ Client VM Java Hotspot™ Server VM
Platforms	Solaris™ Linux Windows Other

Graphical user interface components


JDK 6 Swing (Java Foundation Classes (JFC))-related APIs & Developer Guides -- from Sun Microsystems - Mozill...
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/javase/6/docs/technotes/guides/swing/index.html
java document
Java
Done

Documentation Contents

Swing
(Java™ Foundation Classes)

The Swing classes (part of the Java™ Foundation Classes (JFC) software) implement a set of components for building graphical user interfaces (GUIs) and adding rich graphics functionality and interactivity to Java applications. The Swing components are implemented entirely in the Java programming language. The pluggable look and feel lets you create GUIs that can either look the same across platforms *or* can assume the look and feel of the current OS platform (such as Microsoft Windows, Solaris™ or Linux).

Overview

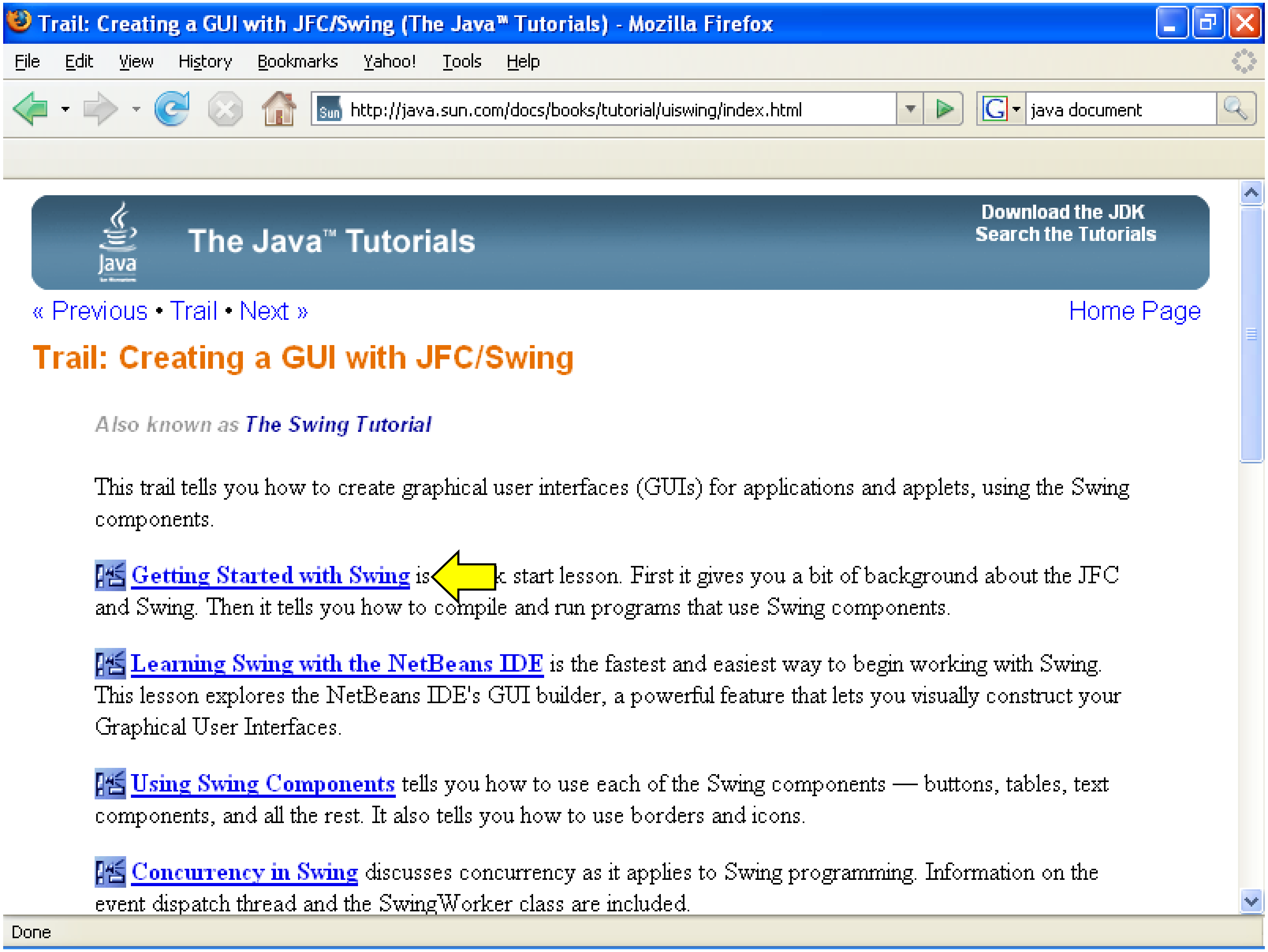
- About the JFC and Swing from The Swing Tutorial.

API Specification

- API Reference
- Focus Model Specification

Tutorials and Programmer's Guides

- The Swing Tutorial



http://java.sun.com/docs/books/tutorial/uiswing/index.html

java document

The Java™ Tutorials

« Previous • Trail • Next »   Home Page

Trail: Creating a GUI with JFC/Swing

*Also known as **The Swing Tutorial***

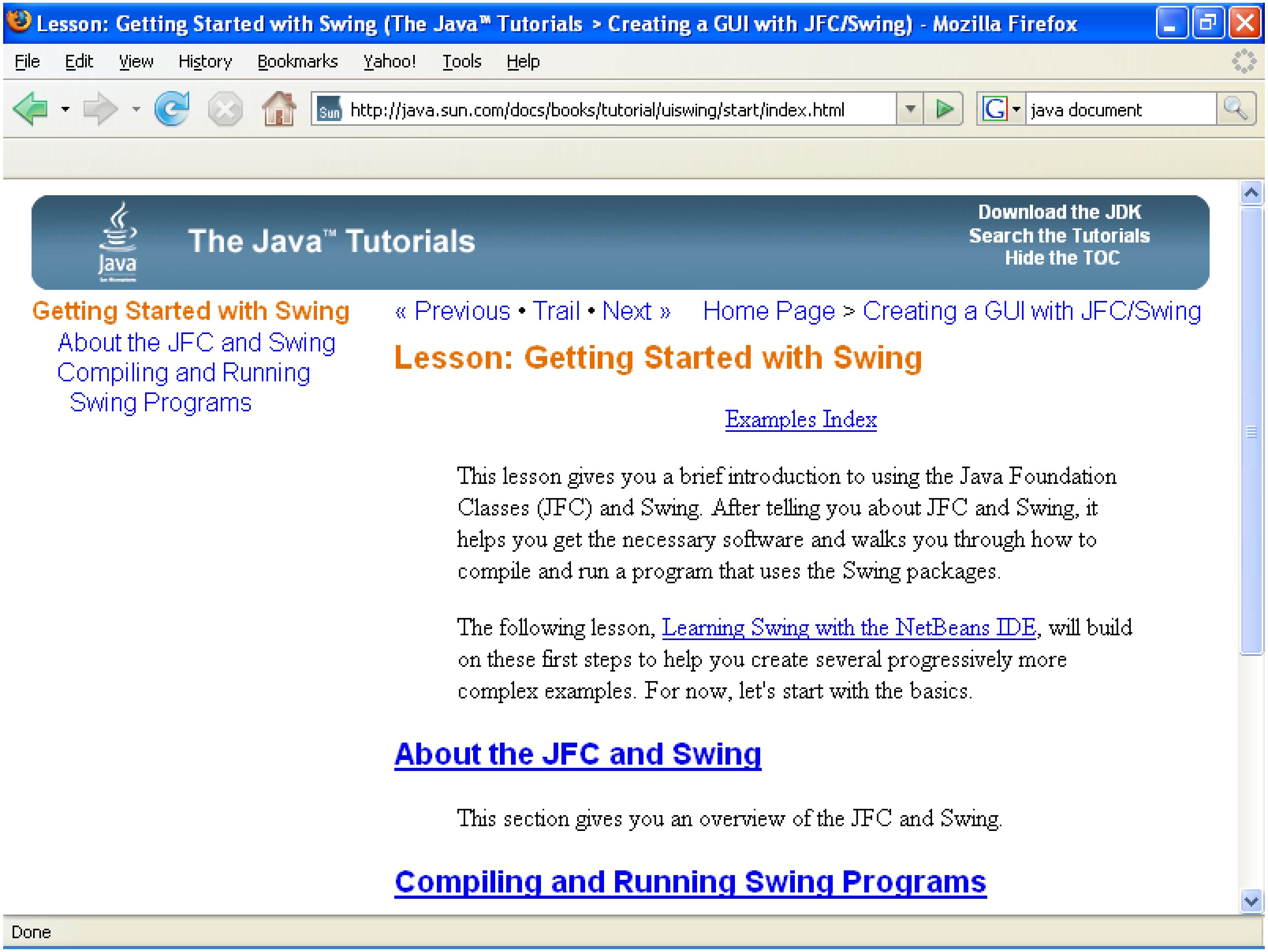
This trail tells you how to create graphical user interfaces (GUIs) for applications and applets, using the Swing components.

Getting Started with Swing is k start lesson. First it gives you a bit of background about the JFC and Swing. Then it tells you how to compile and run programs that use Swing components.

Learning Swing with the NetBeans IDE is the fastest and easiest way to begin working with Swing. This lesson explores the NetBeans IDE's GUI builder, a powerful feature that lets you visually construct your Graphical User Interfaces.

Using Swing Components tells you how to use each of the Swing components — buttons, tables, text components, and all the rest. It also tells you how to use borders and icons.

Concurrency in Swing discusses concurrency as it applies to Swing programming. Information on the event dispatch thread and the SwingWorker class are included.


Lesson: Getting Started with Swing (The Java™ Tutorials > Creating a GUI with JFC/Swing) - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/docs/books/tutorial/uiswing/start/index.html
java document
The Java™ Tutorials
Download the JDK
Search the Tutorials
Hide the TOC
Getting Started with Swing
About the JFC and Swing
Compiling and Running Swing Programs
« Previous • Trail • Next » Home Page > Creating a GUI with JFC/Swing
Done

Lesson: Getting Started with Swing

Examples Index

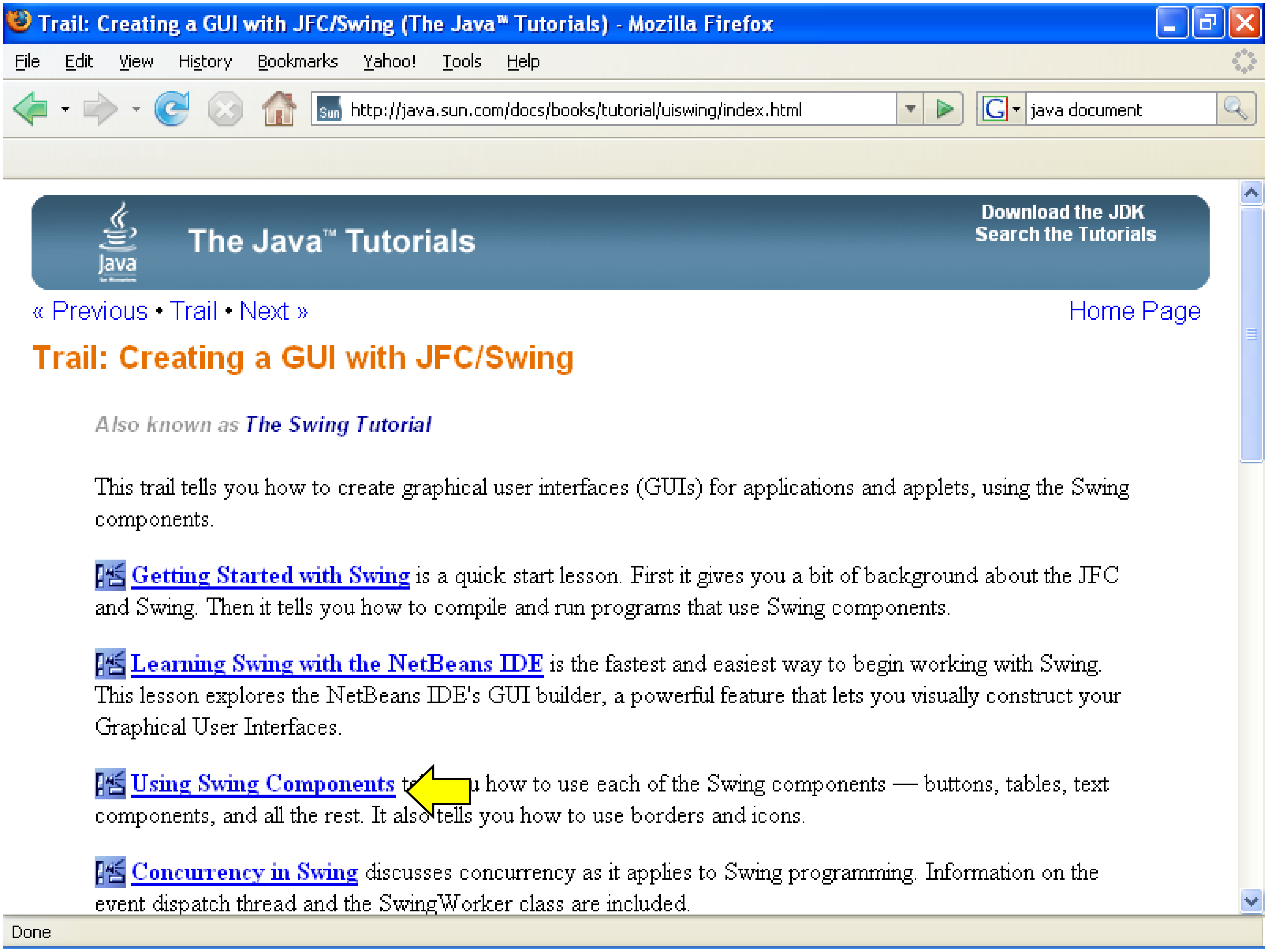
This lesson gives you a brief introduction to using the Java Foundation Classes (JFC) and Swing. After telling you about JFC and Swing, it helps you get the necessary software and walks you through how to compile and run a program that uses the Swing packages.

The following lesson, Learning Swing with the NetBeans IDE, will build on these first steps to help you create several progressively more complex examples. For now, let's start with the basics.

About the JFC and Swing

This section gives you an overview of the JFC and Swing.

Compiling and Running Swing Programs


Trail: Creating a GUI with JFC/Swing (The Java™ Tutorials) - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/docs/books/tutorial/uiswing/index.html
java document
The Java™ Tutorials
Download the JDK
Search the Tutorials
« Previous • Trail • Next »
Home Page
Done

Trail: Creating a GUI with JFC/Swing

Also known as ***The Swing Tutorial***

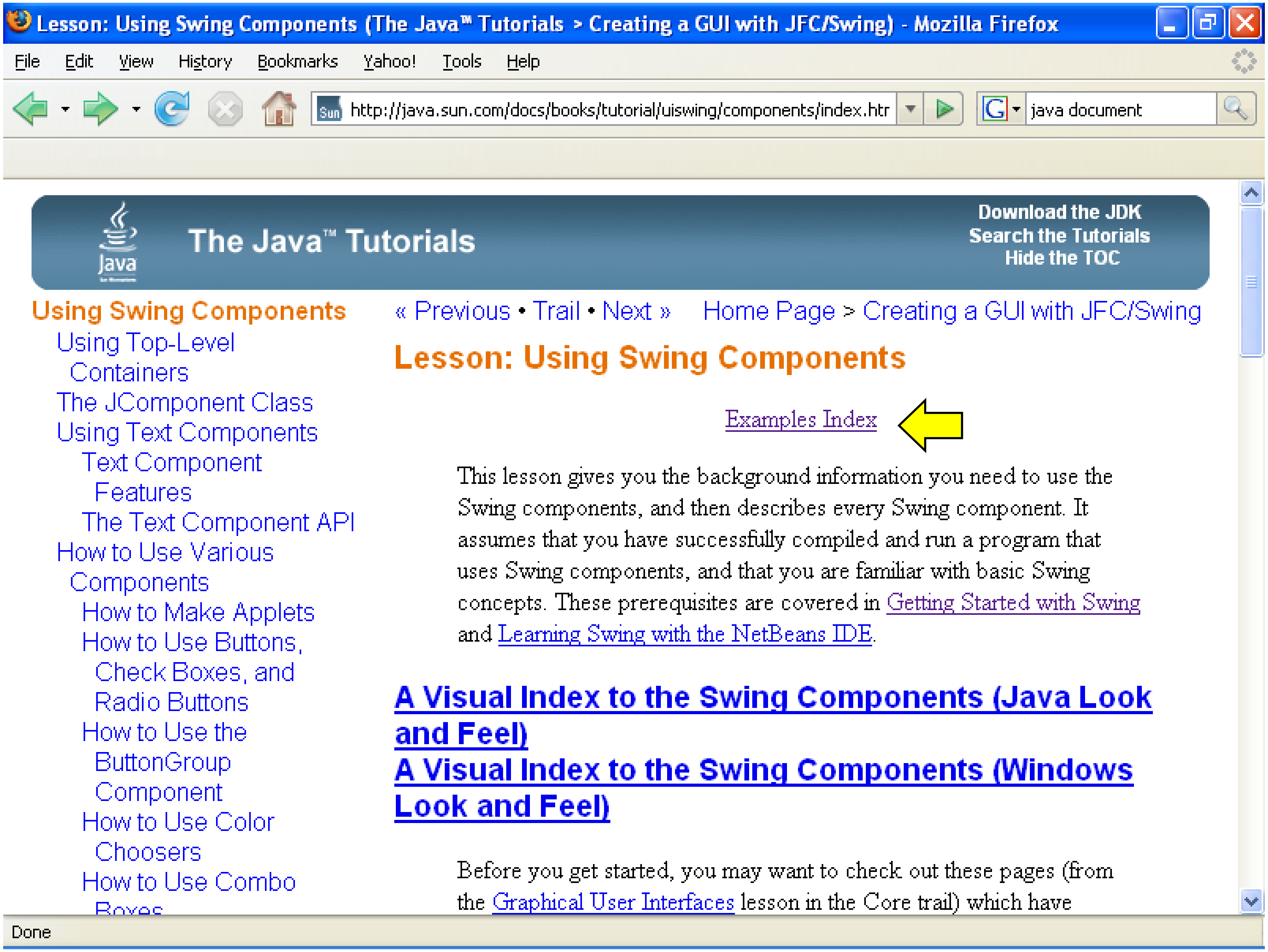
This trail tells you how to create graphical user interfaces (GUIs) for applications and applets, using the Swing components.

Getting Started with Swing is a quick start lesson. First it gives you a bit of background about the JFC and Swing. Then it tells you how to compile and run programs that use Swing components.

Learning Swing with the NetBeans IDE is the fastest and easiest way to begin working with Swing. This lesson explores the NetBeans IDE's GUI builder, a powerful feature that lets you visually construct your Graphical User Interfaces.

Using Swing Components t u how to use each of the Swing components — buttons, tables, text components, and all the rest. It also tells you how to use borders and icons.

Concurrency in Swing discusses concurrency as it applies to Swing programming. Information on the event dispatch thread and the SwingWorker class are included.


Lesson: Using Swing Components (The Java™ Tutorials > Creating a GUI with JFC/Swing) - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/docs/books/tutorial/uiswing/components/index.htr
java document
The Java™ Tutorials
Download the JDK
Search the Tutorials
Hide the TOC
Done

Using Swing Components
Using Top-Level Containers
The JComponent Class
Using Text Components
Text Component Features
The Text Component API
How to Use Various Components
How to Make Applets
How to Use Buttons, Check Boxes, and Radio Buttons
How to Use the ButtonGroup Component
How to Use Color Choosers
How to Use Combo Boxes

« Previous • Trail • Next » Home Page > Creating a GUI with JFC/Swing

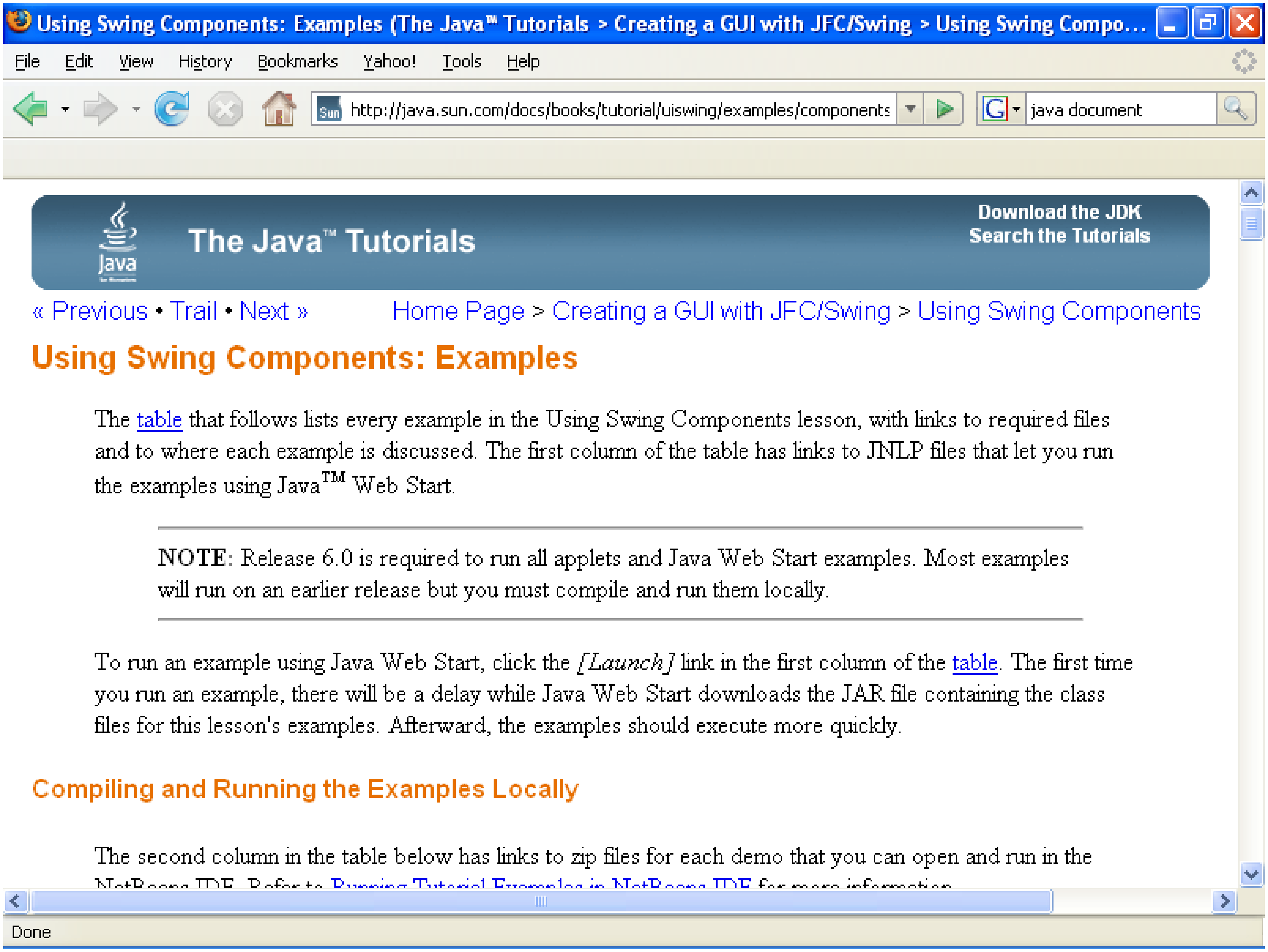
Lesson: Using Swing Components

Examples Index

This lesson gives you the background information you need to use the Swing components, and then describes every Swing component. It assumes that you have successfully compiled and run a program that uses Swing components, and that you are familiar with basic Swing concepts. These prerequisites are covered in Getting Started with Swing and Learning Swing with the NetBeans IDE.

A Visual Index to the Swing Components (Java Look and Feel)
A Visual Index to the Swing Components (Windows Look and Feel)

Before you get started, you may want to check out these pages (from the Graphical User Interfaces lesson in the Core trail) which have


Using Swing Components: Examples (The Java™ Tutorials > Creating a GUI with JFC/Swing > Using Swing Compo...
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/docs/books/tutorial/uiswing/examples/components
java document
The Java™ Tutorials
Download the JDK
Search the Tutorials
« Previous • Trail • Next »
Home Page > Creating a GUI with JFC/Swing > Using Swing Components
Done

Using Swing Components: Examples

The table that follows lists every example in the Using Swing Components lesson, with links to required files and to where each example is discussed. The first column of the table has links to JNLP files that let you run the examples using Java™ Web Start.

NOTE: Release 6.0 is required to run all applets and Java Web Start examples. Most examples will run on an earlier release but you must compile and run them locally.

To run an example using Java Web Start, click the *[Launch]* link in the first column of the table. The first time you run an example, there will be a delay while Java Web Start downloads the JAR file containing the class files for this lesson's examples. Afterward, the examples should execute more quickly.

Compiling and Running the Examples Locally

The second column in the table below has links to zip files for each demo that you can open and run in the NetBeans IDE. Refer to Running Tutorial Examples in NetBeans IDE for more information.

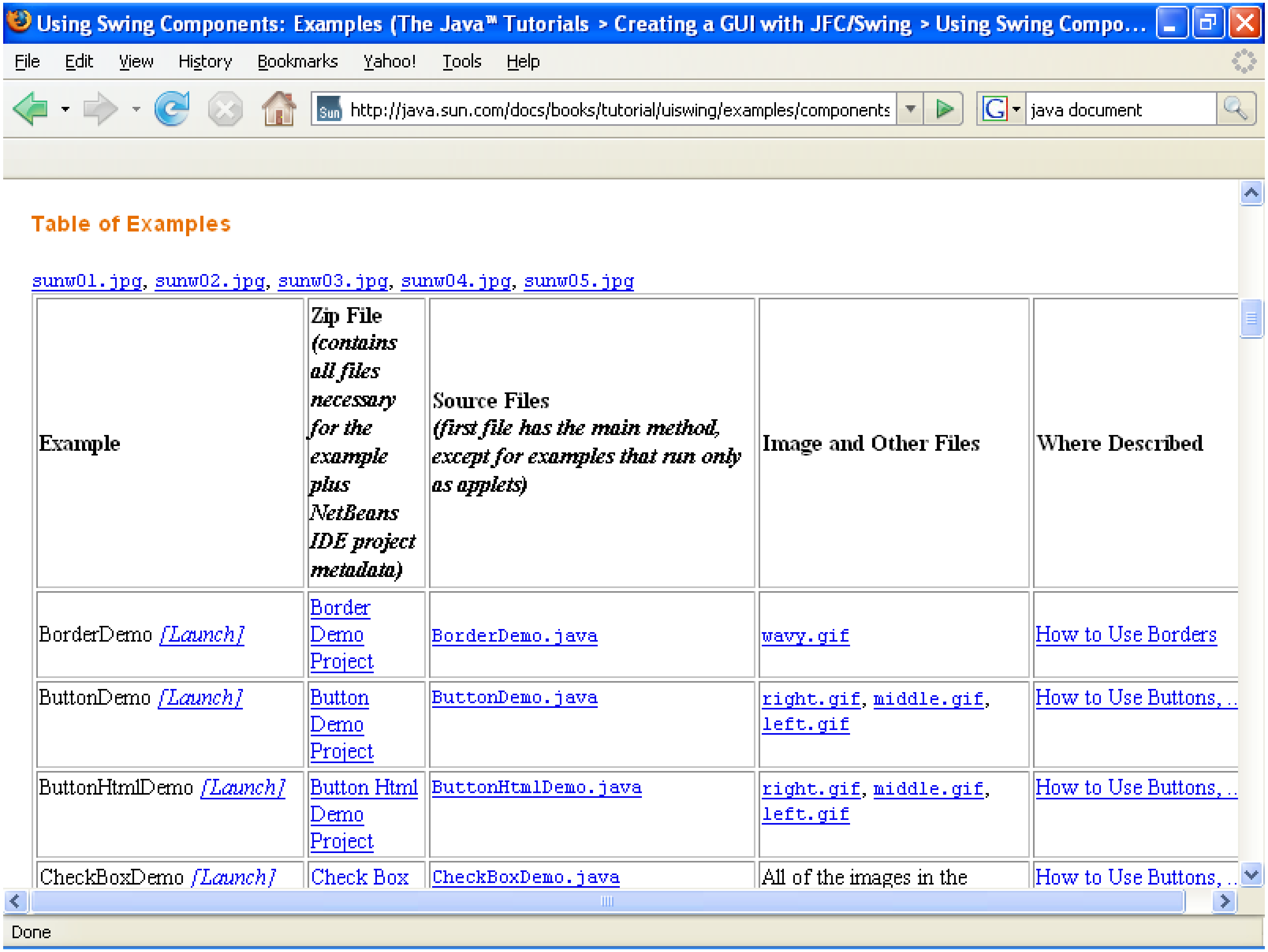
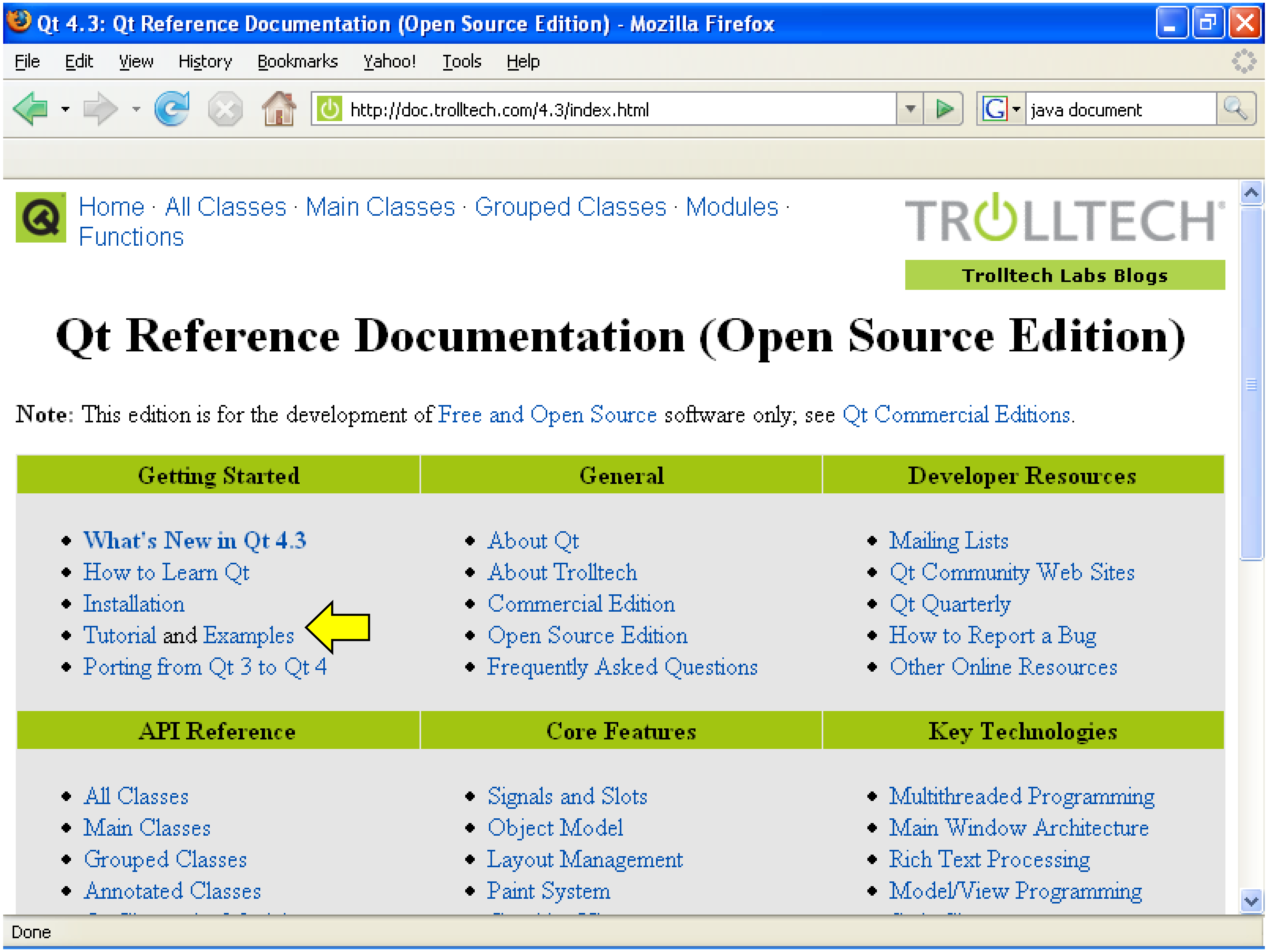

Using Swing Components: Examples (The Java™ Tutorials > Creating a GUI with JFC/Swing > Using Swing Compo...
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/docs/books/tutorial/uiswing/examples/components
java document
Done

Table of Examples

sunw01.jpg, sunw02.jpg, sunw03.jpg, sunw04.jpg, sunw05.jpg

Example	Zip File *(contains all files necessary for the example plus NetBeans IDE project metadata)*	Source Files *(first file has the main method, except for examples that run only as applets)*	Image and Other Files	Where Described
BorderDemo *[Launch]*	Border Demo Project	BorderDemo.java	wavy.gif	How to Use Borders
ButtonDemo *[Launch]*	Button Demo Project	ButtonDemo.java	right.gif, middle.gif, left.gif	How to Use Buttons, ..
ButtonHtmlDemo *[Launch]*	Button Html Demo Project	ButtonHtmlDemo.java	right.gif, middle.gif, left.gif	How to Use Buttons, ..
CheckBoxDemo *[Launch]*	Check Box	CheckBoxDemo.java	All of the images in the	How to Use Buttons, ..


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
java document
TROLLTECH
Trolltech Labs Blogs
Done

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

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

- **What's New in Qt 4.3**
- How to Learn Qt
- Installation
- Tutorial and Examples
- Porting from Qt 3 to Qt 4

General

- About Qt
- About Trolltech
- Commercial Edition
- Open Source Edition
- Frequently Asked Questions

Developer Resources

- Mailing Lists
- Qt Community Web Sites
- Qt Quarterly
- How to Report a Bug
- Other Online Resources

API Reference

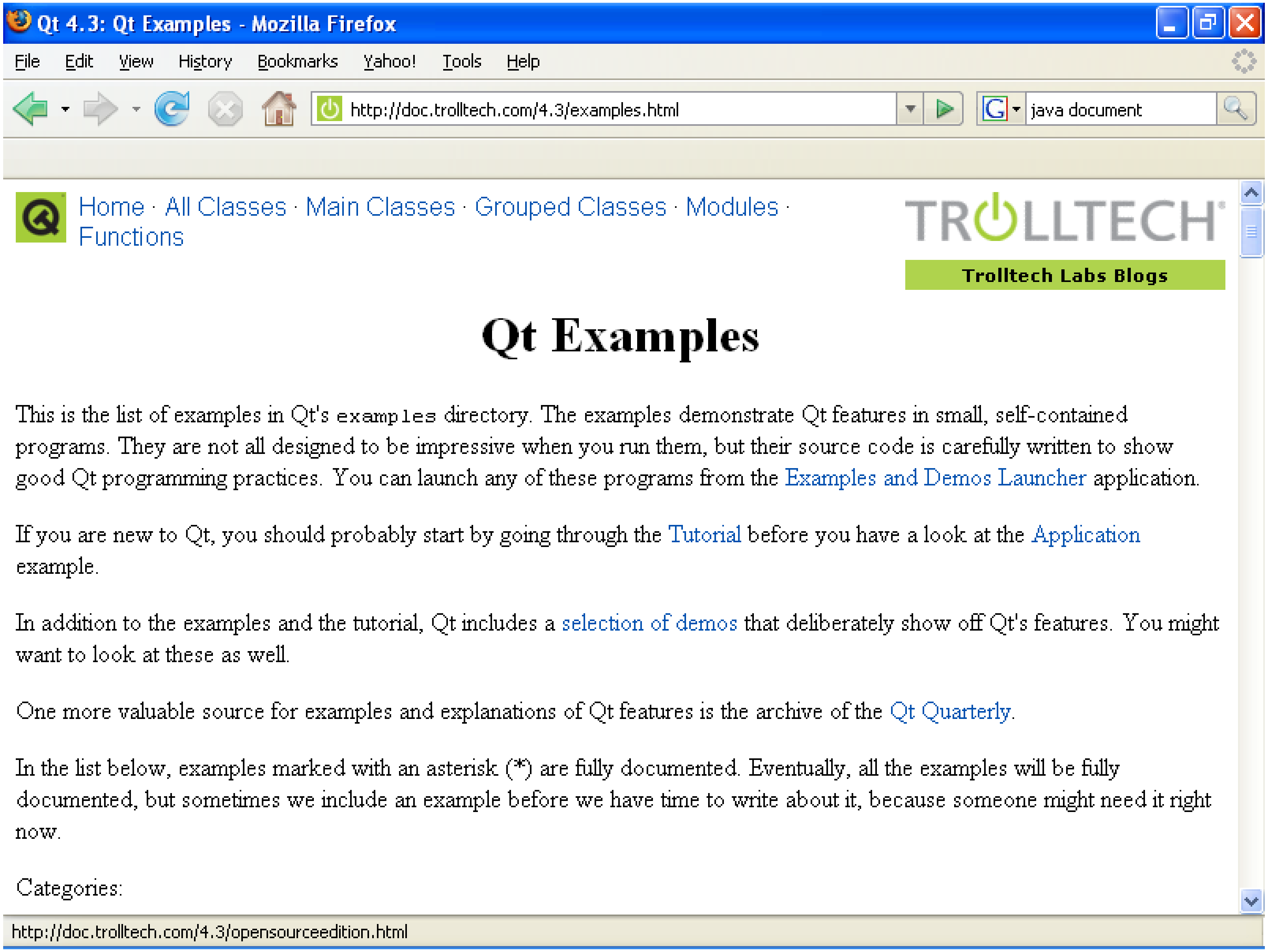
- All Classes
- Main Classes
- Grouped Classes
- Annotated Classes

Core Features

- Signals and Slots
- Object Model
- Layout Management
- Paint System

Key Technologies

- Multithreaded Programming
- Main Window Architecture
- Rich Text Processing
- Model/View Programming


Qt 4.3: Qt Examples - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/examples.html
java document
Home · All Classes · Main Classes · Grouped Classes · Modules · Functions
TROLLTECH
Trolltech Labs Blogs
http://doc.trolltech.com/4.3/opensourceedition.html

Qt Examples

This is the list of examples in Qt's `examples` directory. The examples demonstrate Qt features in small, self-contained programs. They are not all designed to be impressive when you run them, but their source code is carefully written to show good Qt programming practices. You can launch any of these programs from the Examples and Demos Launcher application.

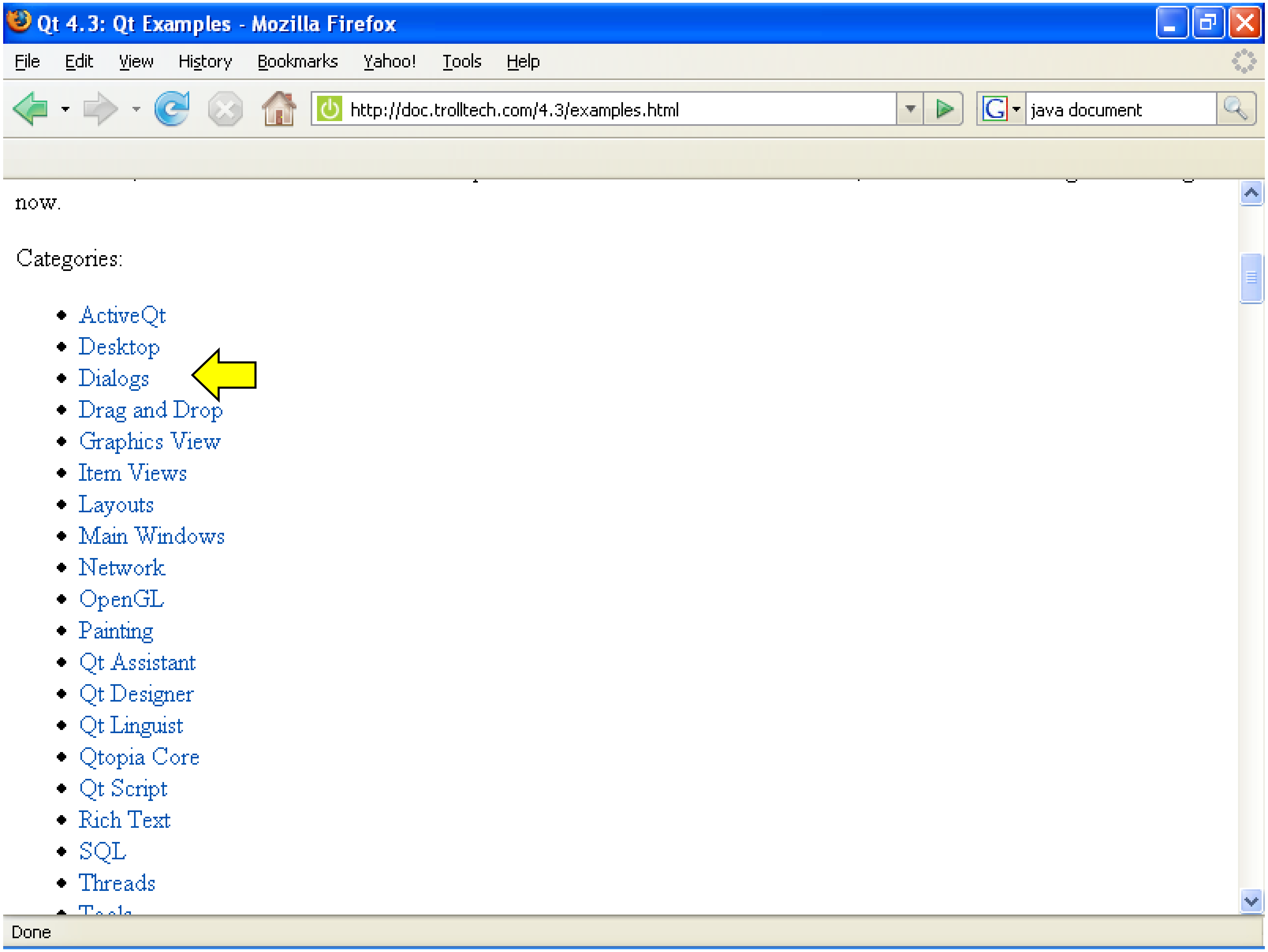
If you are new to Qt, you should probably start by going through the Tutorial before you have a look at the Application example.

In addition to the examples and the tutorial, Qt includes a selection of demos that deliberately show off Qt's features. You might want to look at these as well.

One more valuable source for examples and explanations of Qt features is the archive of the Qt Quarterly.

In the list below, examples marked with an asterisk (*) are fully documented. Eventually, all the examples will be fully documented, but sometimes we include an example before we have time to write about it, because someone might need it right now.

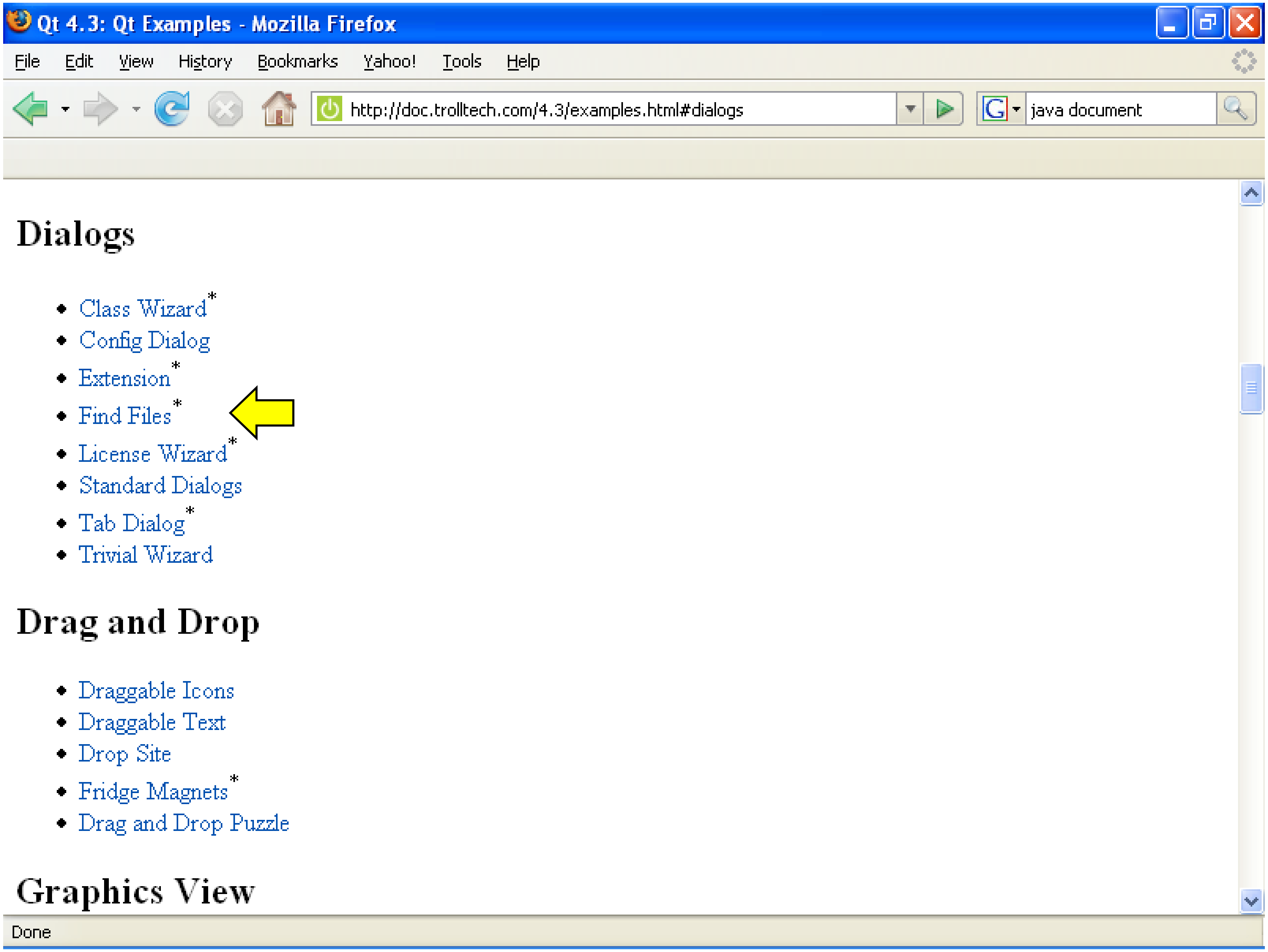
Categories:


Qt 4.3: Qt Examples - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/examples.html
java document
Done

now.

Categories:

- ActiveQt
- Desktop
- Dialogs
- Drag and Drop
- Graphics View
- Item Views
- Layouts
- Main Windows
- Network
- OpenGL
- Painting
- Qt Assistant
- Qt Designer
- Qt Linguist
- Qtopia Core
- Qt Script
- Rich Text
- SQL
- Threads
- Tools


Qt 4.3: Qt Examples - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/examples.html#dialogs
java document
Done

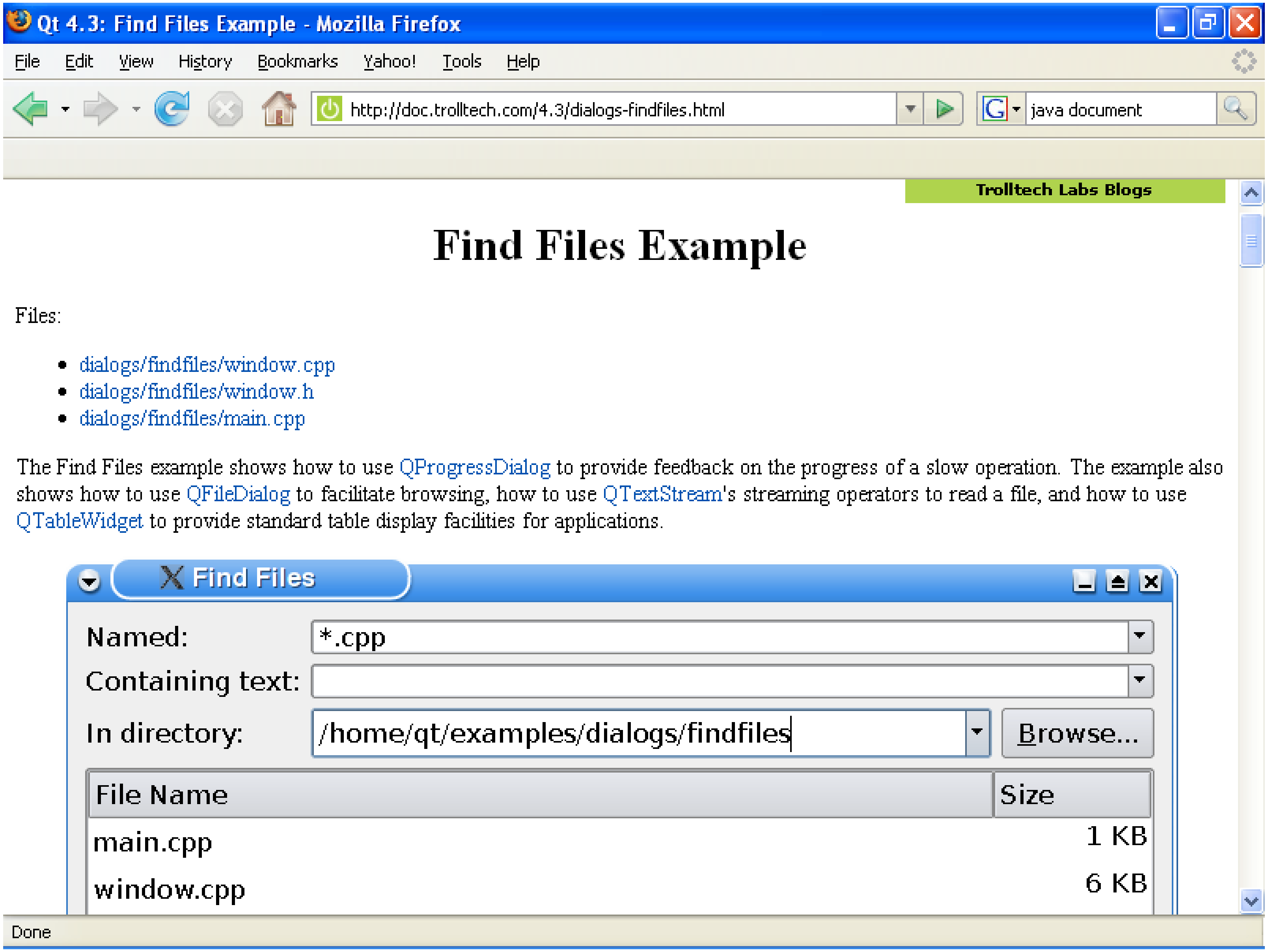
Dialogs

- Class Wizard*
- Config Dialog
- Extension*
- Find Files*
- License Wizard*
- Standard Dialogs
- Tab Dialog*
- Trivial Wizard

Drag and Drop

- Draggable Icons
- Draggable Text
- Drop Site
- Fridge Magnets*
- Drag and Drop Puzzle

Graphics View


Qt 4.3: Find Files Example - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://doc.trolltech.com/4.3/dialogs-findfiles.html
java document
Trolltech Labs Blogs

Find Files Example

Files:

- dialogs/findfiles/window.cpp
- dialogs/findfiles/window.h
- dialogs/findfiles/main.cpp

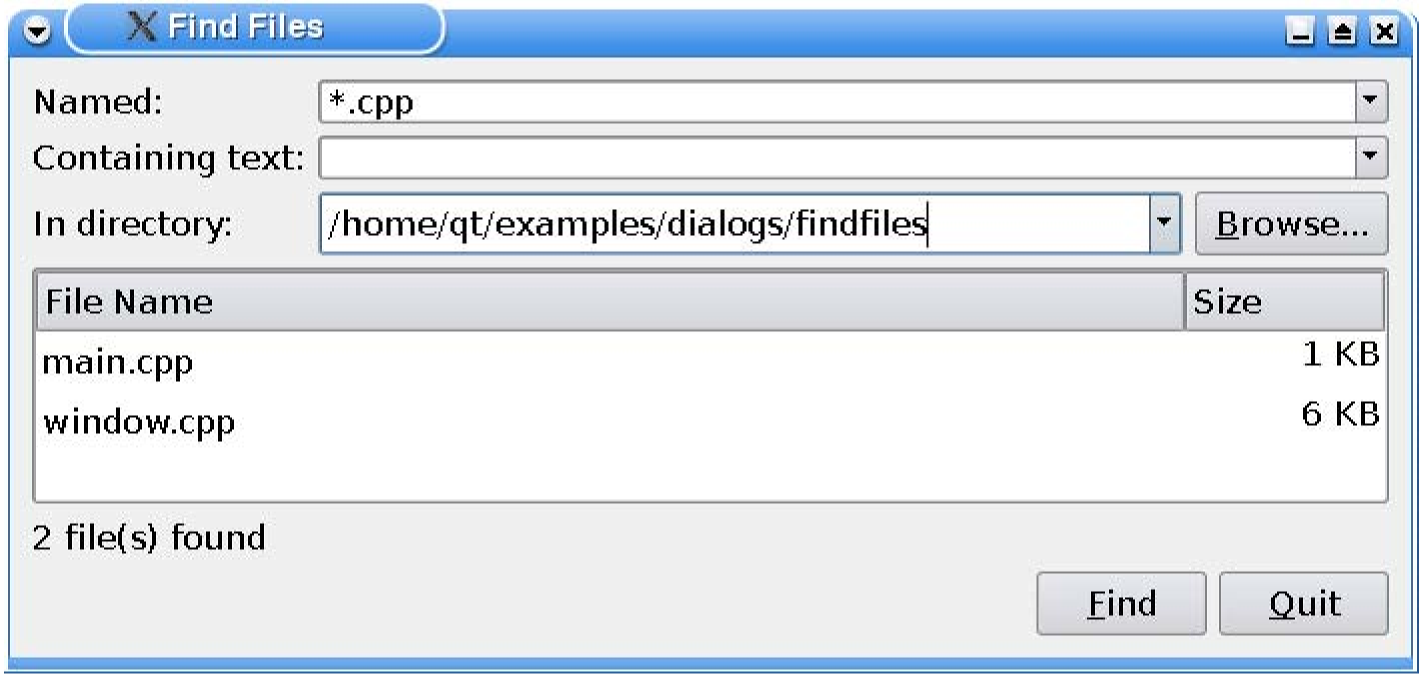
The Find Files example shows how to use QProgressDialog to provide feedback on the progress of a slow operation. The example also shows how to use QFileDialog to facilitate browsing, how to use QTextStream's streaming operators to read a file, and how to use QTableWidget to provide standard table display facilities for applications.

Named: *.cpp

Containing text:

In directory: /home/qt/examples/dialogs/findfiles

File Name	Size
main.cpp	1 KB
window.cpp	6 KB



Named: *.cpp

Containing text:

In directory: /home/qt/examples/dialogs/findfiles

File Name	Size
main.cpp	1 KB
window.cpp	6 KB

2 file(s) found

ECE 462
Object-Oriented Programming using C++ and Java

Development Environment

Yung-Hsiang Lu

yunglu@purdue.edu

Demonstrations

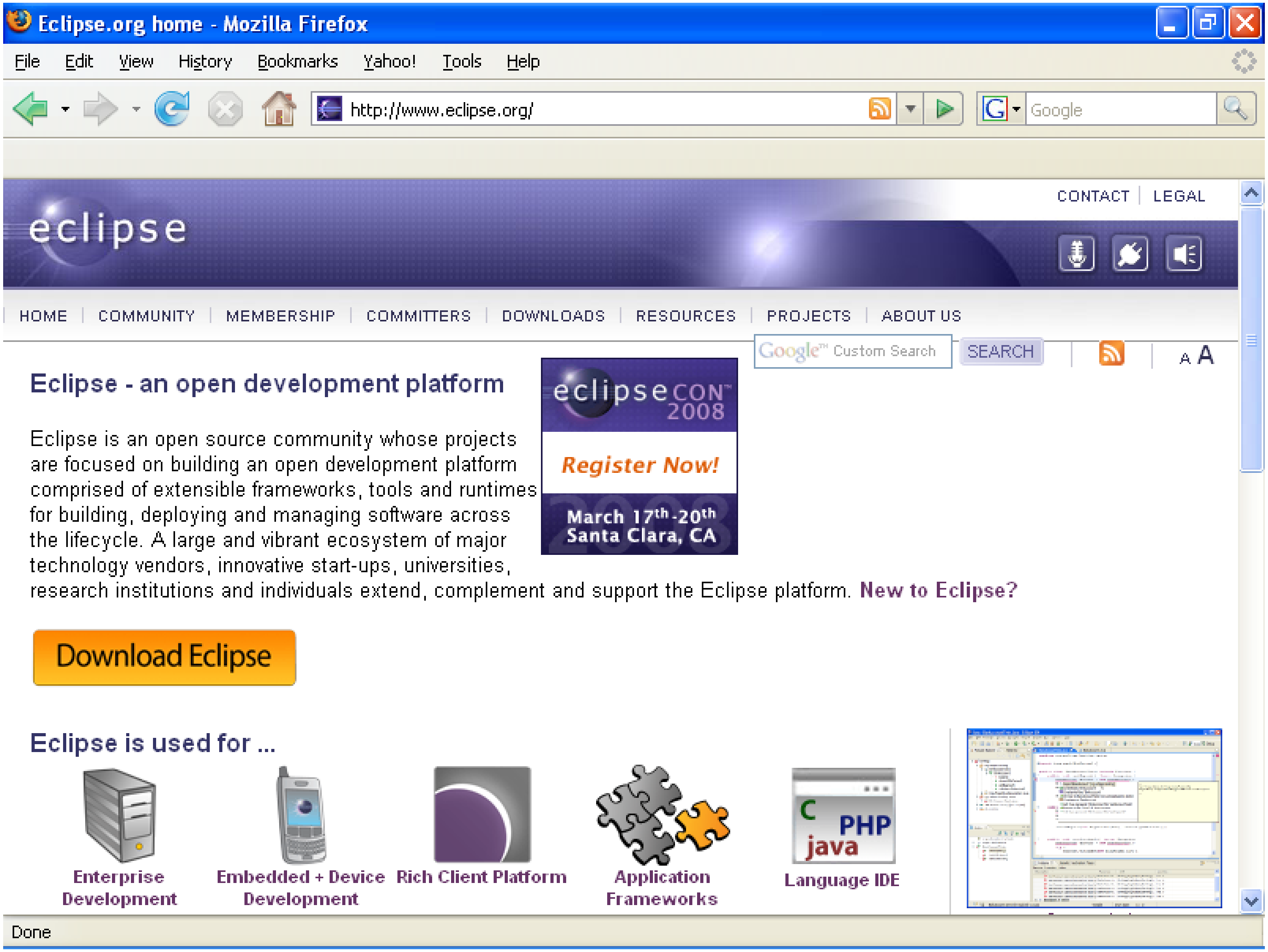
- develop C++ project in Eclipse
- develop Java project in Eclipse
- develop Java project in Netbeans (in handout)
- compile / execute C++ program in Linux shell
- compile / execute Java program in Linux shell

- Program: Person and Student classes.

Set up Execution Environment

- You can use Netbeans or Eclipse for developing Java or C++ projects.
- Please remember to use your ee462xxx account in MSEE190. Your personal Purdue account will not work.
- Do not use the ee462xxx account for any other purpose.
- After the final exam, **the account will be erased and the password will be reset**.

Install Eclipse at Your Own Computer


Eclipse.org home - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://www.eclipse.org/
Google
eclipse
CONTACT | LEGAL
HOME | COMMUNITY | MEMBERSHIP | COMMITTERS | DOWNLOADS | RESOURCES | PROJECTS | ABOUT US
Google™ Custom Search
SEARCH
A A
eclipseCON™ 2008
Register Now!
March 17th-20th
Santa Clara, CA
Download Eclipse
C
PHP
java
Done

Eclipse - an open development platform

Eclipse is an open source community whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. A large and vibrant ecosystem of major technology vendors, innovative start-ups, universities, research institutions and individuals extend, complement and support the Eclipse platform. **New to Eclipse?**

Eclipse is used for ...

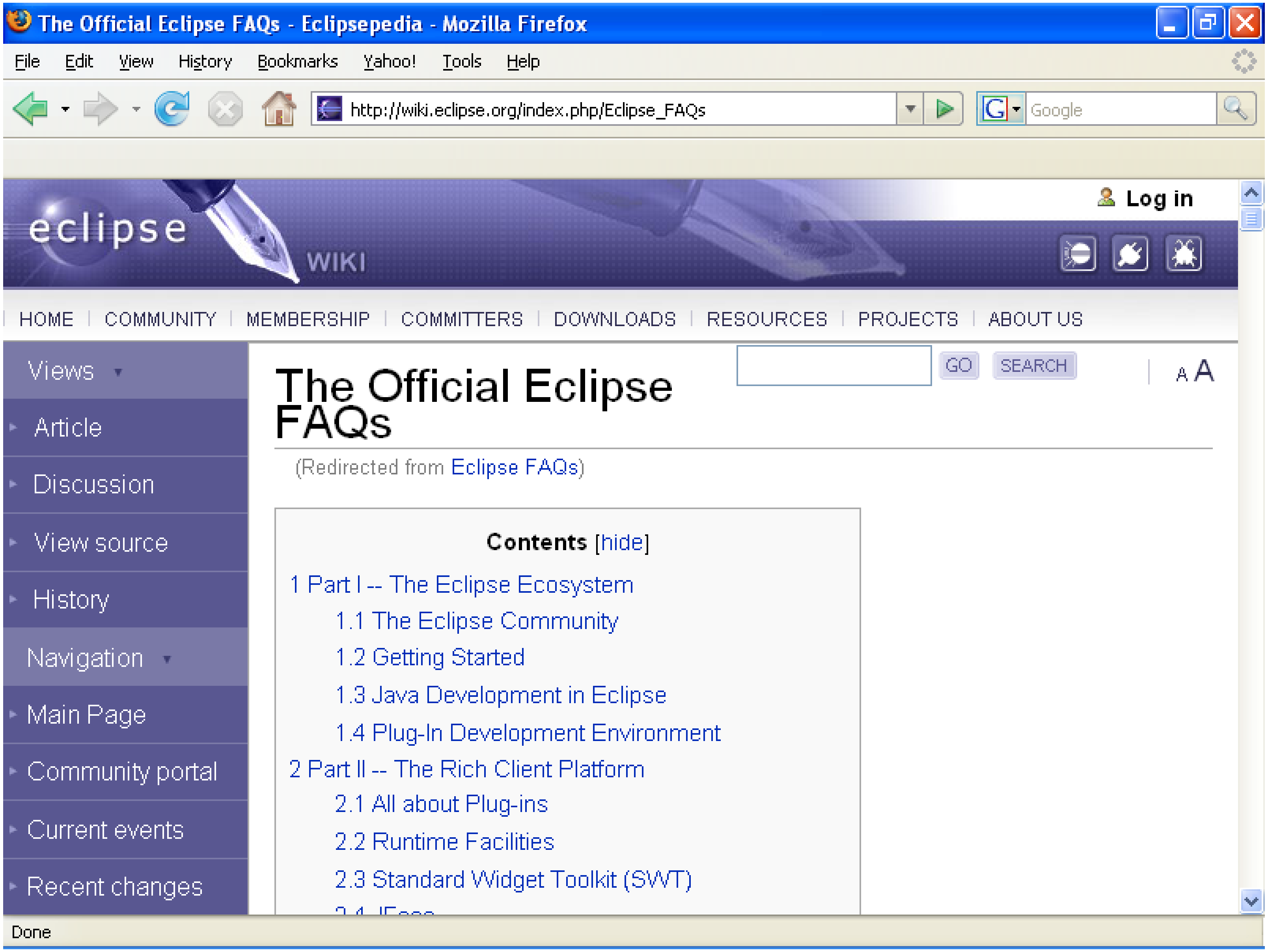
Enterprise Development

Embedded + Device Development

Rich Client Platform

Application Frameworks

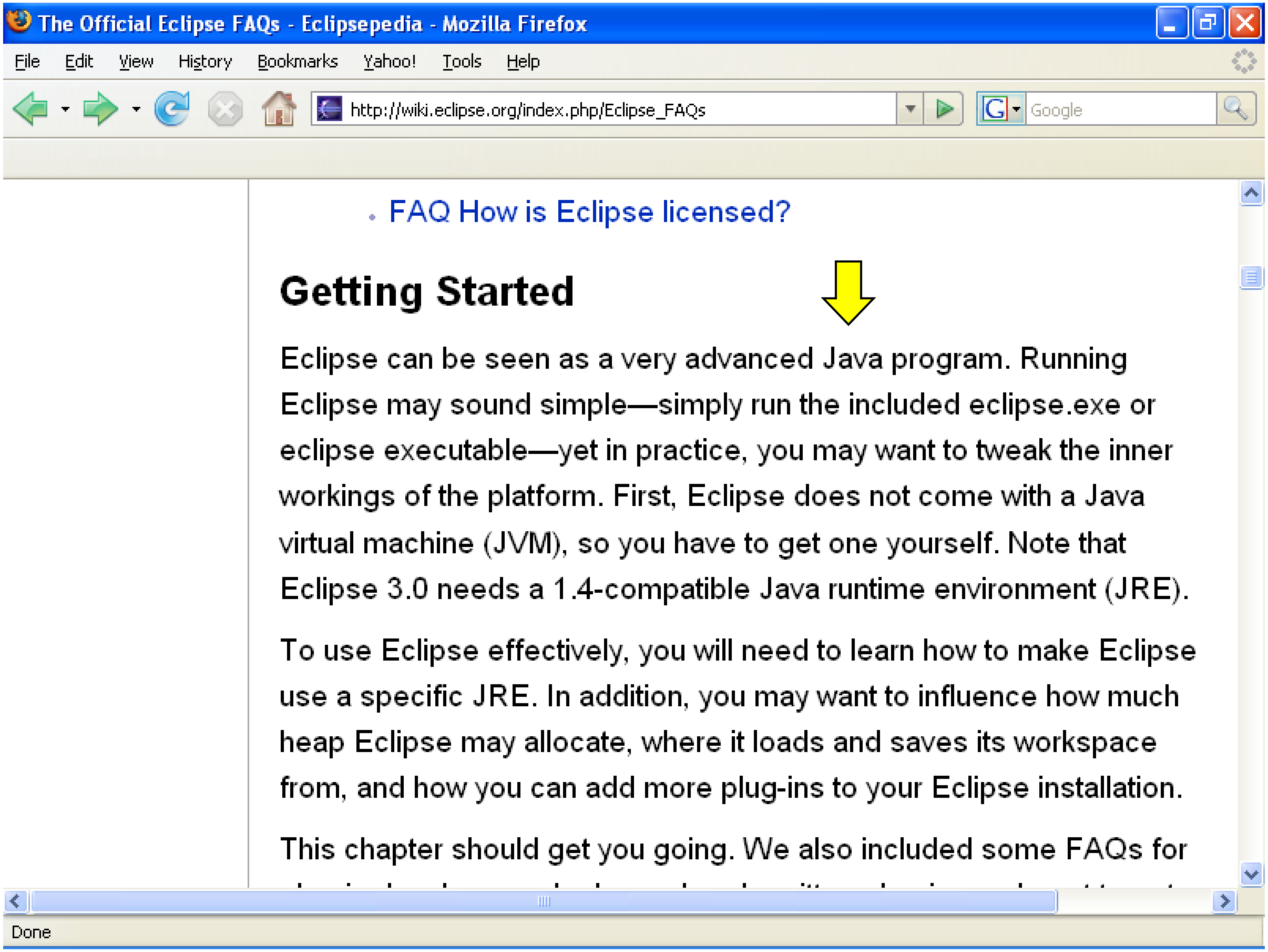
Language IDE



The Official Eclipse FAQs

(Redirected from Eclipse FAQs)

Contents [hide]


The Official Eclipse FAQs - Eclipsepedia - Mozilla Firefox
File
Edit
View
History
Bookmarks
Yahoo!
Tools
Help
http://wiki.eclipse.org/index.php/Eclipse_FAQs
Google
Done

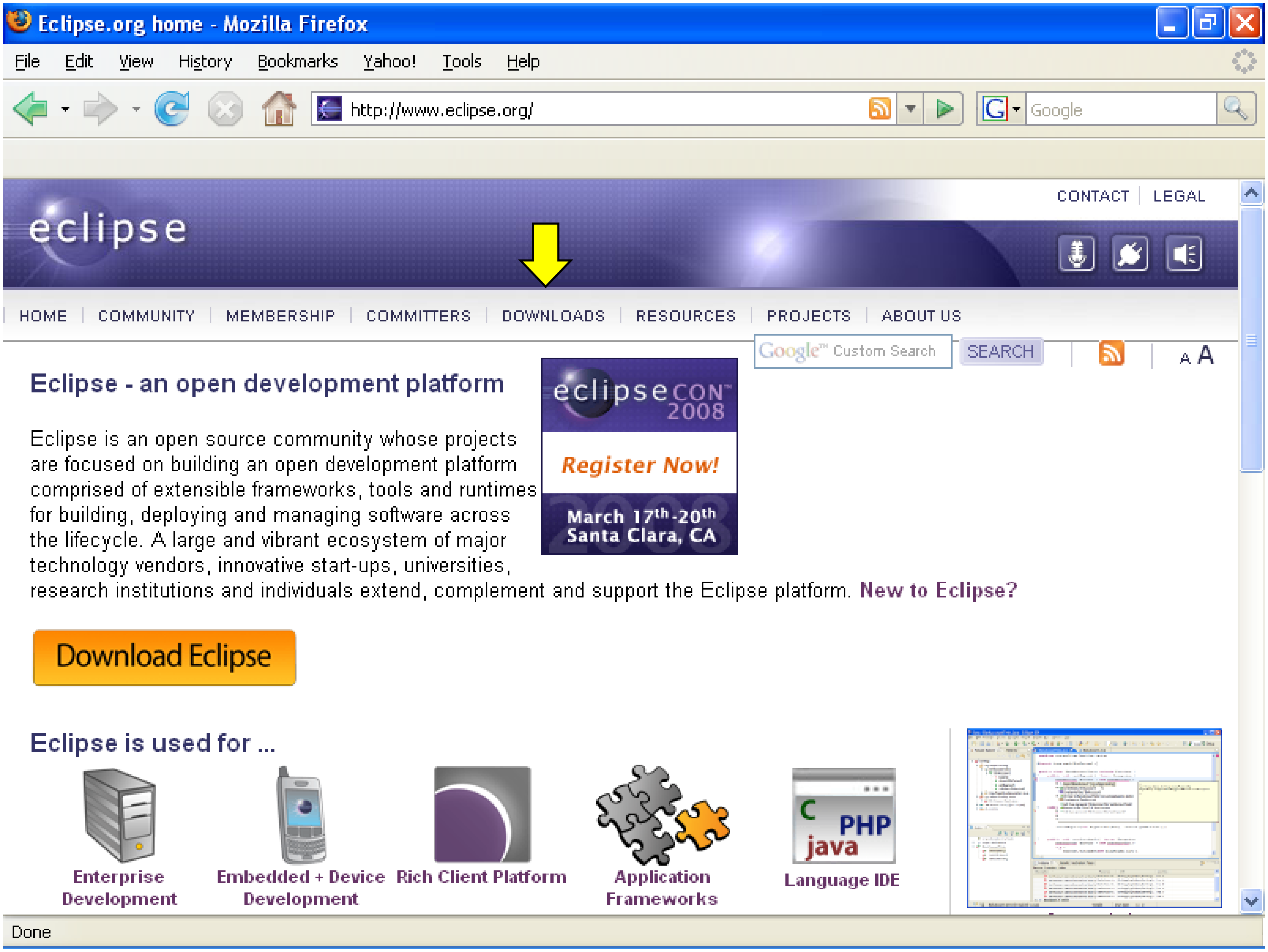
- FAQ How is Eclipse licensed?

Getting Started

Eclipse can be seen as a very advanced Java program. Running Eclipse may sound simple—simply run the included eclipse.exe or eclipse executable—yet in practice, you may want to tweak the inner workings of the platform. First, Eclipse does not come with a Java virtual machine (JVM), so you have to get one yourself. Note that Eclipse 3.0 needs a 1.4-compatible Java runtime environment (JRE).

To use Eclipse effectively, you will need to learn how to make Eclipse use a specific JRE. In addition, you may want to influence how much heap Eclipse may allocate, where it loads and saves its workspace from, and how you can add more plug-ins to your Eclipse installation.

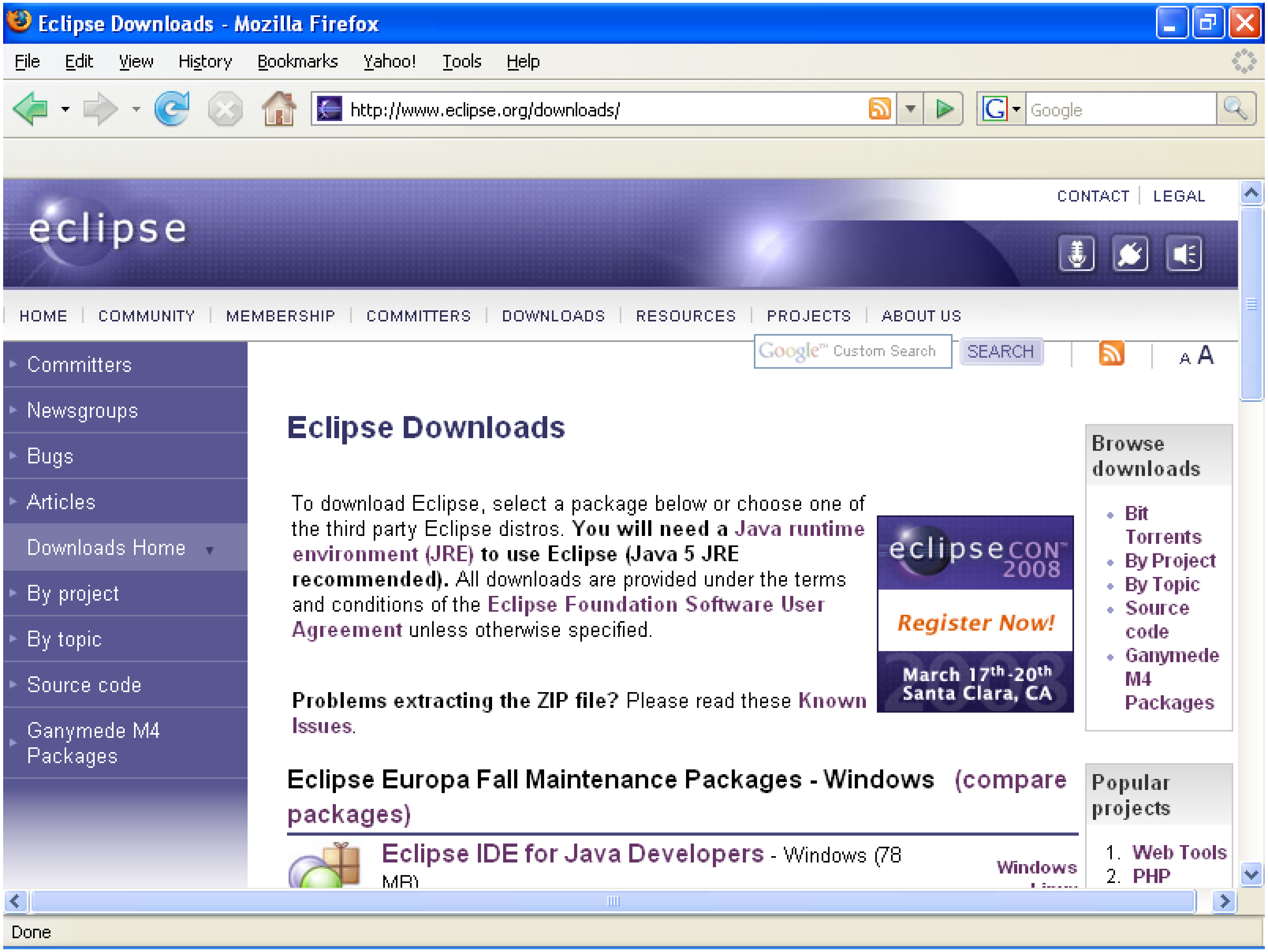
This chapter should get you going. We also included some FAQs for



Eclipse - an open development platform

Eclipse is an open source community whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. A large and vibrant ecosystem of major technology vendors, innovative start-ups, universities, research institutions and individuals extend, complement and support the Eclipse platform. New to Eclipse?

Eclipse is used for ...



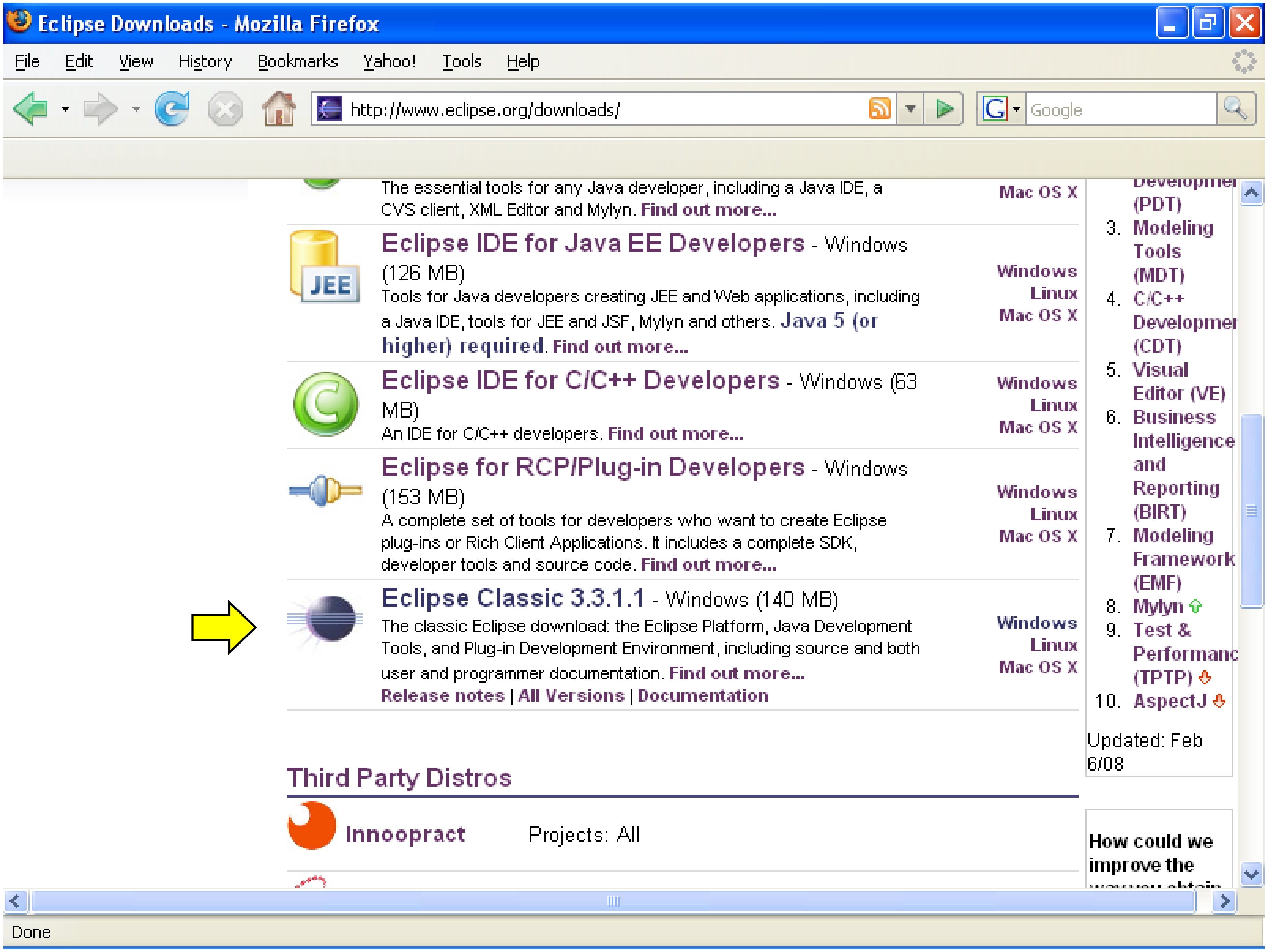
Eclipse Downloads

To download Eclipse, select a package below or choose one of the third party Eclipse distros. **You will need a Java runtime environment (JRE) to use Eclipse (Java 5 JRE recommended).** All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Problems extracting the ZIP file? Please read these Known Issues.

Eclipse Europa Fall Maintenance Packages - Windows (compare packages)

Eclipse IDE for Java Developers - Windows (78 MB)


Eclipse Downloads - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://www.eclipse.org/downloads/
Google
Done

The essential tools for any Java developer, including a Java IDE, a CVS client, XML Editor and Mylyn. Find out more... Mac OS X

Eclipse IDE for Java EE Developers - Windows (126 MB)
Tools for Java developers creating JEE and Web applications, including a Java IDE, tools for JEE and JSF, Mylyn and others. **Java 5 (or higher) required**. Find out more...
Windows Linux Mac OS X

Eclipse IDE for C/C++ Developers - Windows (63 MB)
An IDE for C/C++ developers. Find out more...
Windows Linux Mac OS X

Eclipse for RCP/Plug-in Developers - Windows (153 MB)
A complete set of tools for developers who want to create Eclipse plug-ins or Rich Client Applications. It includes a complete SDK, developer tools and source code. Find out more...
Windows Linux Mac OS X

Eclipse Classic 3.3.1.1 - Windows (140 MB)
The classic Eclipse download: the Eclipse Platform, Java Development Tools, and Plug-in Development Environment, including source and both user and programmer documentation. Find out more...
Release notes | All Versions | Documentation
Windows Linux Mac OS X

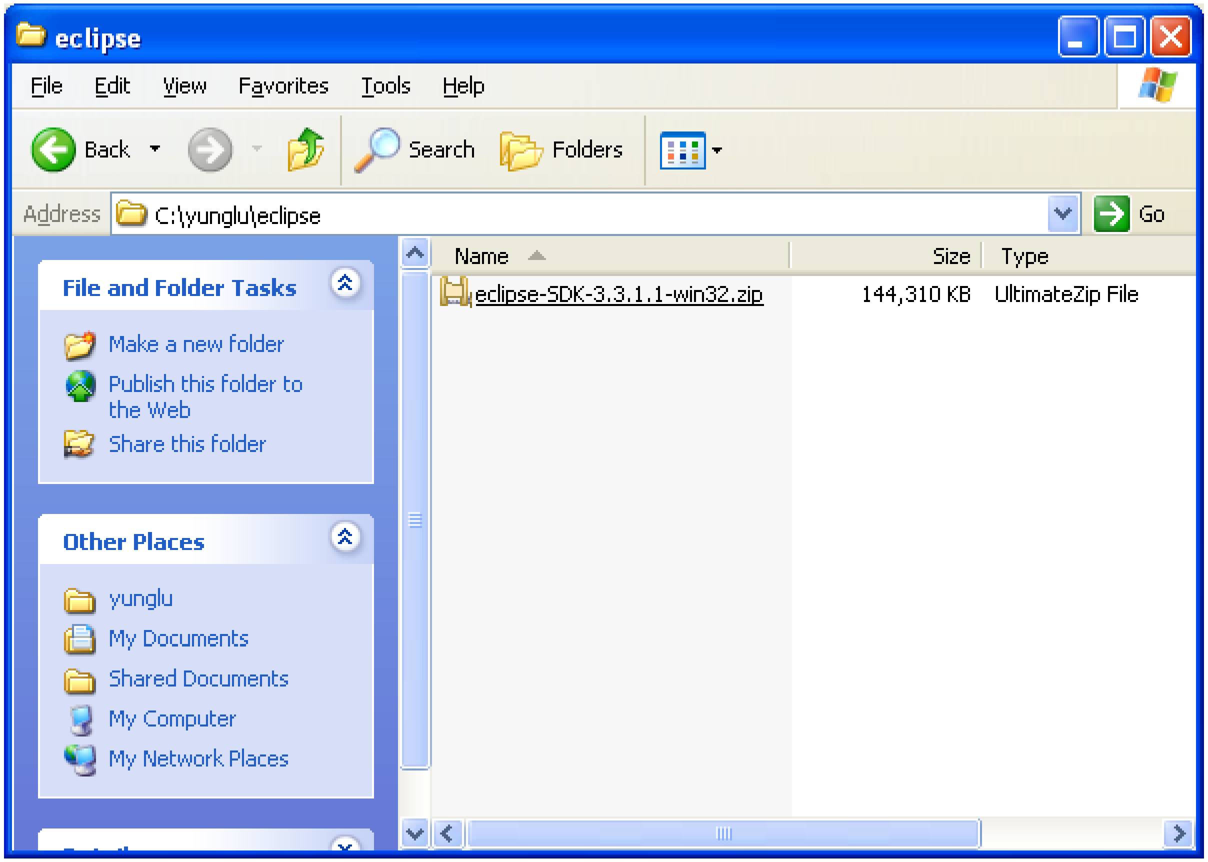
Developmer (PDT)
3. Modeling Tools (MDT)
4. C/C++ Developmer (CDT)
5. Visual Editor (VE)
6. Business Intelligence and Reporting (BIRT)
7. Modeling Framework (EMF)
8. Mylyn
9. Test & Performanc (TPTP)
10. AspectJ

Updated: Feb 6/08

Third Party Distros

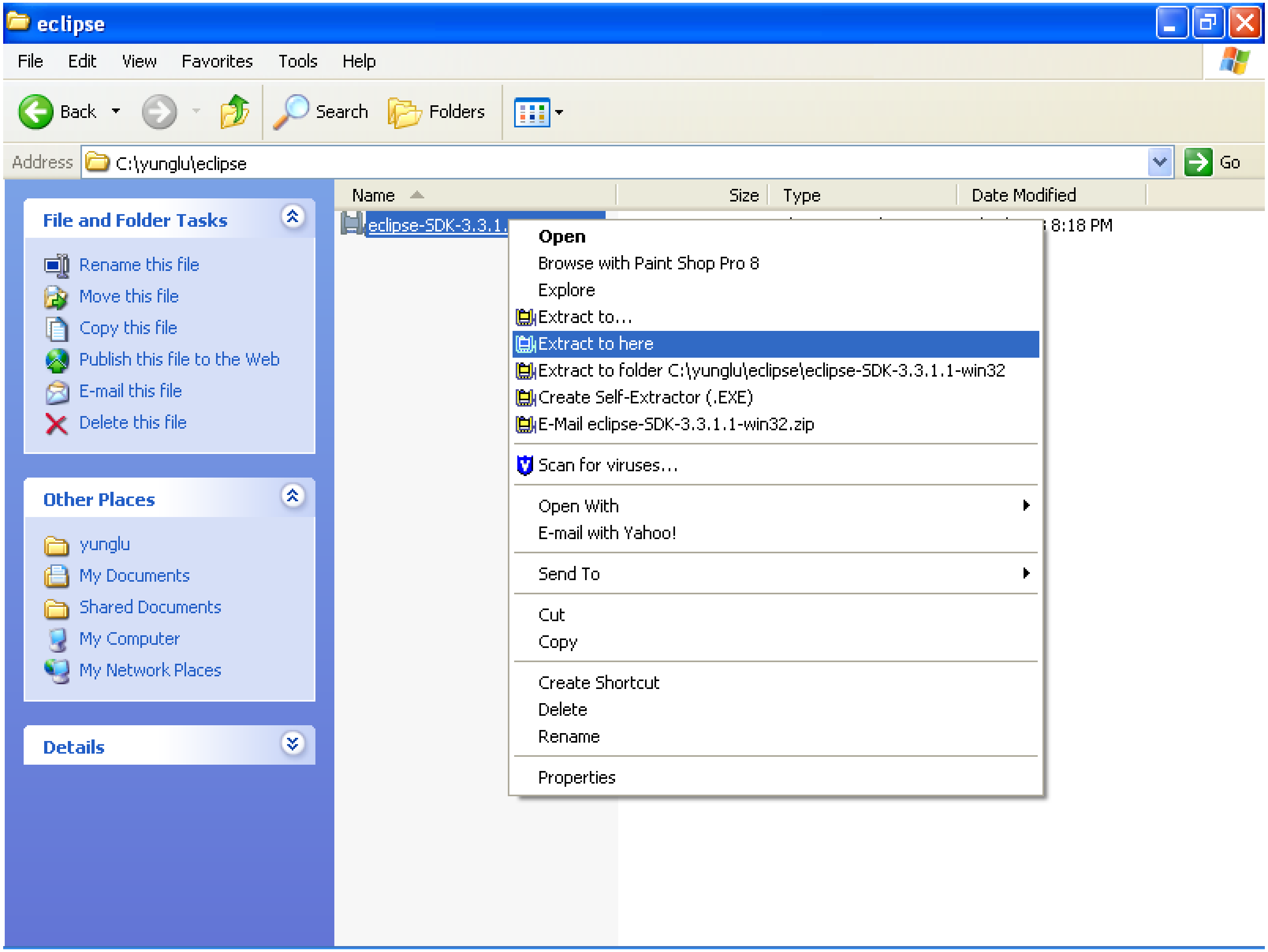
Innoopract Projects: All

How could we improve the

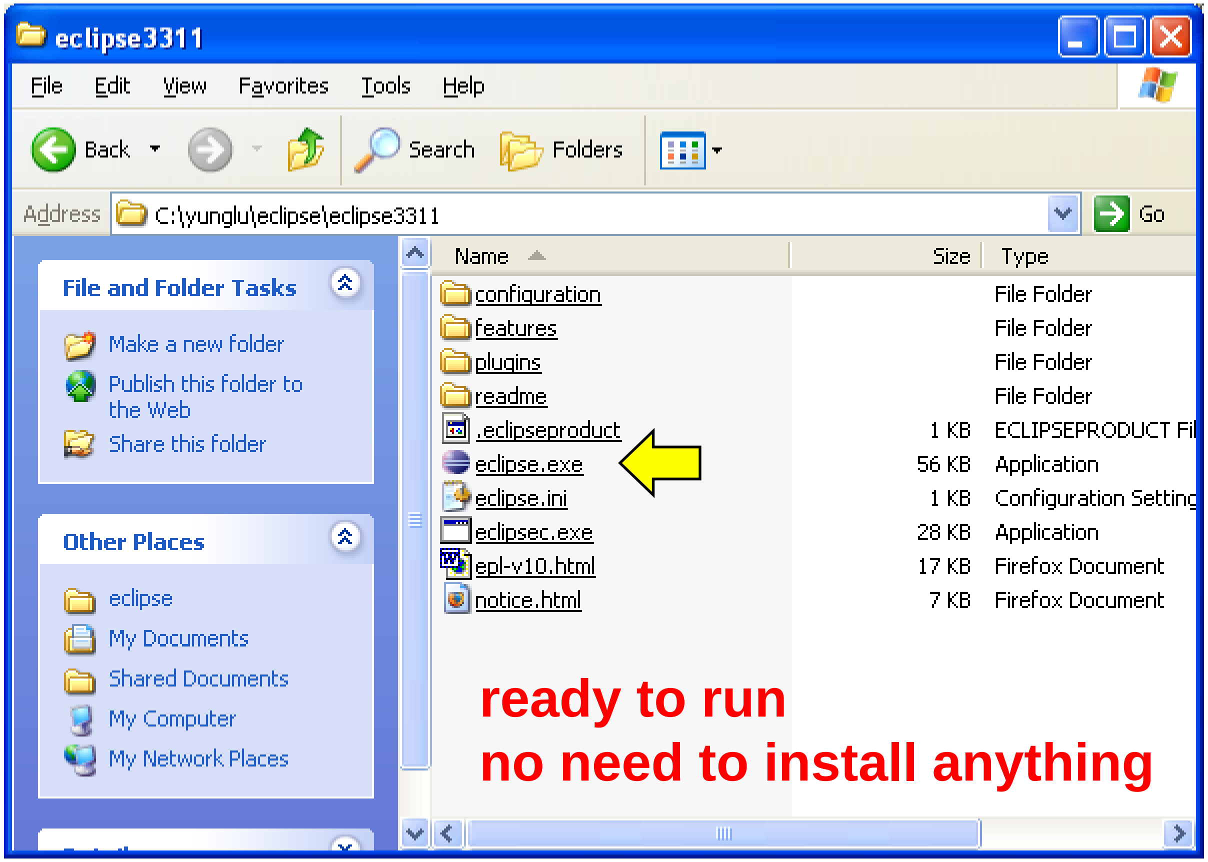


Address C:\yunglu\eclipse

Name	Size	Type
eclipse-SDK-3.3.1.1-win32.zip	144,310 KB	UltimateZip File



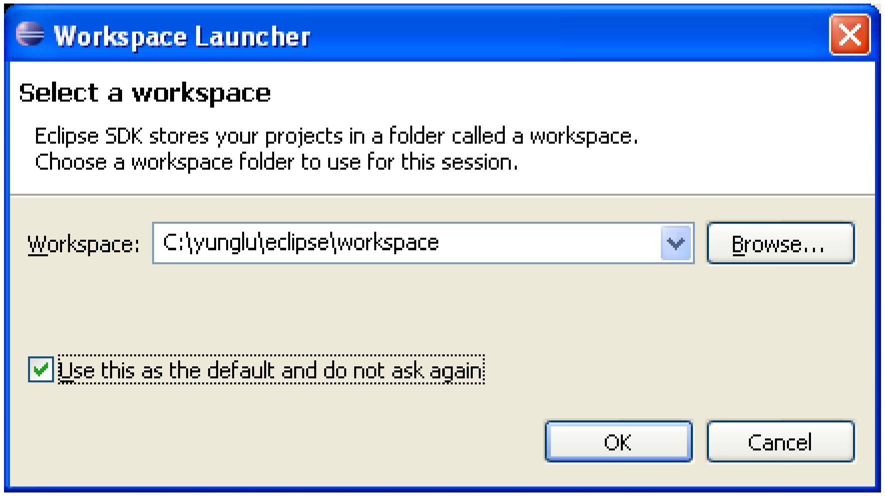
Address C:\yunglu\eclipse



Address: C:\yunglu\eclipse\eclipse3311

Name	Size	Type
configuration		File Folder
features		File Folder
plugins		File Folder
readme		File Folder
.eclipseproduct	1 KB	ECLIPSEPRODUCT Fil
eclipse.exe	56 KB	Application
eclipse.ini	1 KB	Configuration Setting
eclipsec.exe	28 KB	Application
epl-v10.html	17 KB	Firefox Document
notice.html	7 KB	Firefox Document

ready to run
no need to install anything

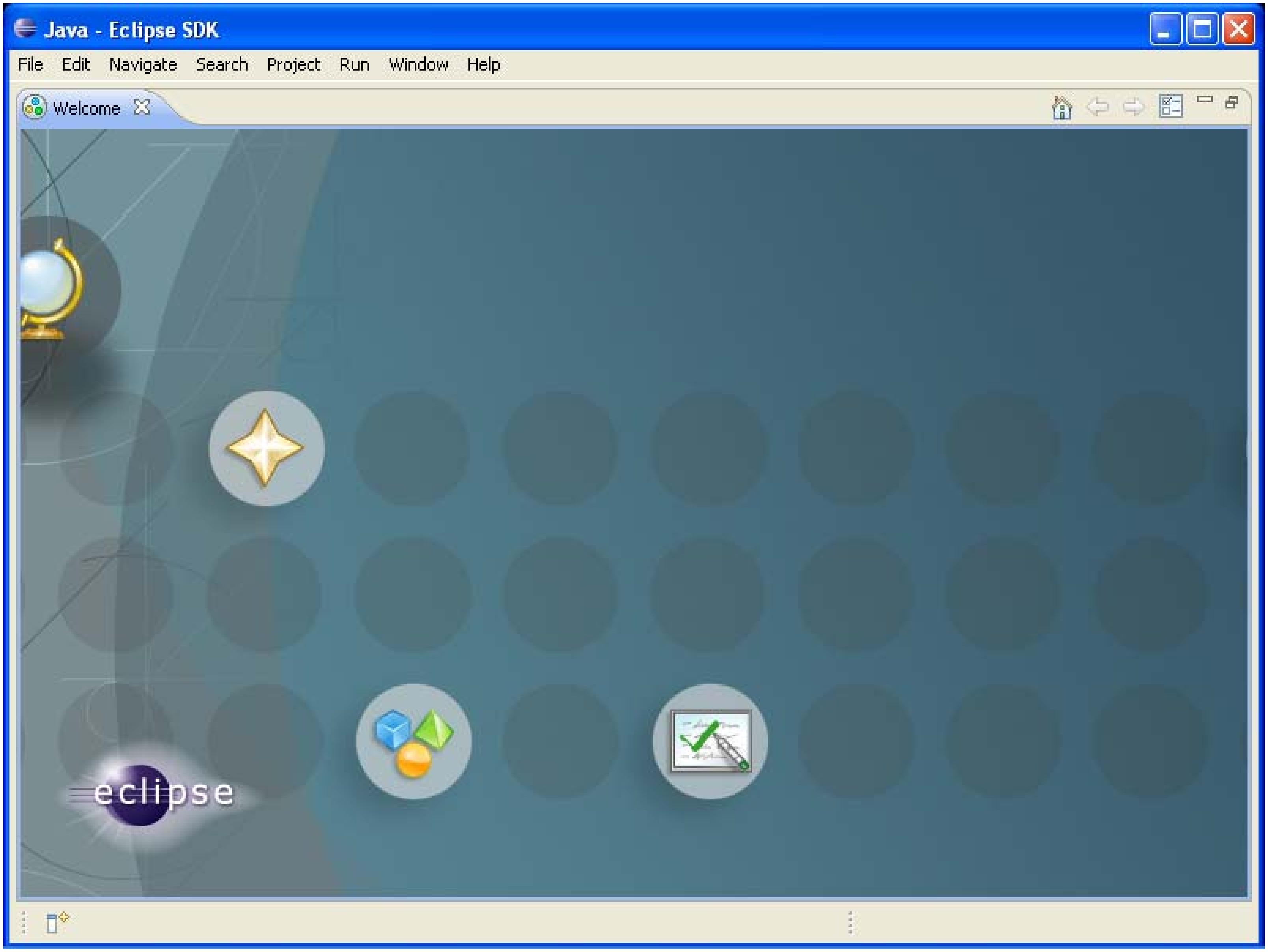


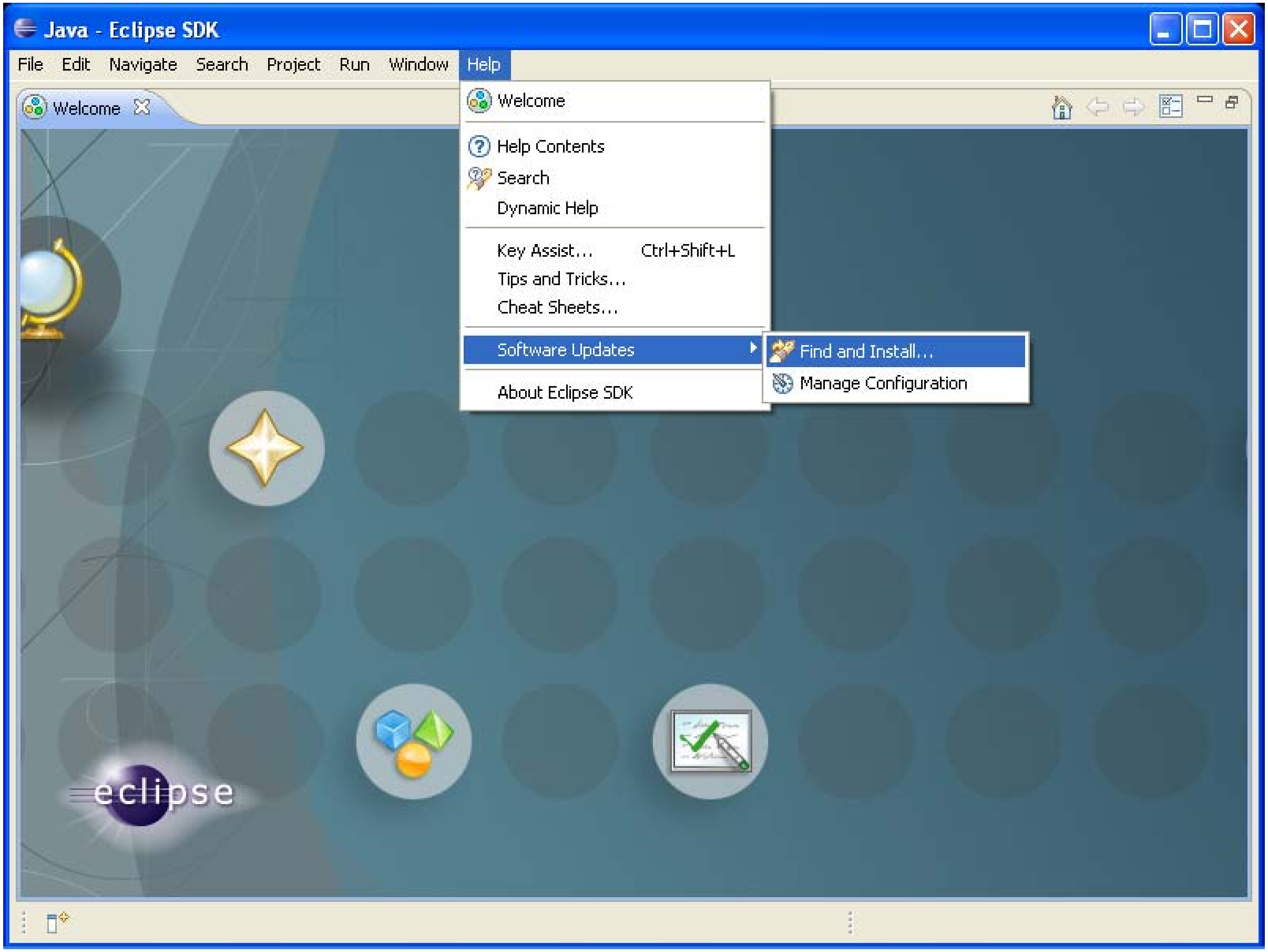
Select a workspace

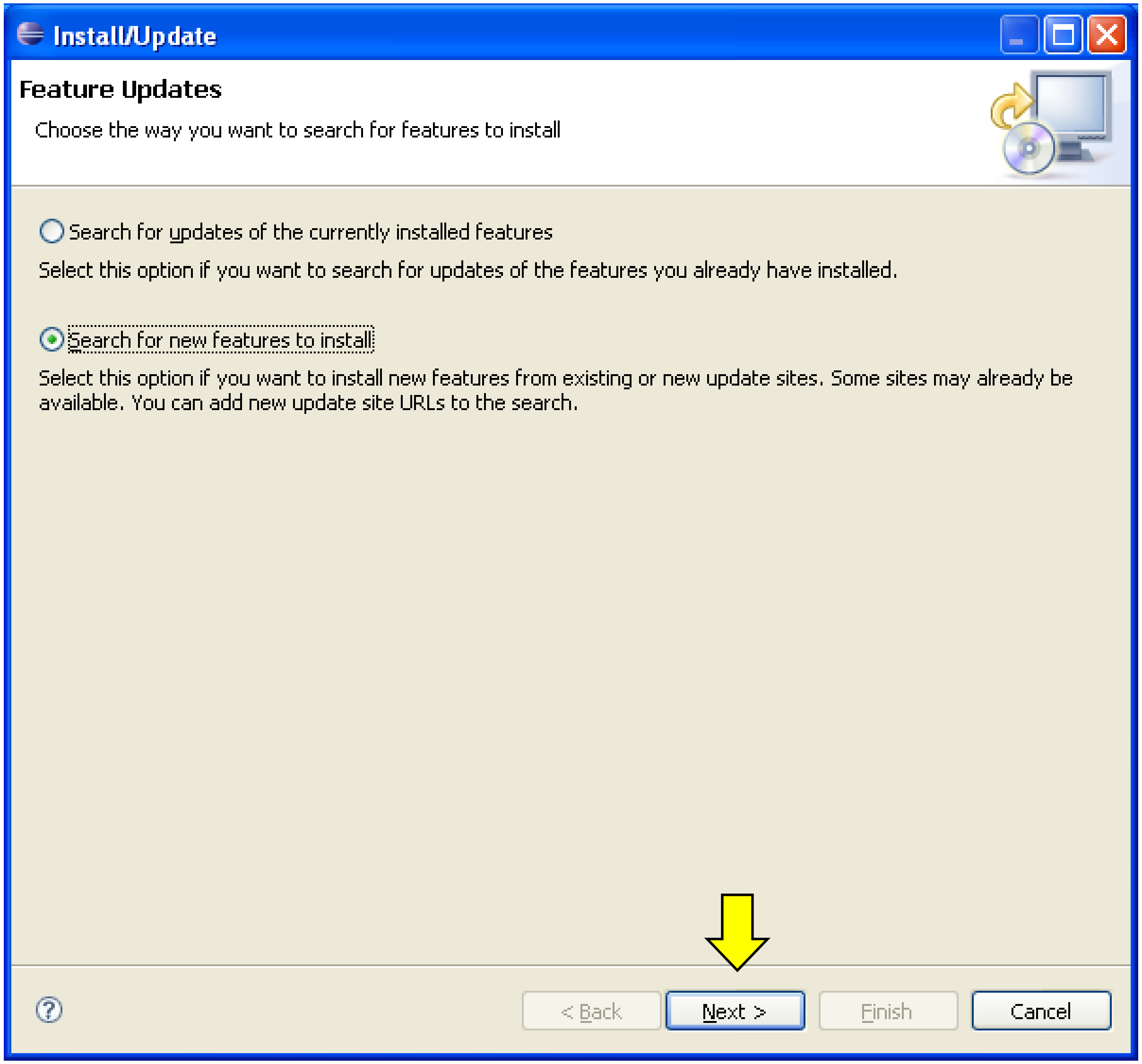
Eclipse SDK stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\yunglu\eclipse\workspace

☑ Use this as the default and do not ask again


Java - Eclipse SDK
File
Edit
Navigate
Search
Project
Run
Window
Help
Welcome
eclipse


Java - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Welcome
Welcome
Help Contents
Search
Dynamic Help
Key Assist... Ctrl+Shift+L
Tips and Tricks...
Cheat Sheets...
Software Updates
About Eclipse SDK
Find and Install...
Manage Configuration
eclipse



Feature Updates

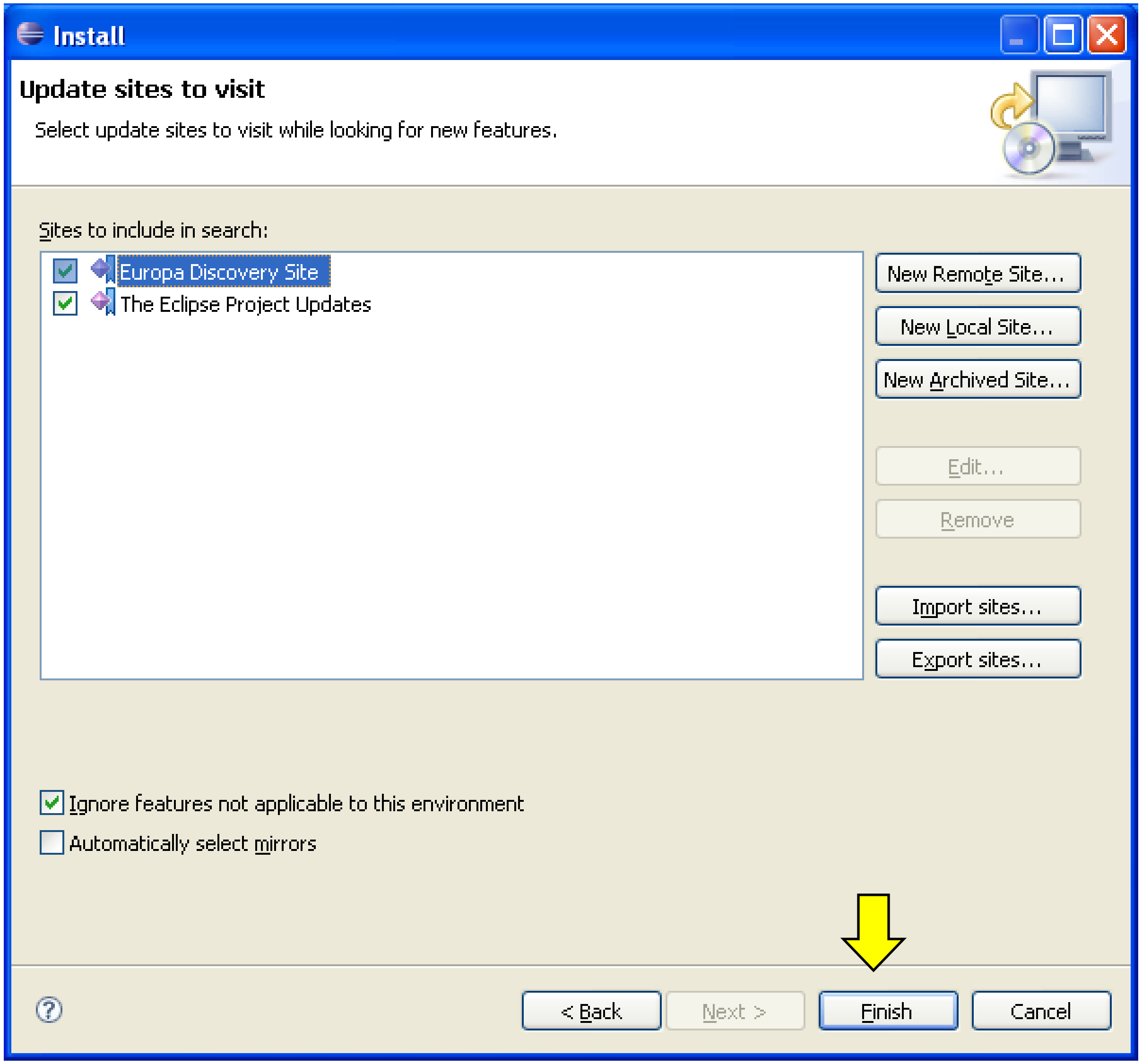
Choose the way you want to search for features to install

○ Search for updates of the currently installed features

Select this option if you want to search for updates of the features you already have installed.

● Search for new features to install

Select this option if you want to install new features from existing or new update sites. Some sites may already be available. You can add new update site URLs to the search.



Install

Update sites to visit

Select update sites to visit while looking for new features.

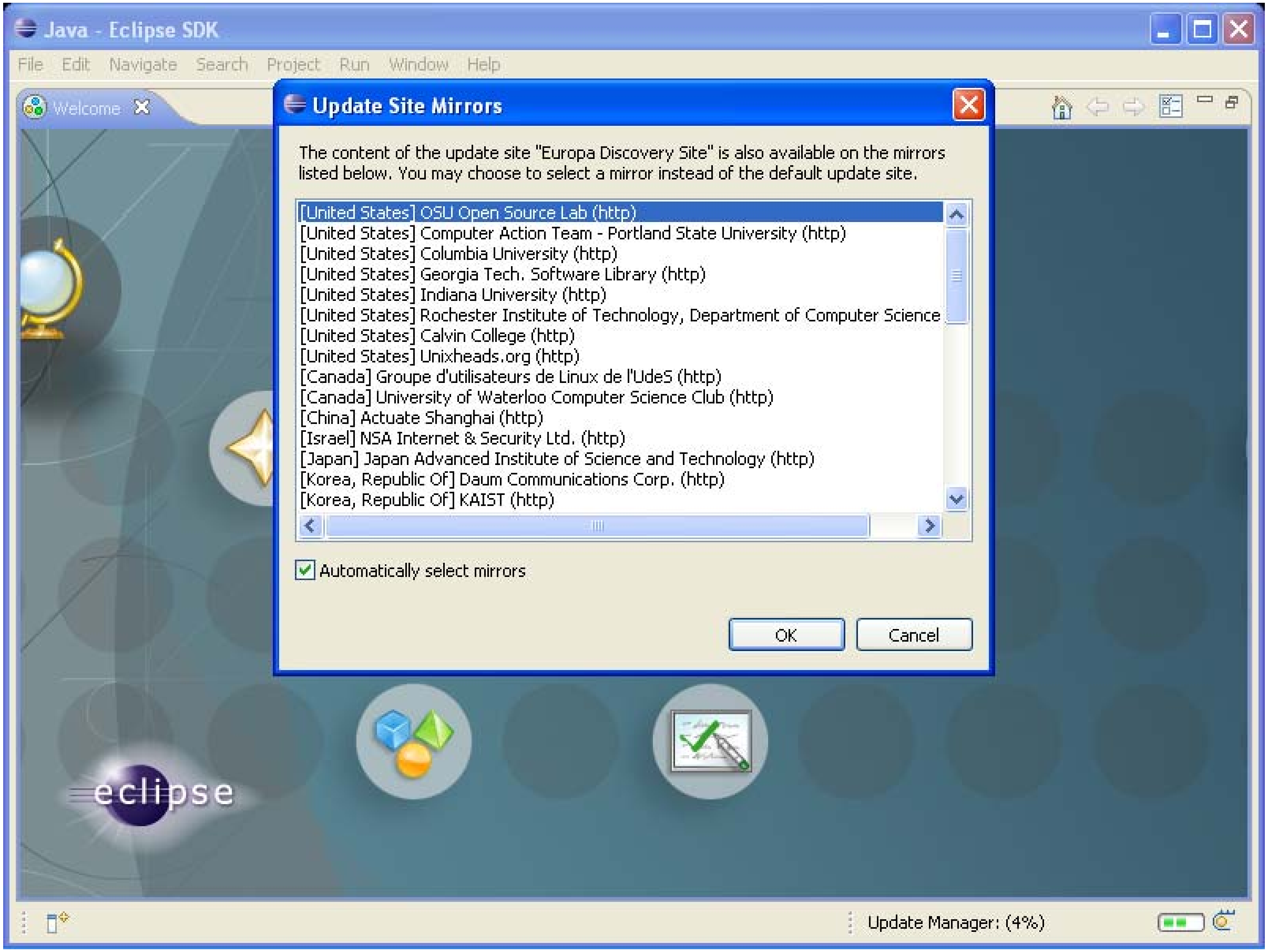
Sites to include in search:

☑ Europa Discovery Site

☑ The Eclipse Project Updates

☑ Ignore features not applicable to this environment

□ Automatically select mirrors

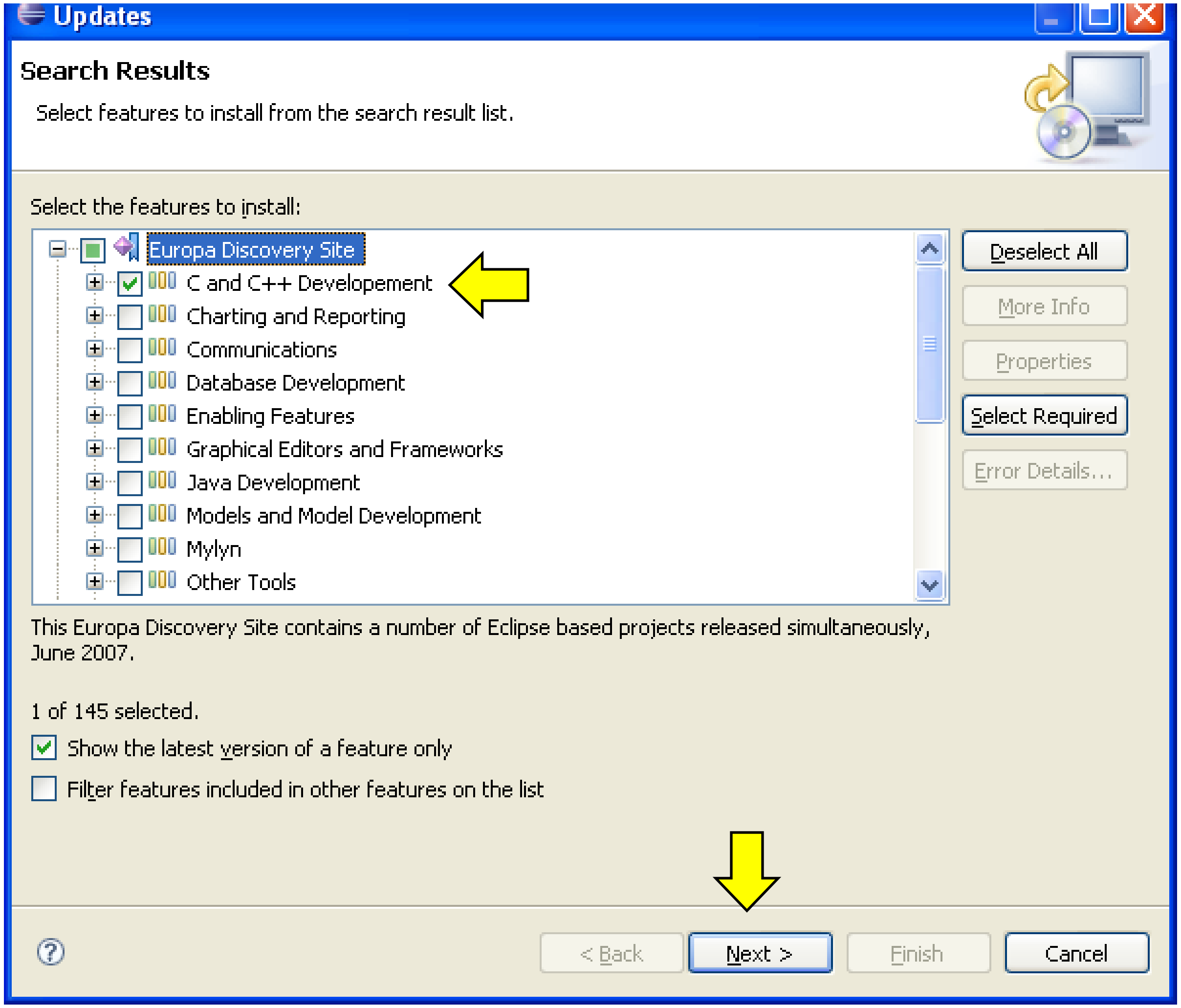


The content of the update site "Europa Discovery Site" is also available on the mirrors listed below. You may choose to select a mirror instead of the default update site.

[United States] OSU Open Source Lab (http)
[United States] Computer Action Team - Portland State University (http)
[United States] Columbia University (http)
[United States] Georgia Tech. Software Library (http)
[United States] Indiana University (http)
[United States] Rochester Institute of Technology, Department of Computer Science
[United States] Calvin College (http)
[United States] Unixheads.org (http)
[Canada] Groupe d'utilisateurs de Linux de l'UdeS (http)
[Canada] University of Waterloo Computer Science Club (http)
[China] Actuate Shanghai (http)
[Israel] NSA Internet & Security Ltd. (http)
[Japan] Japan Advanced Institute of Science and Technology (http)
[Korea, Republic Of] Daum Communications Corp. (http)
[Korea, Republic Of] KAIST (http)

☑ Automatically select mirrors

Update Manager: (4%)



Updates

Search Results

Select features to install from the search result list.

Select the features to install:

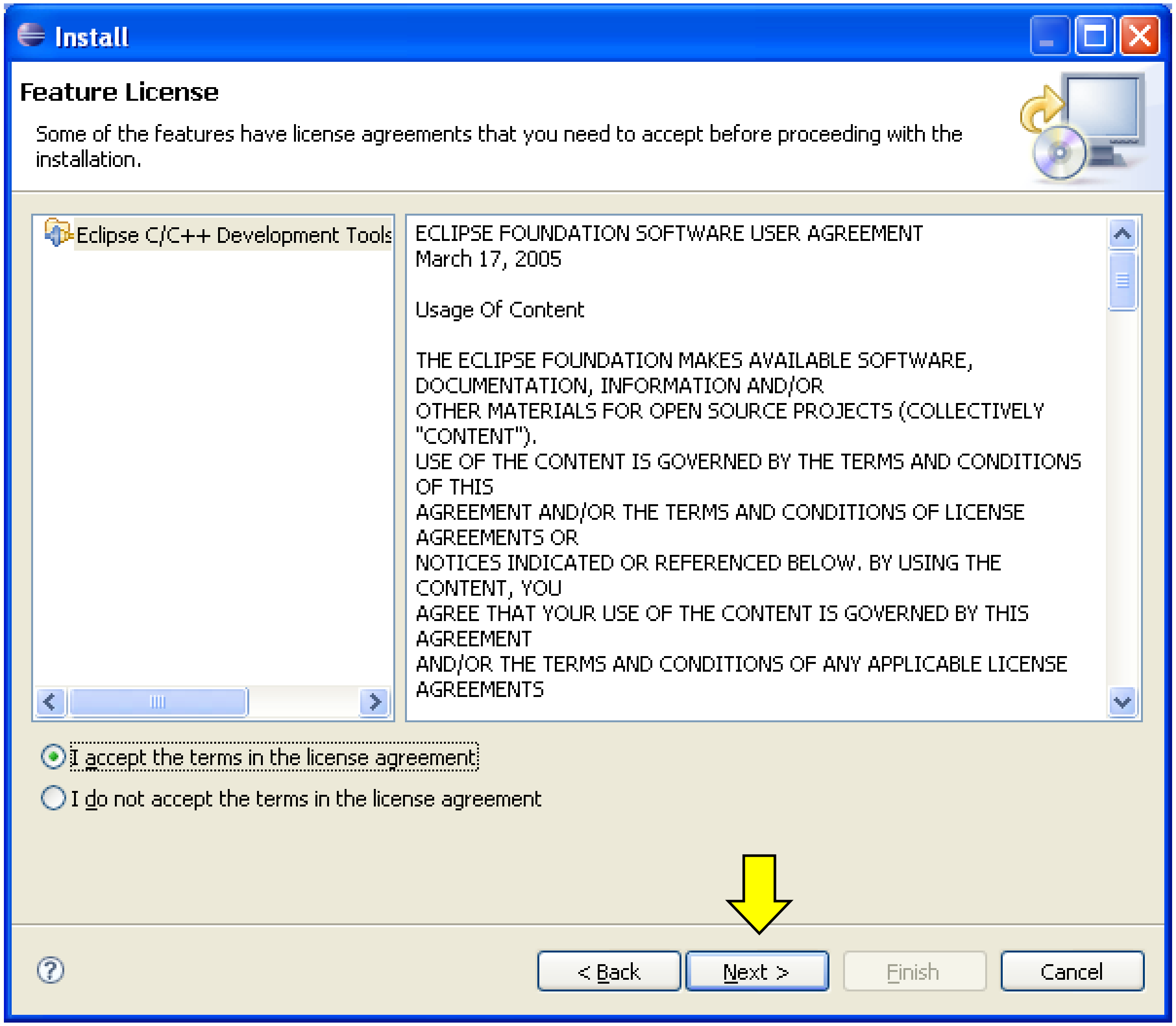
- ■ Europa Discovery Site
  - ☑ C and C++ Development
  - □ Charting and Reporting
  - □ Communications
  - □ Database Development
  - □ Enabling Features
  - □ Graphical Editors and Frameworks
  - □ Java Development
  - □ Models and Model Development
  - □ Mylyn
  - □ Other Tools

This Europa Discovery Site contains a number of Eclipse based projects released simultaneously, June 2007.

1 of 145 selected.

☑ Show the latest version of a feature only

□ Filter features included in other features on the list



Feature License

Some of the features have license agreements that you need to accept before proceeding with the installation.

Eclipse C/C++ Development Tools

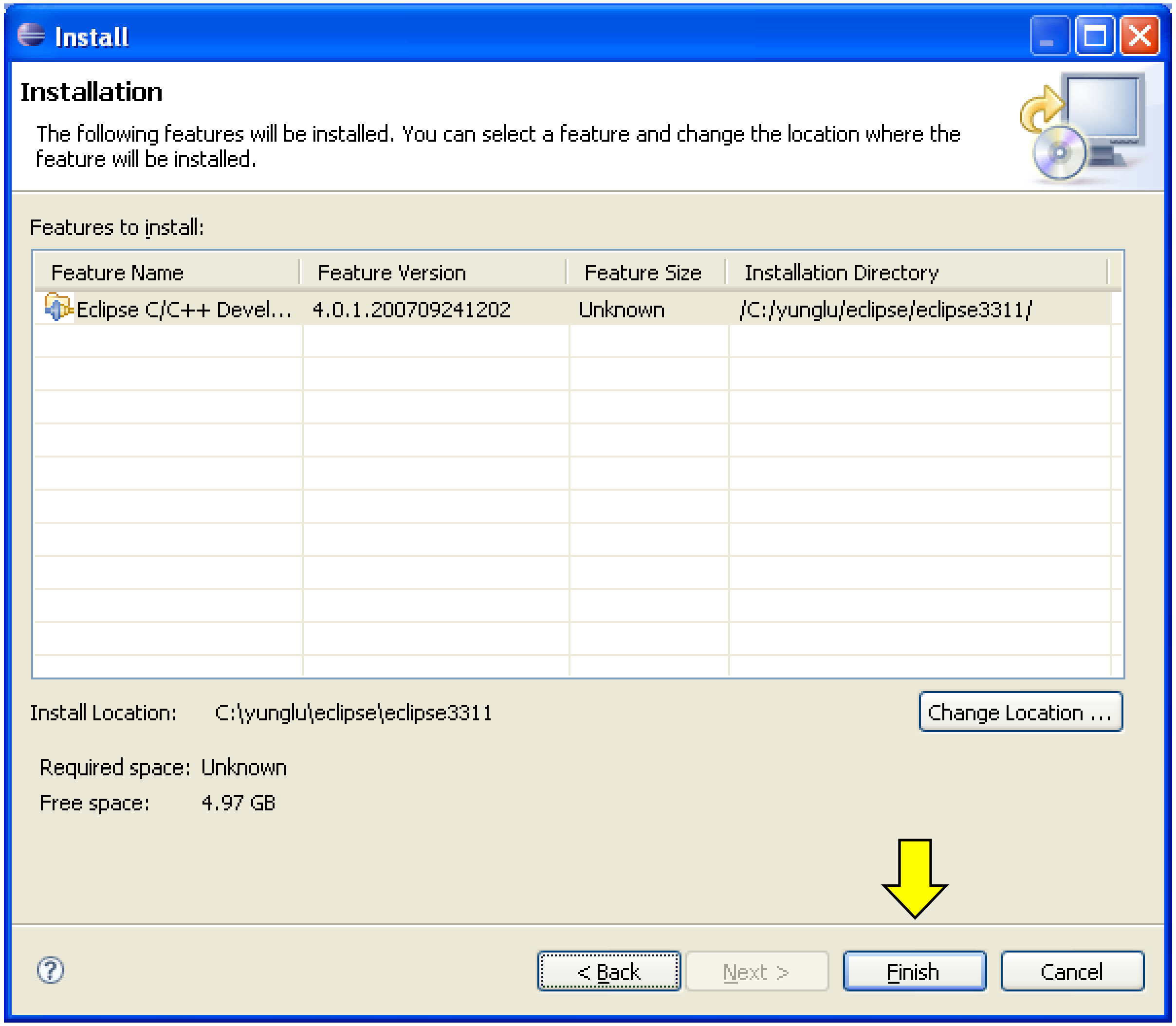
ECLIPSE FOUNDATION SOFTWARE USER AGREEMENT
March 17, 2005

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement



Installation

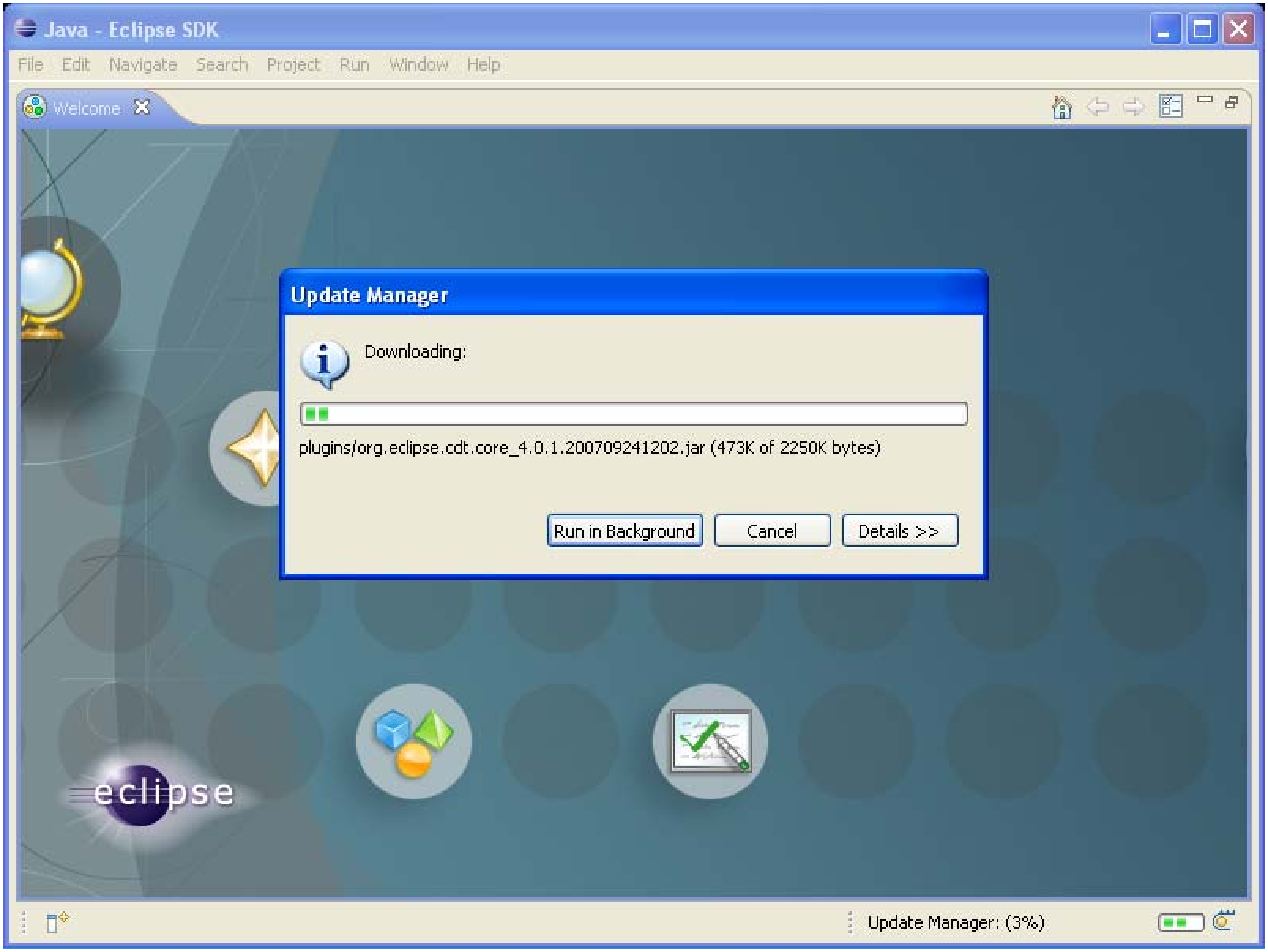
The following features will be installed. You can select a feature and change the location where the feature will be installed.

Features to install:

Feature Name	Feature Version	Feature Size	Installation Directory
Eclipse C/C++ Devel...	4.0.1.200709241202	Unknown	/C:/yunglu/eclipse/eclipse3311/

Install Location: C:\yunglu\eclipse\eclipse3311

Required space: Unknown
Free space: 4.97 GB

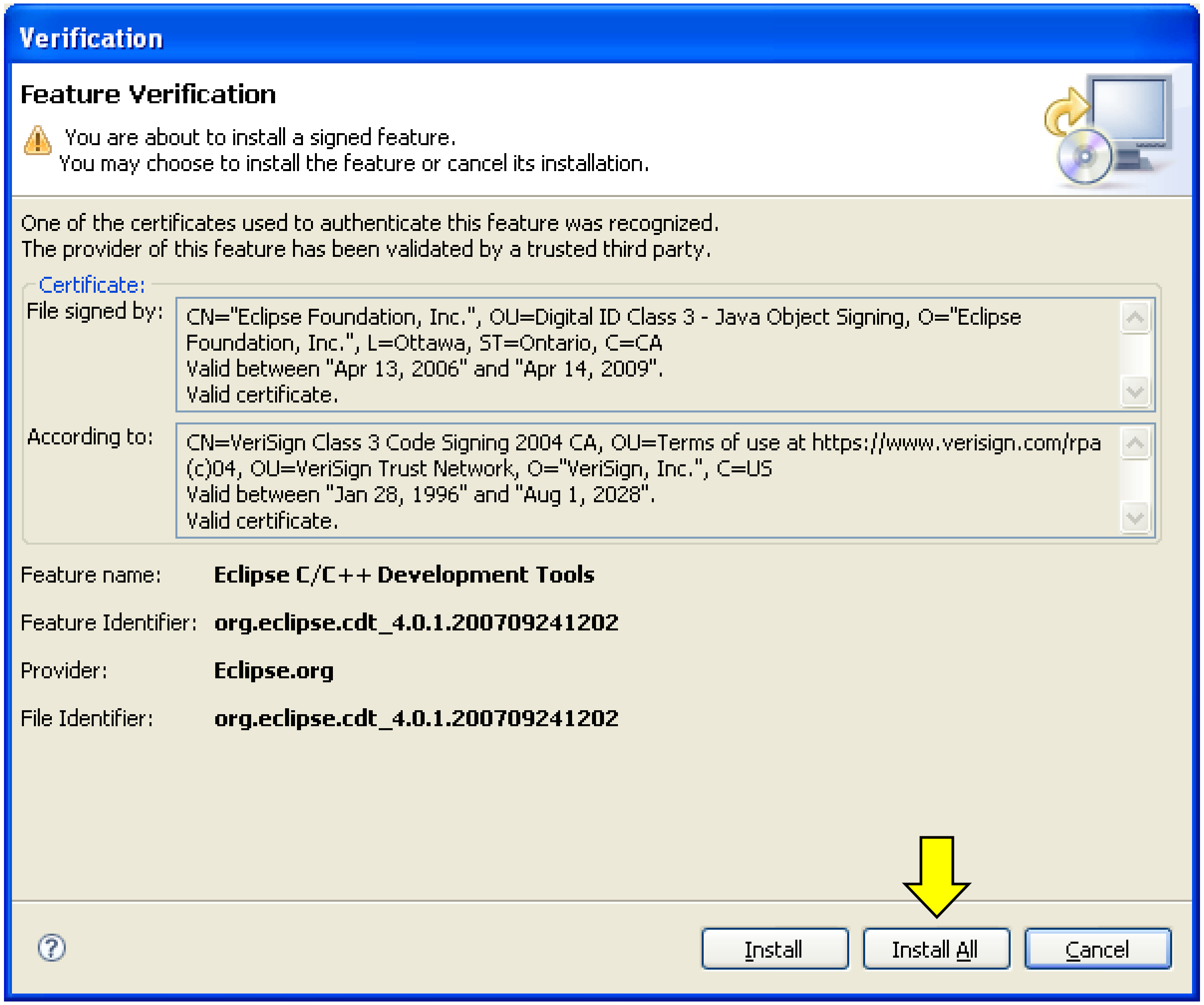


Update Manager

Downloading:

plugins/org.eclipse.cdt.core_4.0.1.200709241202.jar (473K of 2250K bytes)

Update Manager: (3%)



Feature Verification

You are about to install a signed feature.
You may choose to install the feature or cancel its installation.

One of the certificates used to authenticate this feature was recognized.
The provider of this feature has been validated by a trusted third party.

Certificate:

File signed by: CN="Eclipse Foundation, Inc.", OU=Digital ID Class 3 - Java Object Signing, O="Eclipse Foundation, Inc.", L=Ottawa, ST=Ontario, C=CA
Valid between "Apr 13, 2006" and "Apr 14, 2009".
Valid certificate.

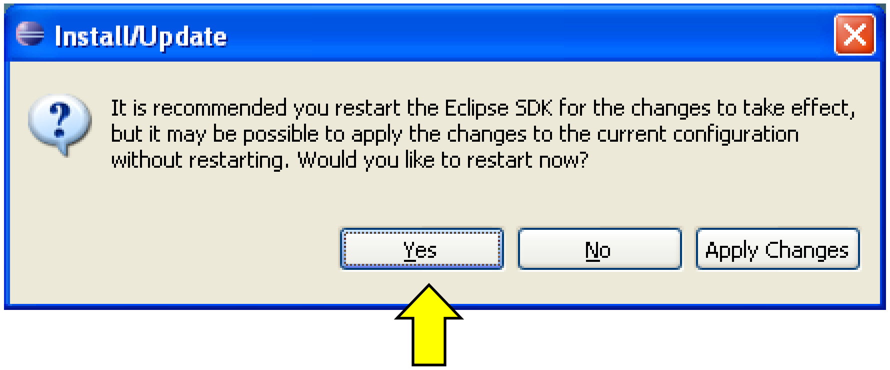
According to: CN=VeriSign Class 3 Code Signing 2004 CA, OU=Terms of use at https://www.verisign.com/rpa (c)04, OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US
Valid between "Jan 28, 1996" and "Aug 1, 2028".
Valid certificate.

Feature name: **Eclipse C/C++ Development Tools**

Feature Identifier: **org.eclipse.cdt_4.0.1.200709241202**

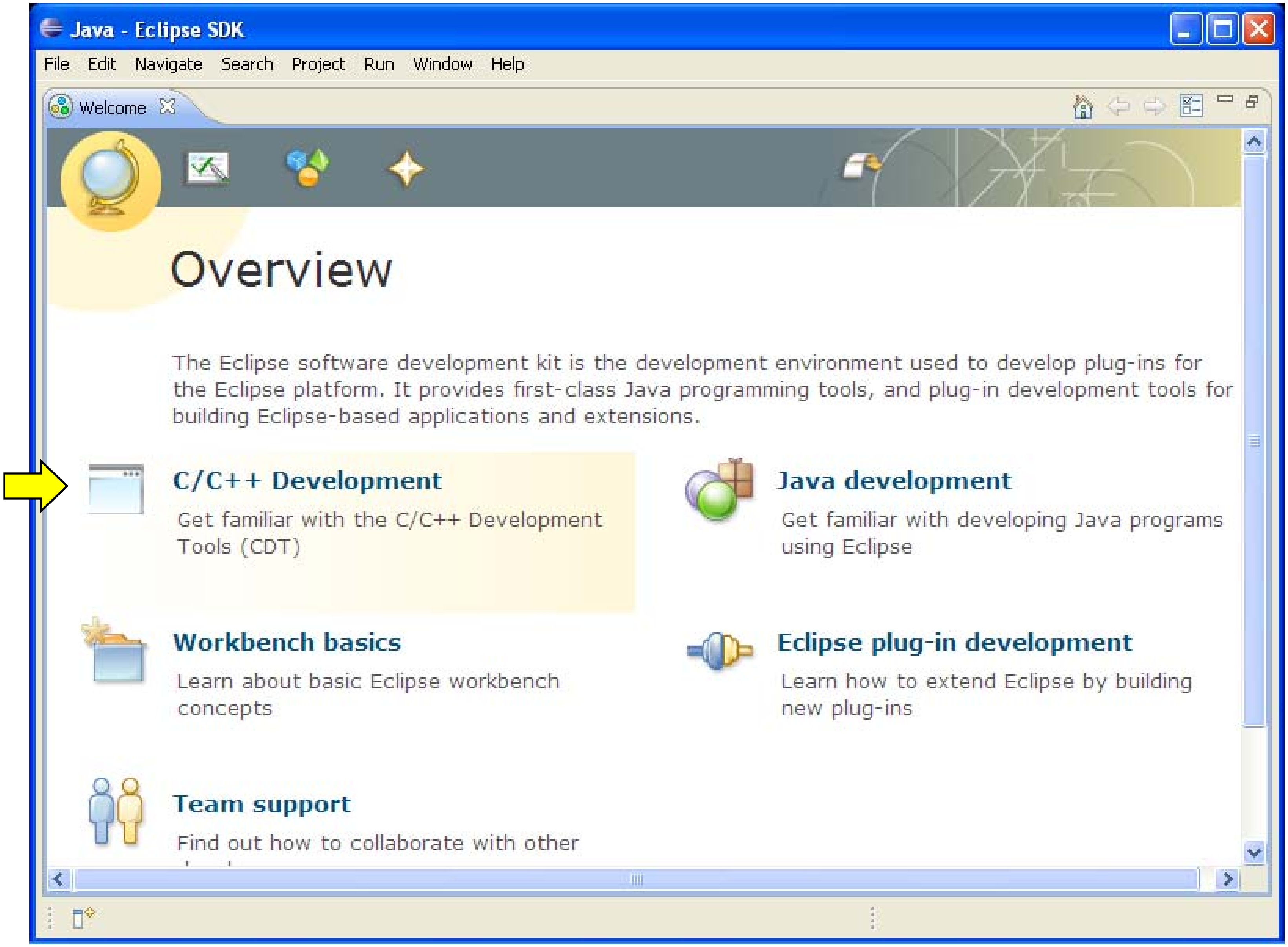
Provider: **Eclipse.org**

File Identifier: **org.eclipse.cdt_4.0.1.200709241202**



Install/Update

It is recommended you restart the Eclipse SDK for the changes to take effect, but it may be possible to apply the changes to the current configuration without restarting. Would you like to restart now?



Overview

The Eclipse software development kit is the development environment used to develop plug-ins for the Eclipse platform. It provides first-class Java programming tools, and plug-in development tools for building Eclipse-based applications and extensions.

C/C++ Development

Get familiar with the C/C++ Development Tools (CDT)

Java development

Get familiar with developing Java programs using Eclipse

Workbench basics

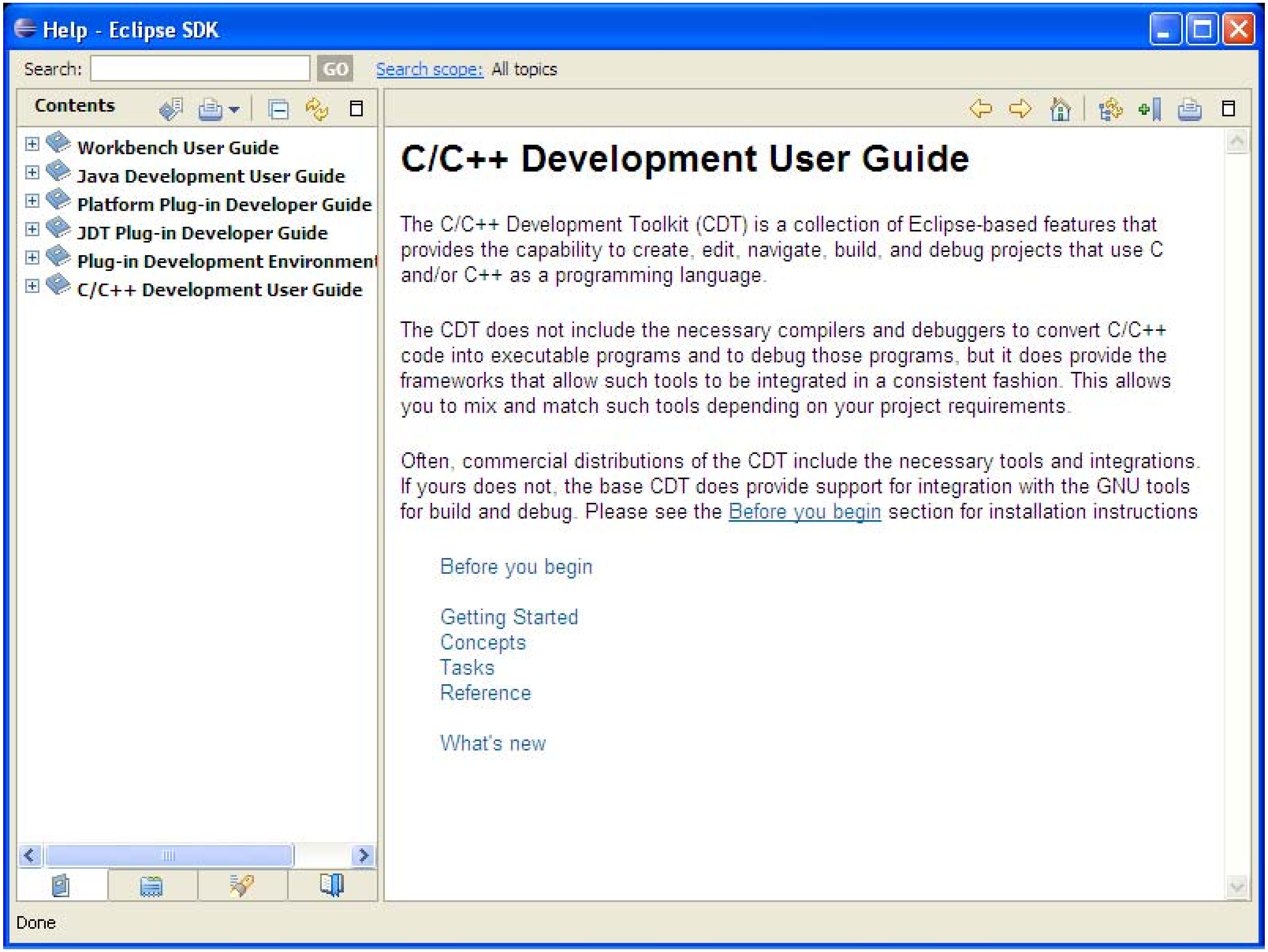
Learn about basic Eclipse workbench concepts

Eclipse plug-in development

Learn how to extend Eclipse by building new plug-ins

Team support

Find out how to collaborate with other


Help - Eclipse SDK
Search:
GO
Search scope: All topics
Contents
Workbench User Guide
Java Development User Guide
Platform Plug-in Developer Guide
JDT Plug-in Developer Guide
Plug-in Development Environment
C/C++ Development User Guide
Done

C/C++ Development User Guide

The C/C++ Development Toolkit (CDT) is a collection of Eclipse-based features that provides the capability to create, edit, navigate, build, and debug projects that use C and/or C++ as a programming language.

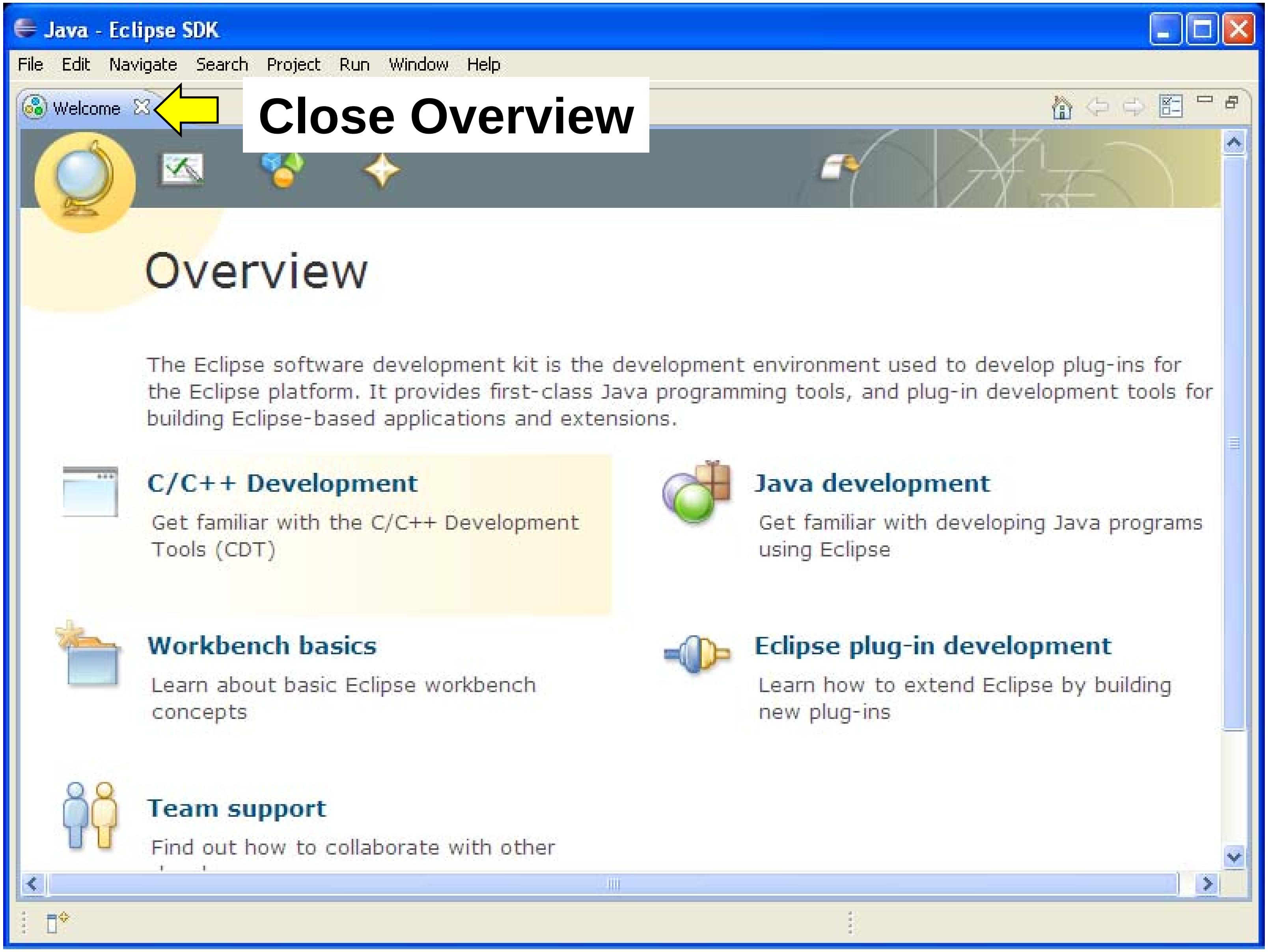
The CDT does not include the necessary compilers and debuggers to convert C/C++ code into executable programs and to debug those programs, but it does provide the frameworks that allow such tools to be integrated in a consistent fashion. This allows you to mix and match such tools depending on your project requirements.

Often, commercial distributions of the CDT include the necessary tools and integrations. If yours does not, the base CDT does provide support for integration with the GNU tools for build and debug. Please see the Before you begin section for installation instructions

Before you begin

Getting Started
Concepts
Tasks
Reference

What's new



Close Overview

Overview

The Eclipse software development kit is the development environment used to develop plug-ins for the Eclipse platform. It provides first-class Java programming tools, and plug-in development tools for building Eclipse-based applications and extensions.

C/C++ Development

Get familiar with the C/C++ Development Tools (CDT)

Java development

Get familiar with developing Java programs using Eclipse

Workbench basics

Learn about basic Eclipse workbench concepts

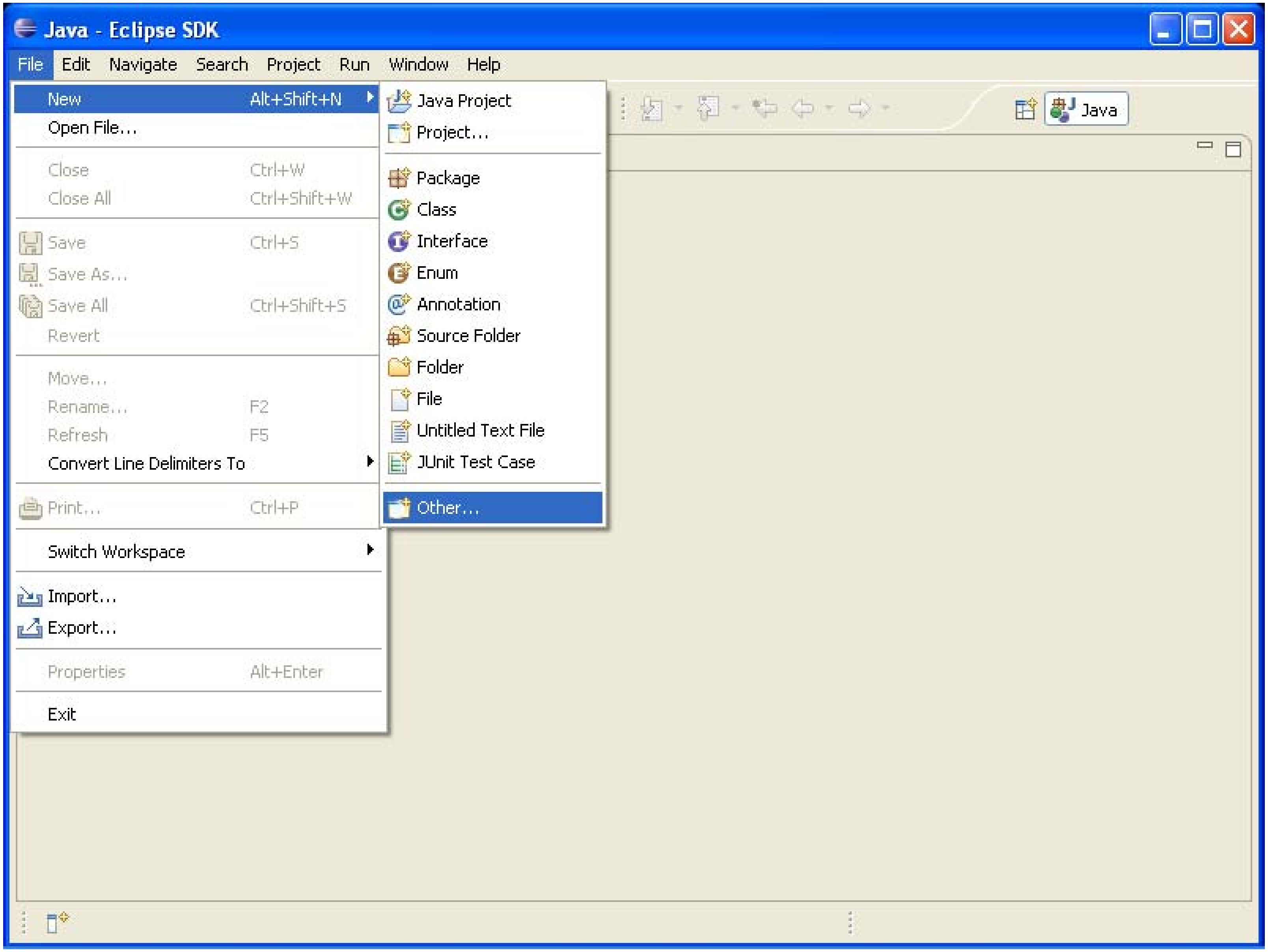
Eclipse plug-in development

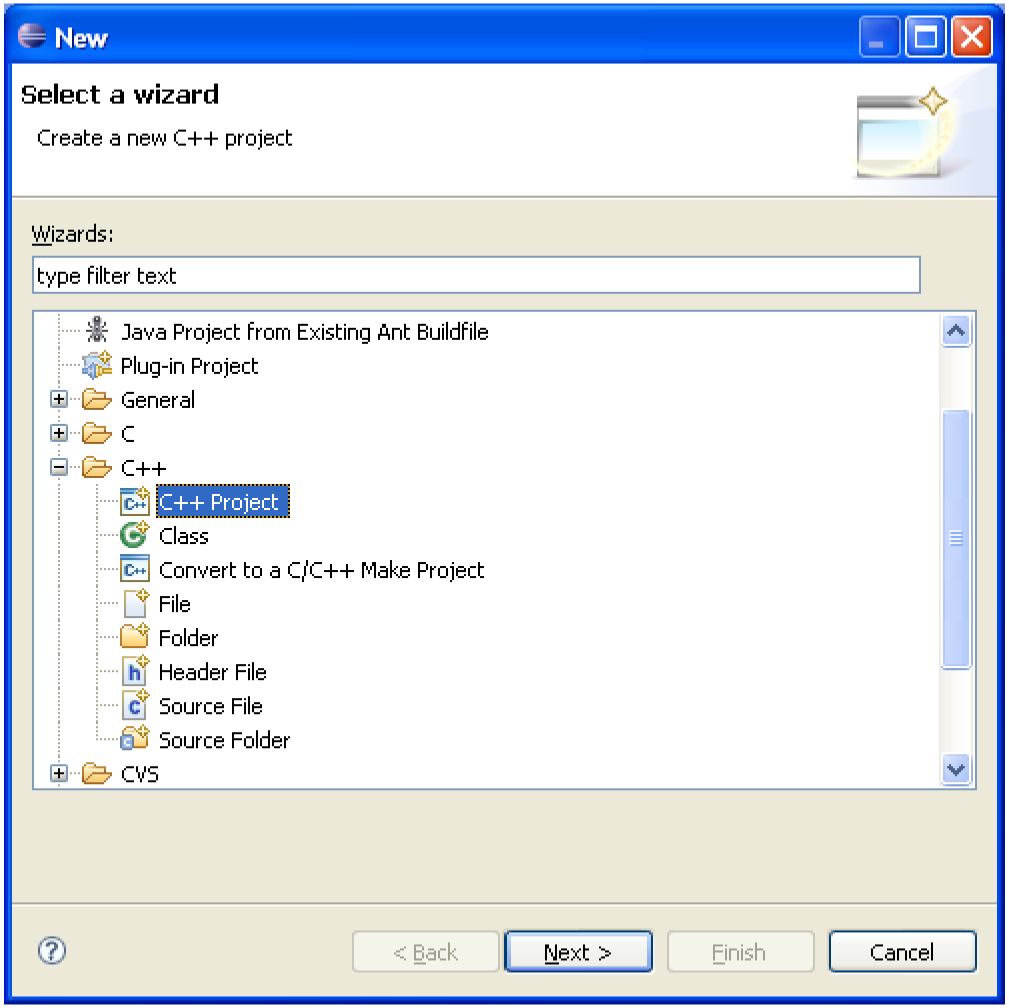
Learn how to extend Eclipse by building new plug-ins

Team support

Find out how to collaborate with other

Develop C++ Projects in Eclipse


Java - Eclipse SDK
File
Edit
Navigate
Search
Project
Run
Window
Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Import...
Export...
Properties Alt+Enter
Exit
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Folder
File
Untitled Text File
JUnit Test Case
Other...
Java



New

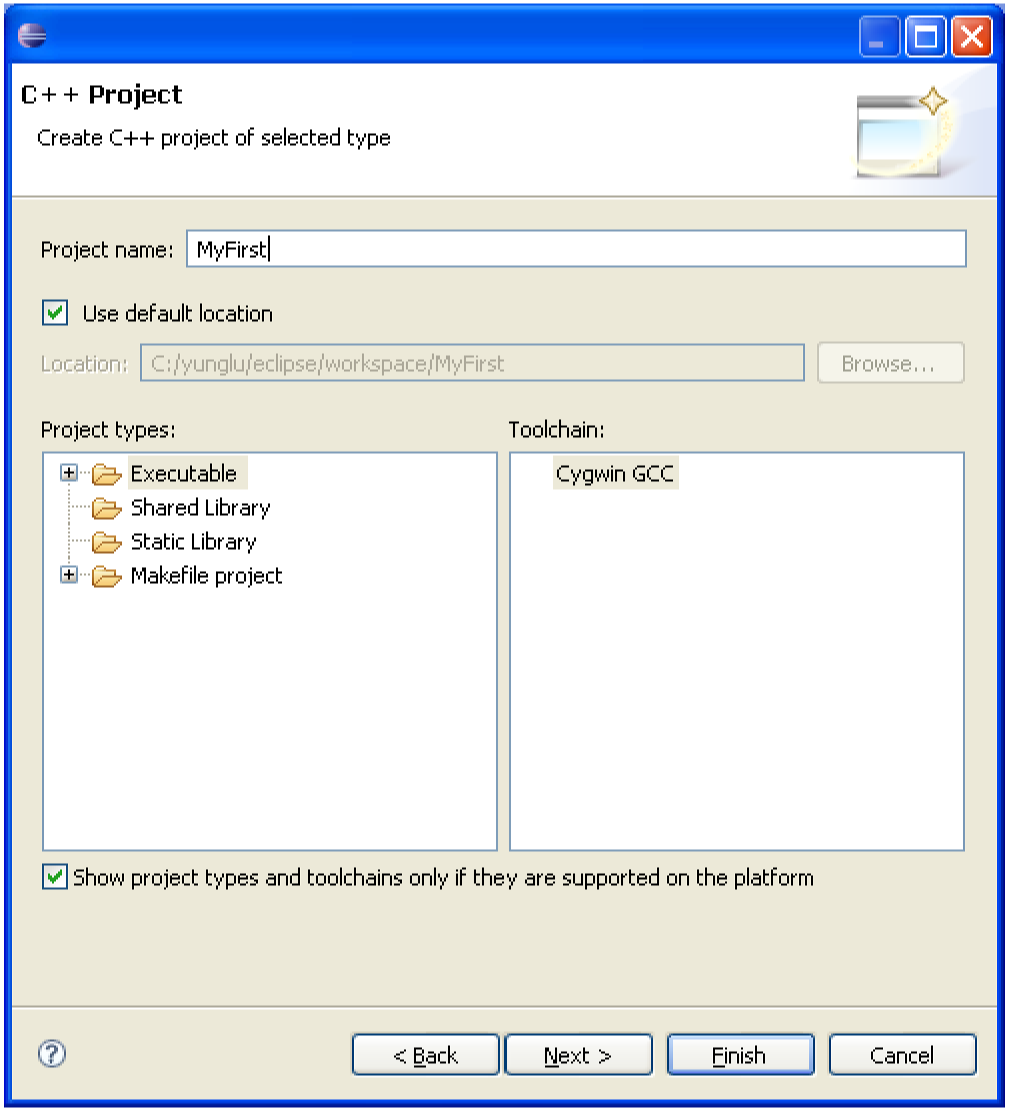
Select a wizard

Create a new C++ project

Wizards:

type filter text

- Java Project from Existing Ant Buildfile
- Plug-in Project
- General
- C
- C++
  - C++ Project
  - Class
  - Convert to a C/C++ Make Project
  - File
  - Folder
  - Header File
  - Source File
  - Source Folder
- CVS



C++ Project

Create C++ project of selected type

Project name: MyFirst

☑ Use default location

Location: C:/yunglu/eclipse/workspace/MyFirst

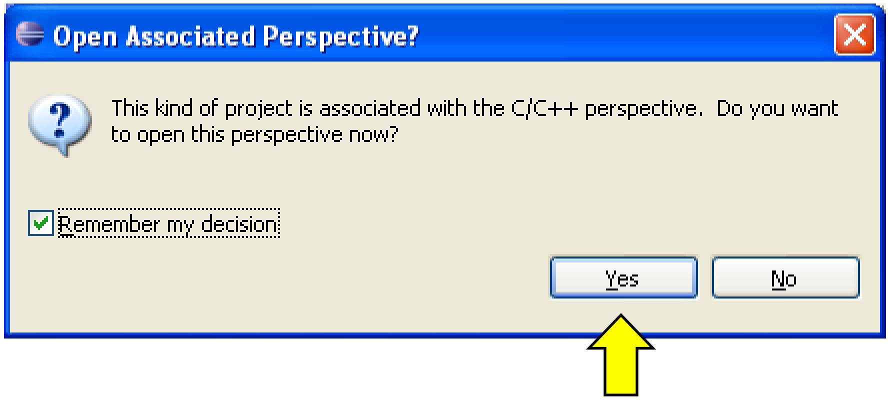
Project types:

- Executable
- Shared Library
- Static Library
- Makefile project

Toolchain:

- Cygwin GCC

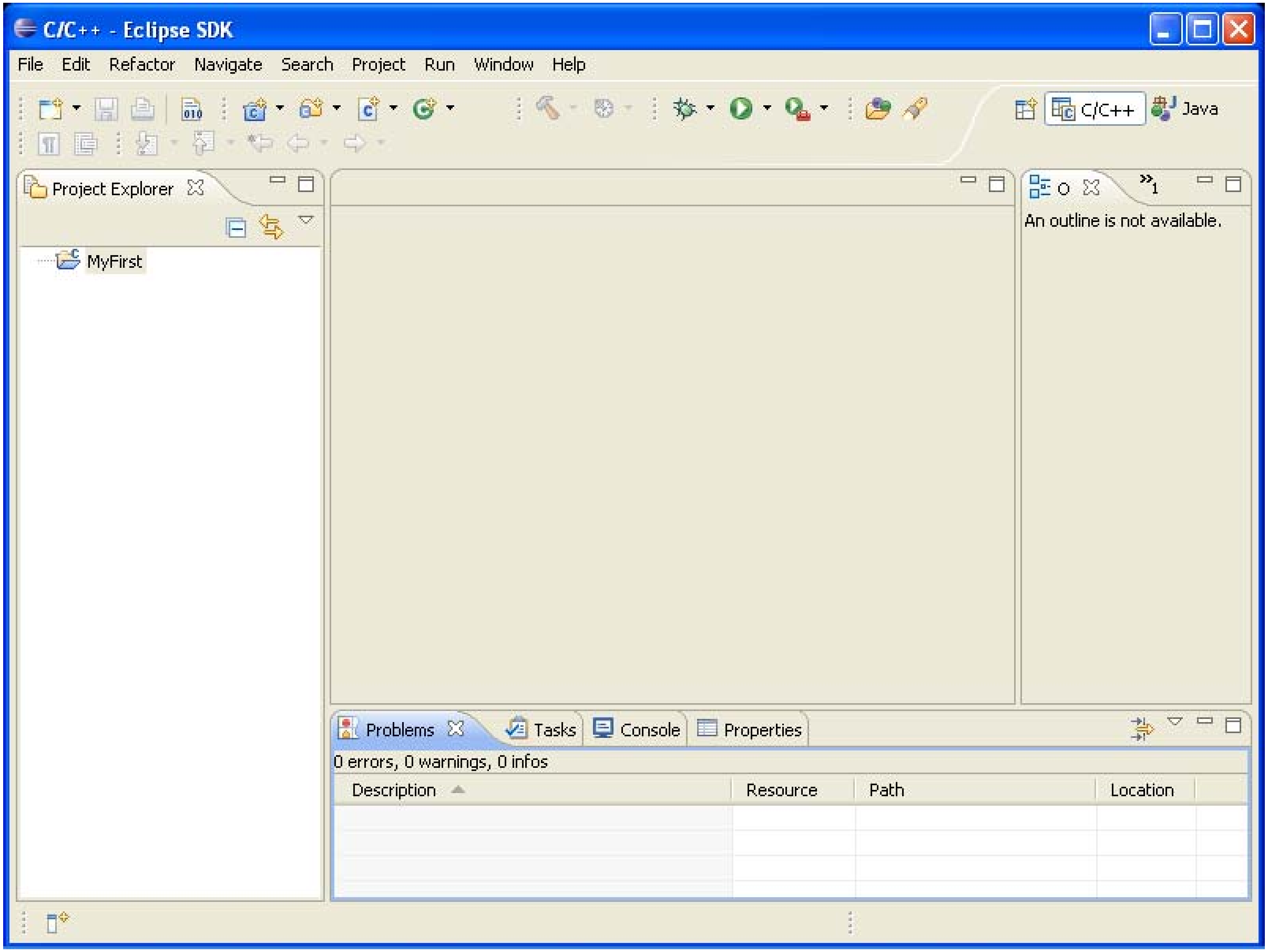
☑ Show project types and toolchains only if they are supported on the platform



Open Associated Perspective?

This kind of project is associated with the C/C++ perspective. Do you want to open this perspective now?

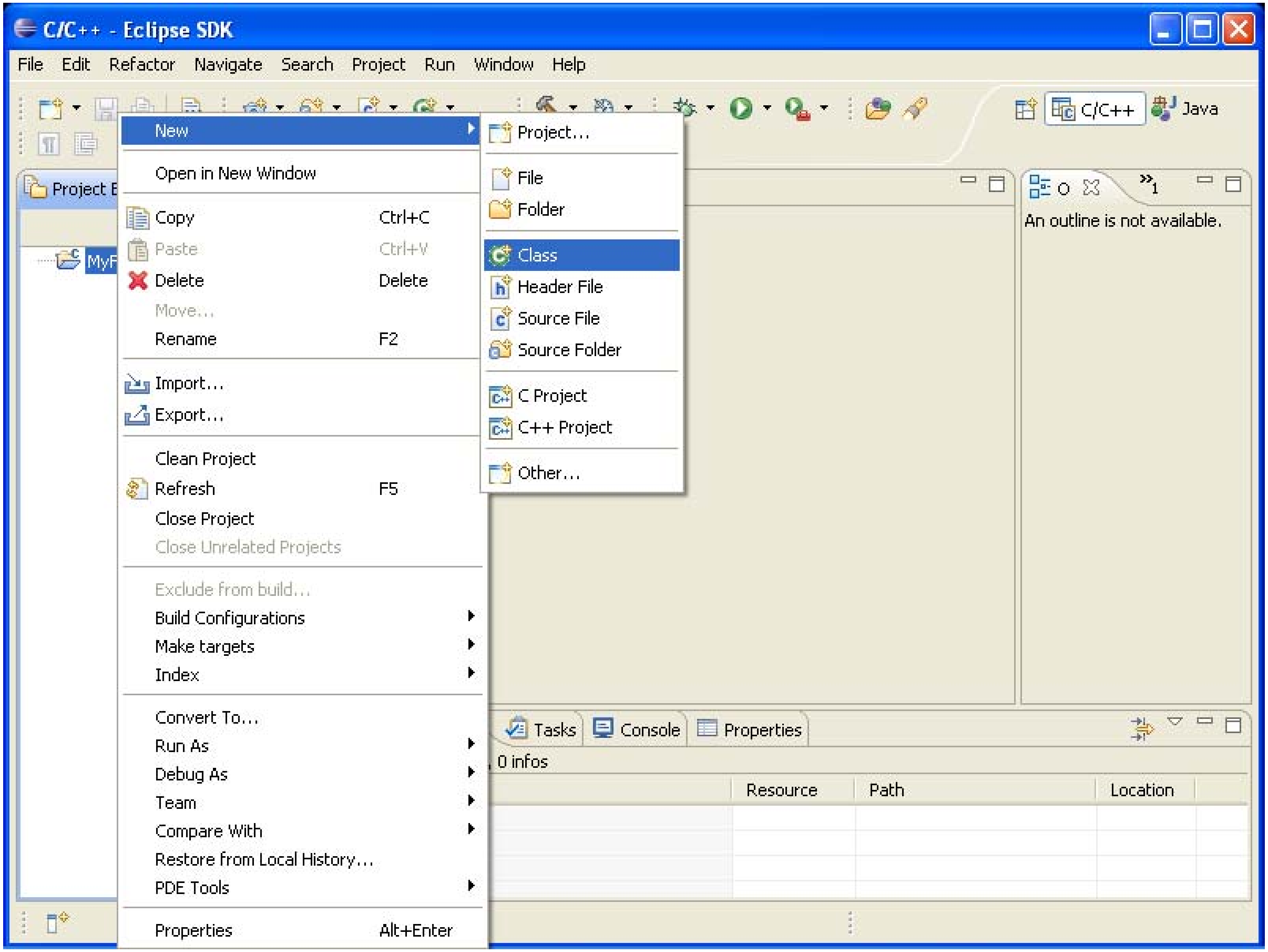
☑ Remember my decision



C/C++ - Eclipse SDK
File Edit Refactor Navigate Search Project Run Window Help
C/C++
Java
Project Explorer
MyFirst
O
An outline is not available.
Problems
Tasks
Console
Properties

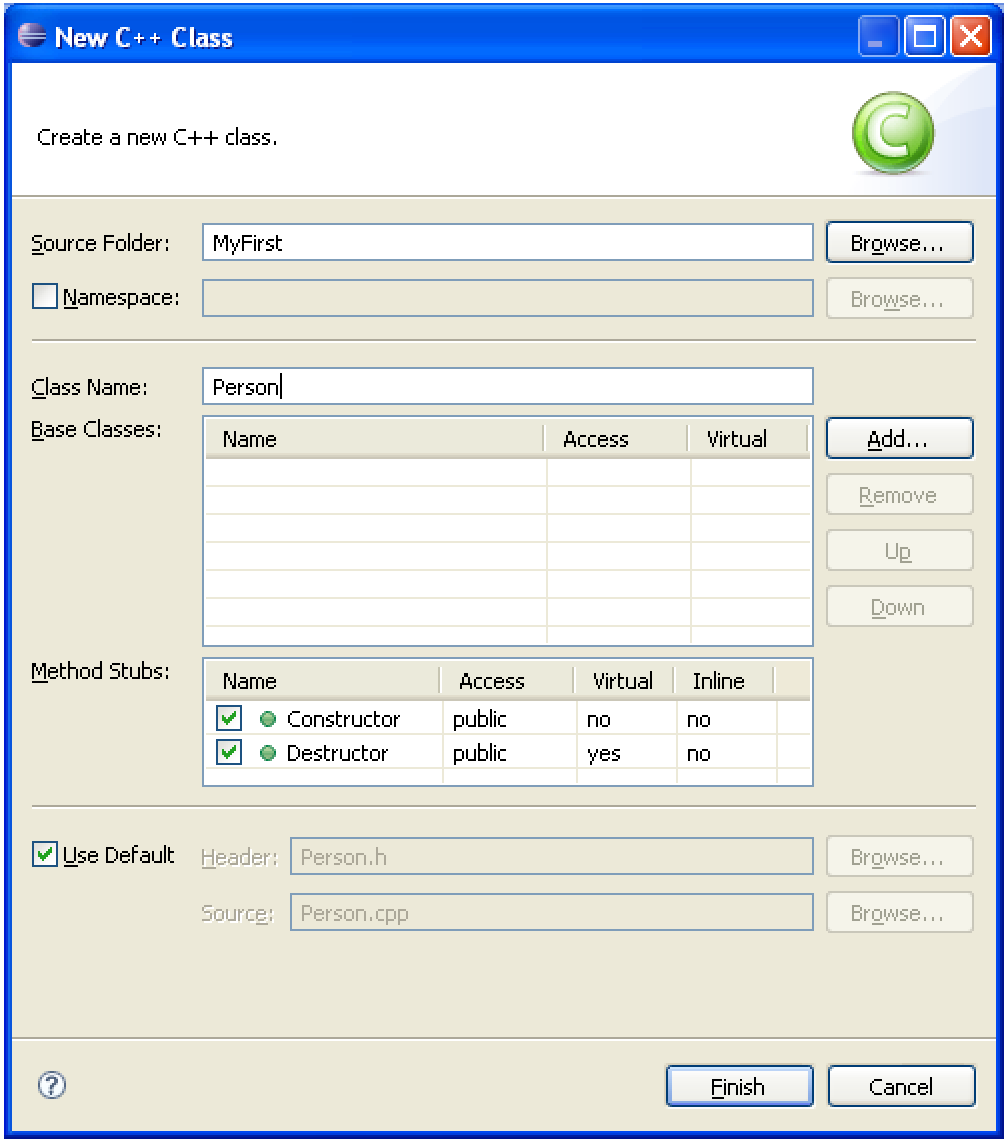
0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location



An outline is not available.

0 infos



Create a new C++ class.

Source Folder: MyFirst

□ Namespace:

Class Name: Person

Base Classes:

Name	Access	Virtual

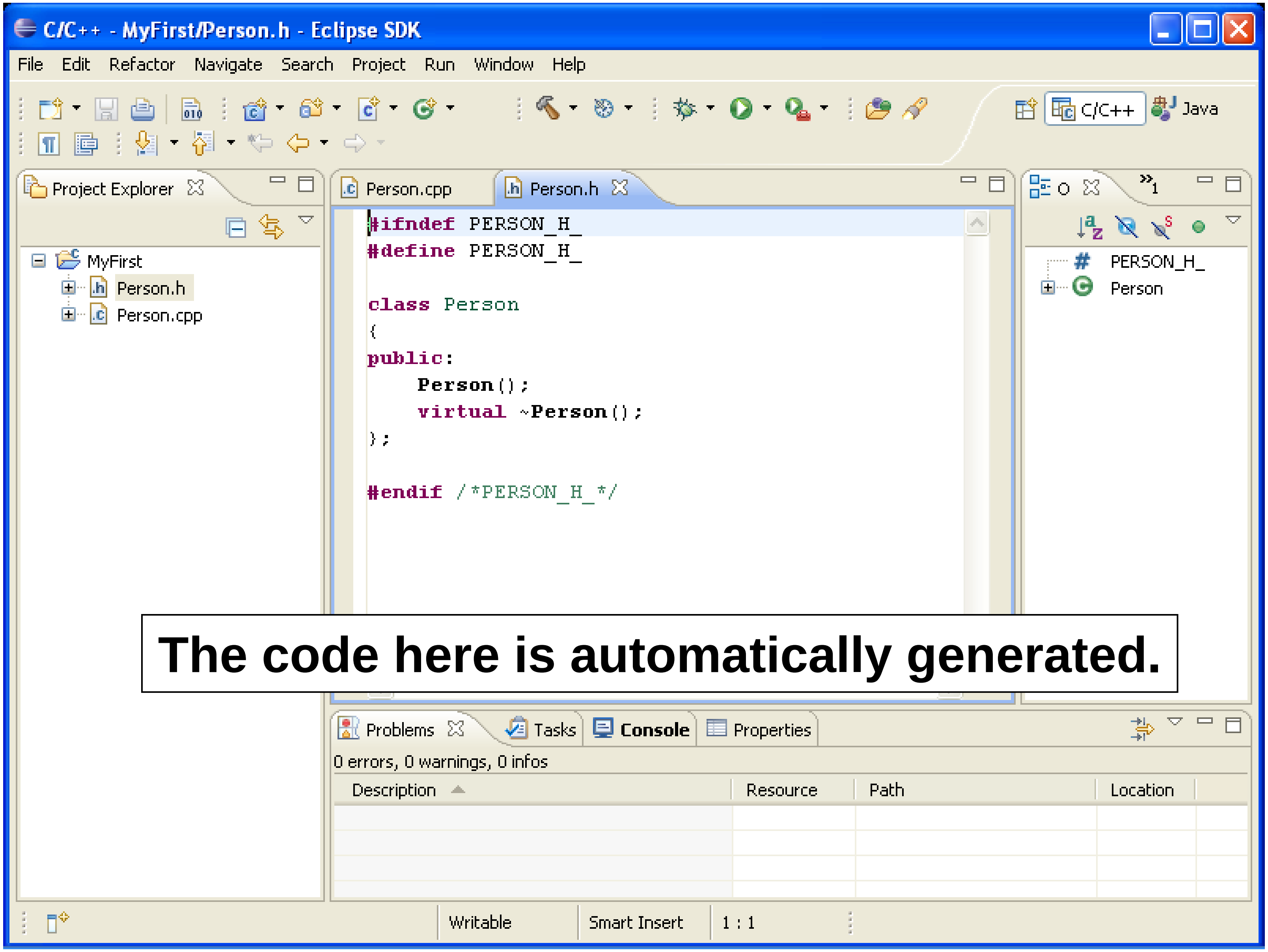
Method Stubs:

Name	Access	Virtual	Inline
☑ ● Constructor	public	no	no
☑ ● Destructor	public	yes	no

☑ Use Default

Header: Person.h

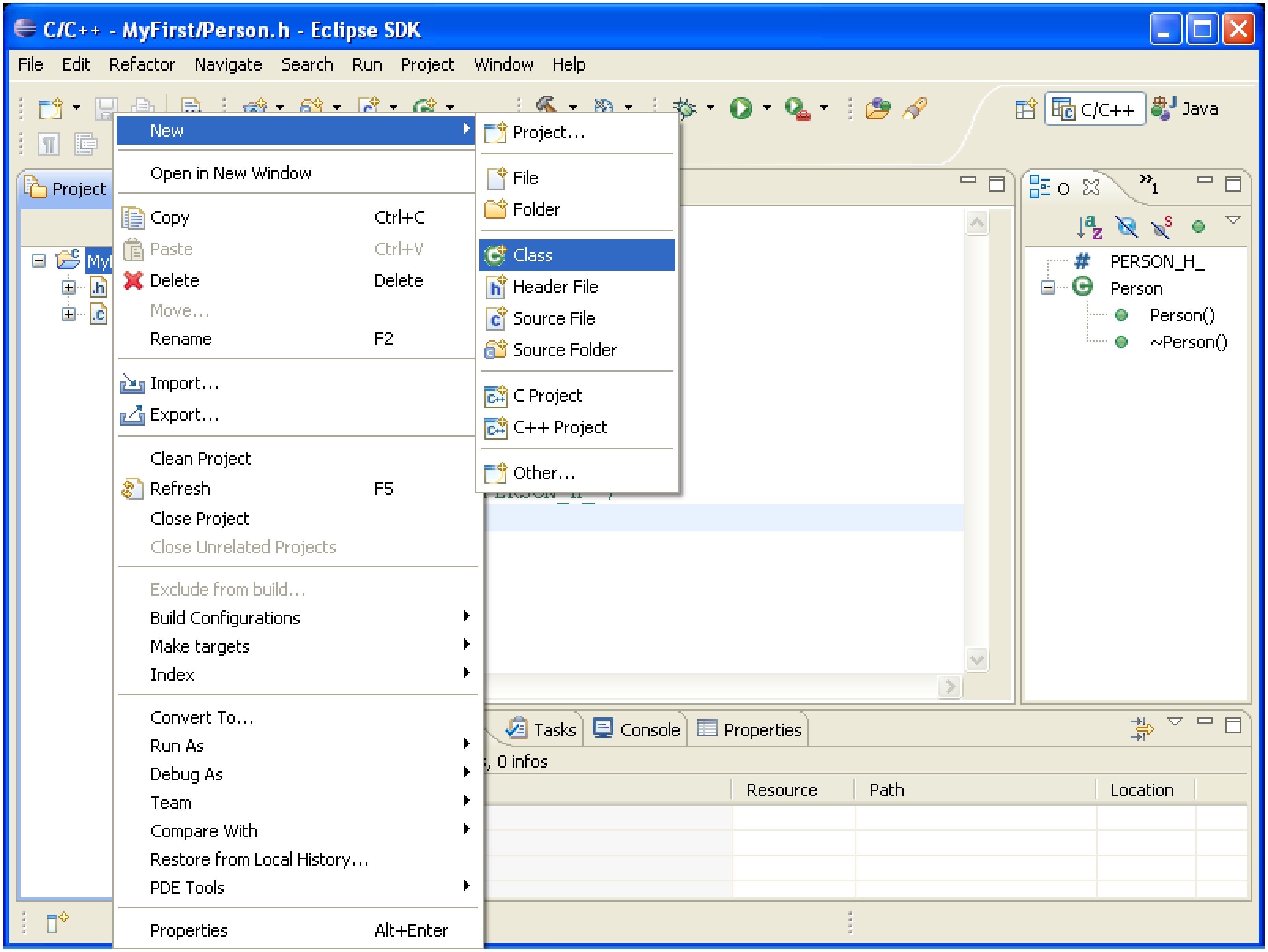
Source: Person.cpp

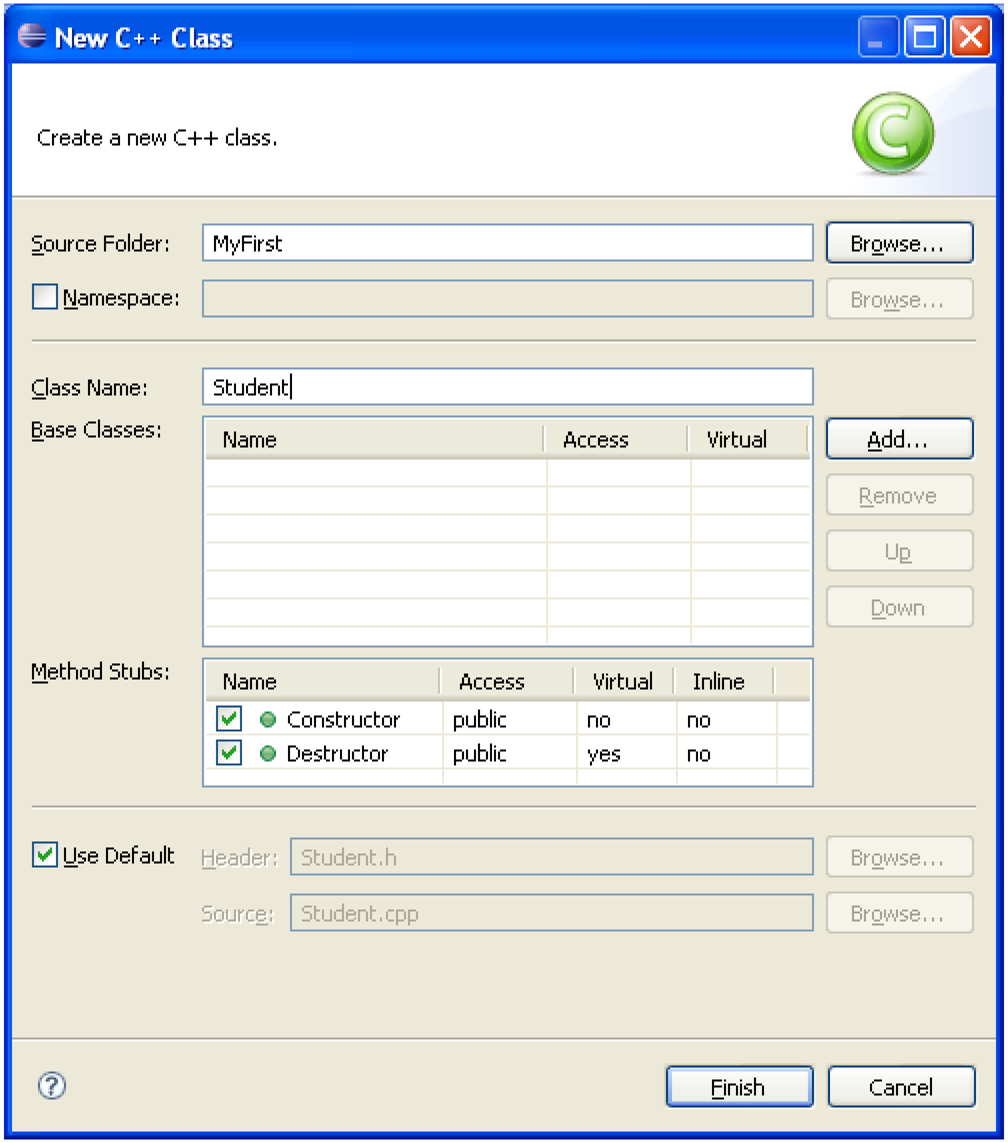


```
#ifndef PERSON_H_
#define PERSON_H_

class Person
{
public:
	Person();
	virtual ~Person();
};

#endif /*PERSON_H_*/
```

0 errors, 0 warnings, 0 infos

The code here is automatically generated.


C/C++ - MyFirst/Person.h - Eclipse SDK
File
Edit
Refactor
Navigate
Search
Run
Project
Window
Help
C/C++
Java
Project
New
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move...
Rename F2
Import...
Export...
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Exclude from build...
Build Configurations
Make targets
Index
Convert To...
Run As
Debug As
Team
Compare With
Restore from Local History...
PDE Tools
Properties Alt+Enter
Project...
File
Folder
Class
Header File
Source File
Source Folder
C Project
C++ Project
Other...
PERSON_H_
Person
Person()
~Person()
Tasks
Console
Properties
, 0 infos
Resource
Path
Location



New C++ Class

Create a new C++ class.

Source Folder: MyFirst

☐ Namespace:

Class Name: Student

Base Classes:

Name	Access	Virtual

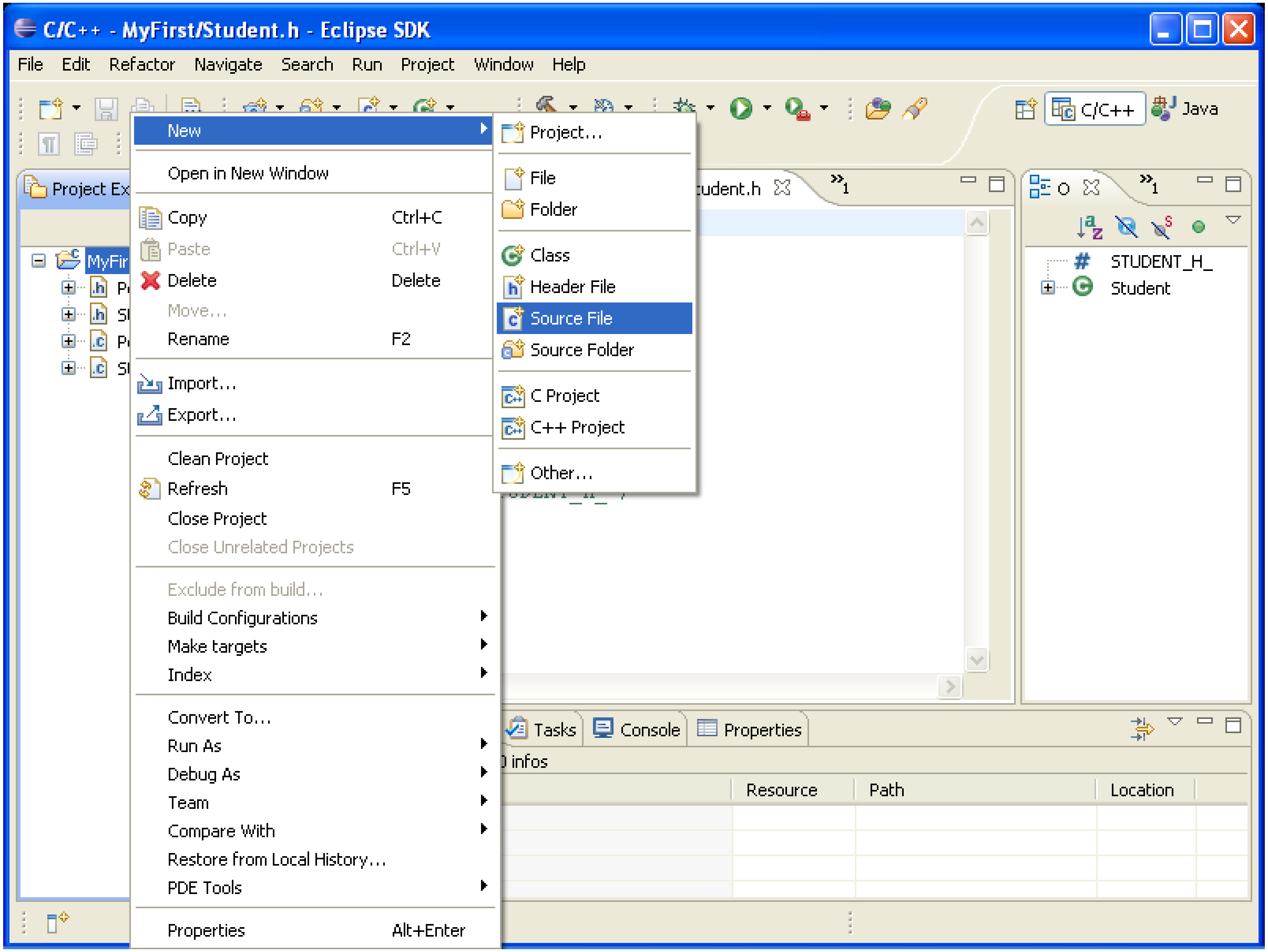
Method Stubs:

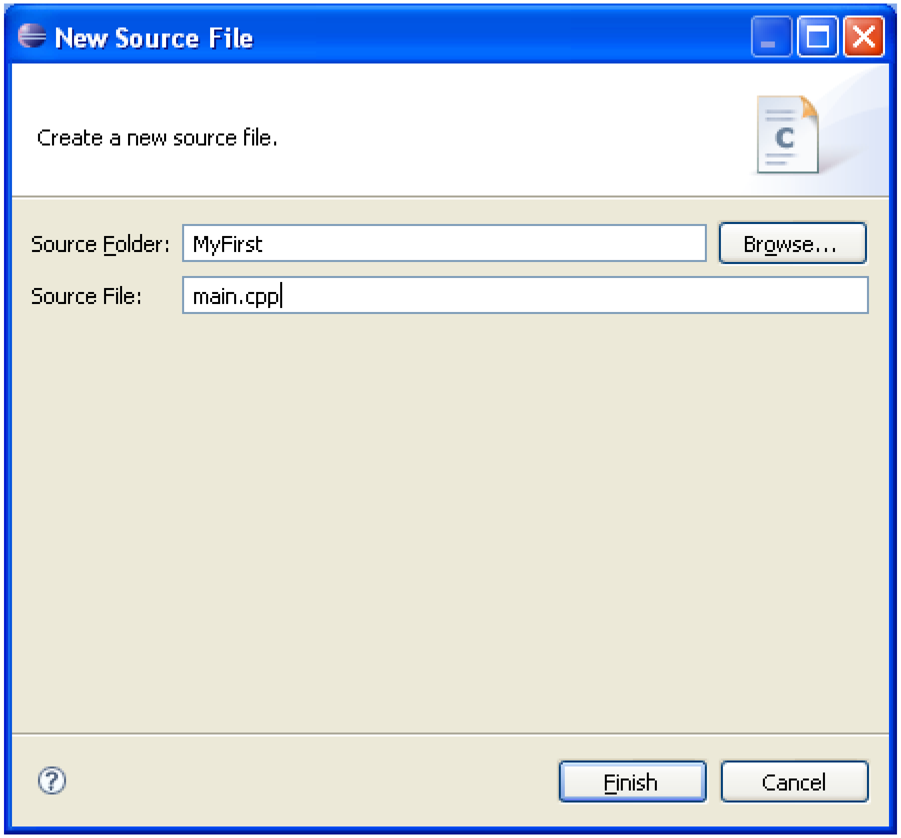
Name	Access	Virtual	Inline
☑ ● Constructor	public	no	no
☑ ● Destructor	public	yes	no

☑ Use Default

Header: Student.h

Source: Student.cpp

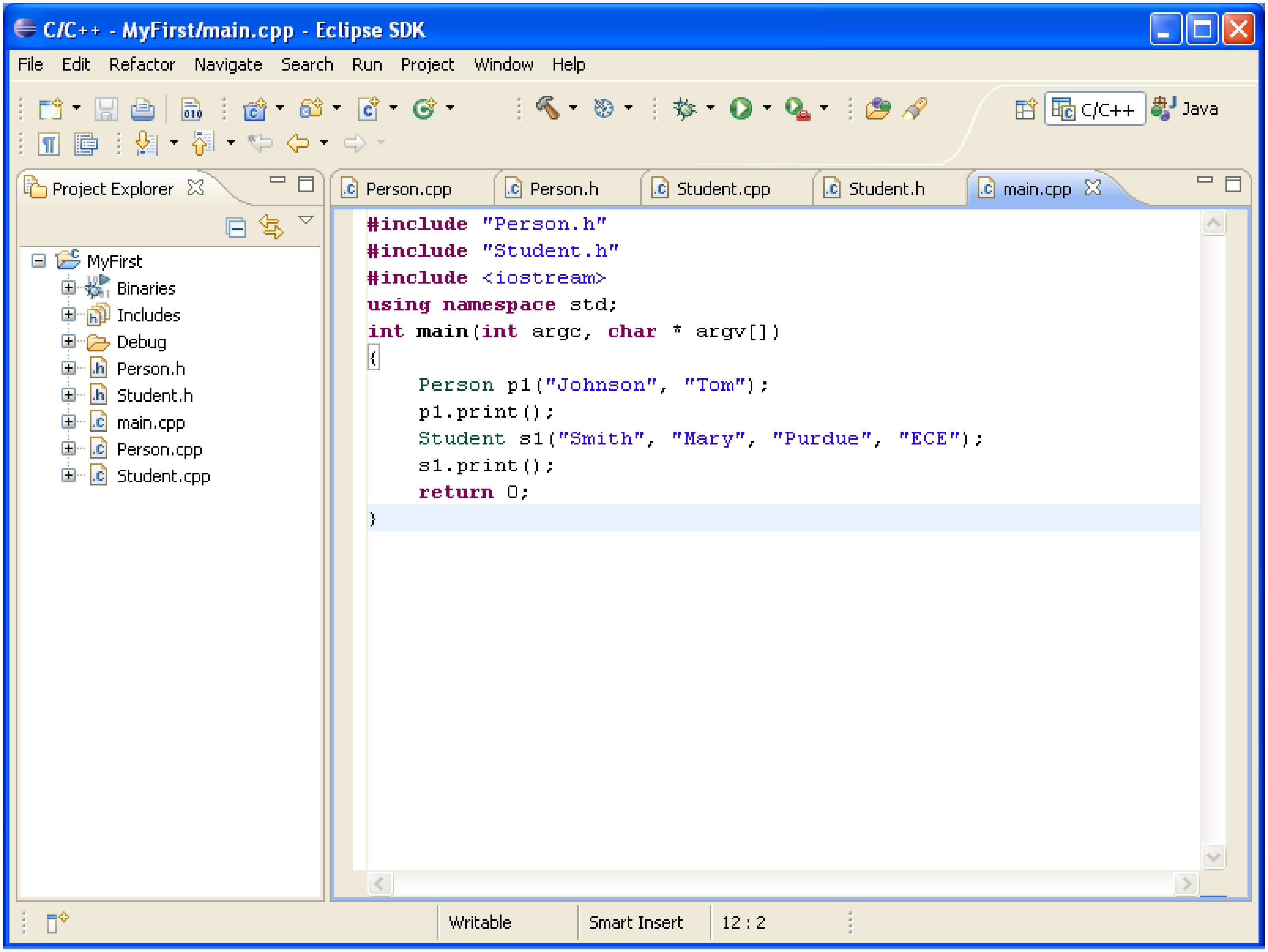

C/C++ - MyFirst/Student.h - Eclipse SDK
File Edit Refactor Navigate Search Run Project Window Help
C/C++
Java
Project Ex
MyFir
New
Open in New Window
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move...
Rename F2
Import...
Export...
Clean Project
Refresh F5
Close Project
Close Unrelated Projects
Exclude from build...
Build Configurations
Make targets
Index
Convert To...
Run As
Debug As
Team
Compare With
Restore from Local History...
PDE Tools
Properties Alt+Enter
Project...
File
Folder
Class
Header File
Source File
Source Folder
C Project
C++ Project
Other...
udent.h
STUDENT_H_
Student
Tasks
Console
Properties
0 infos
Resource
Path
Location



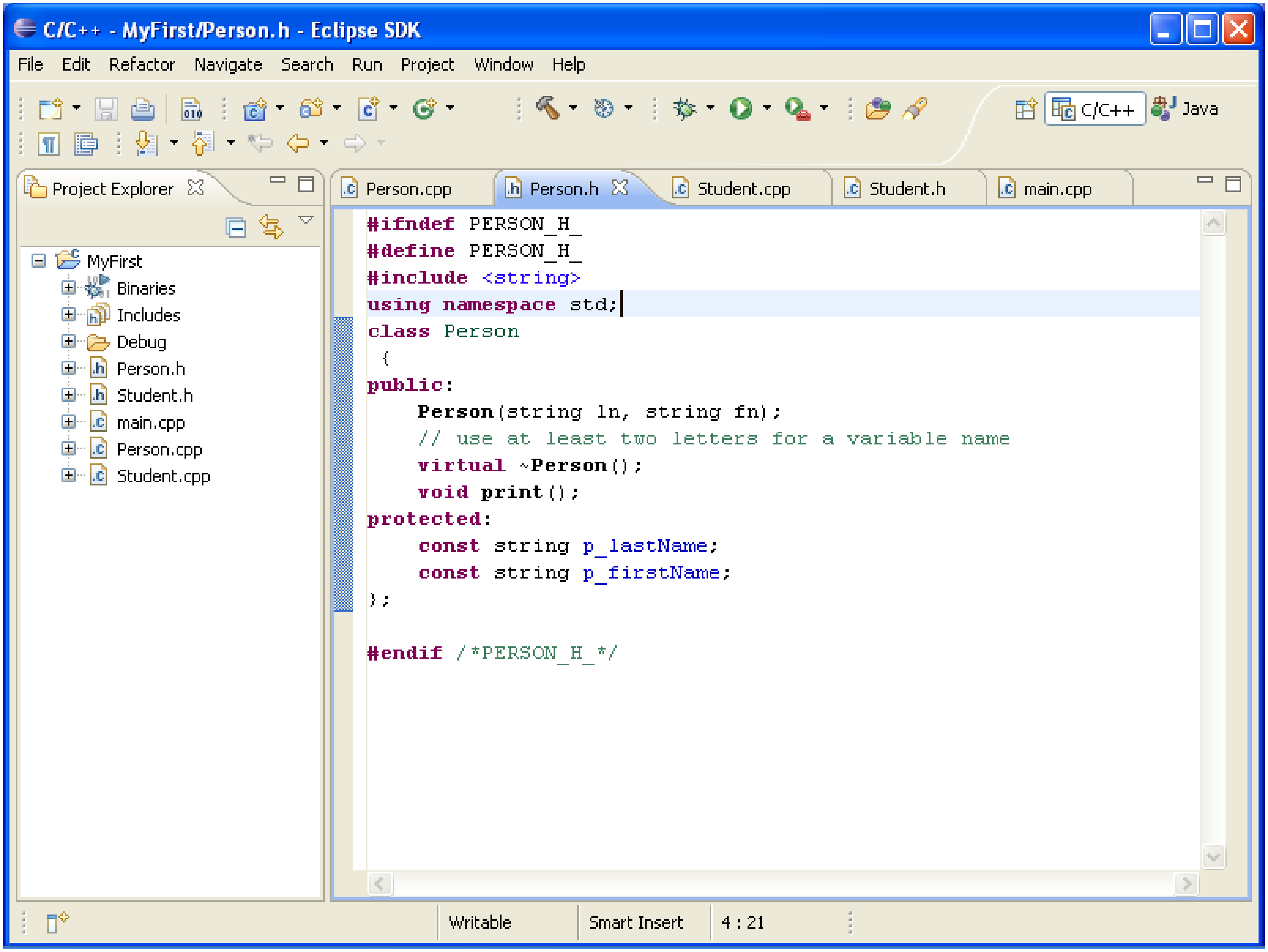
Create a new source file.

Source Folder: MyFirst

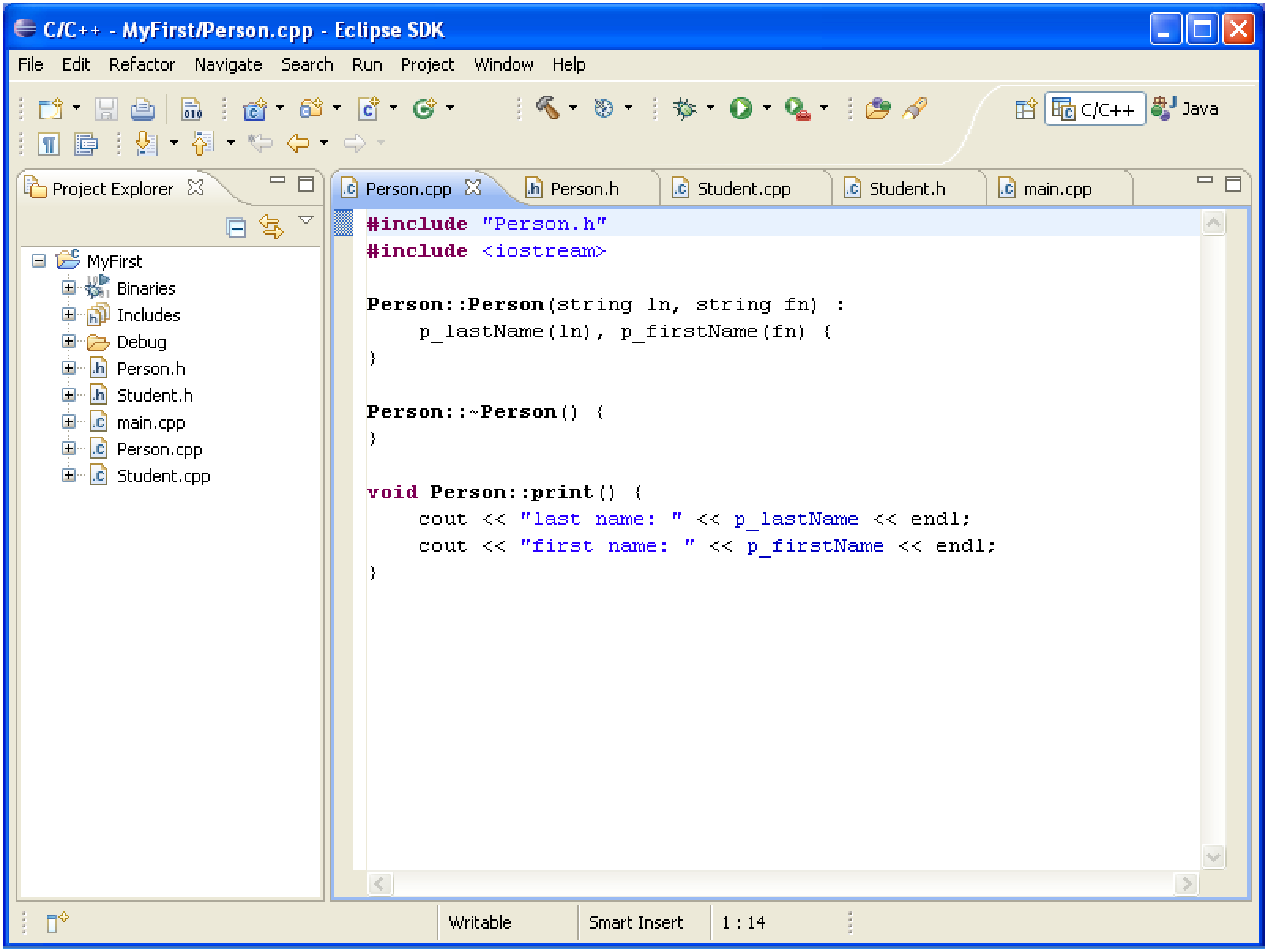
Source File: main.cpp



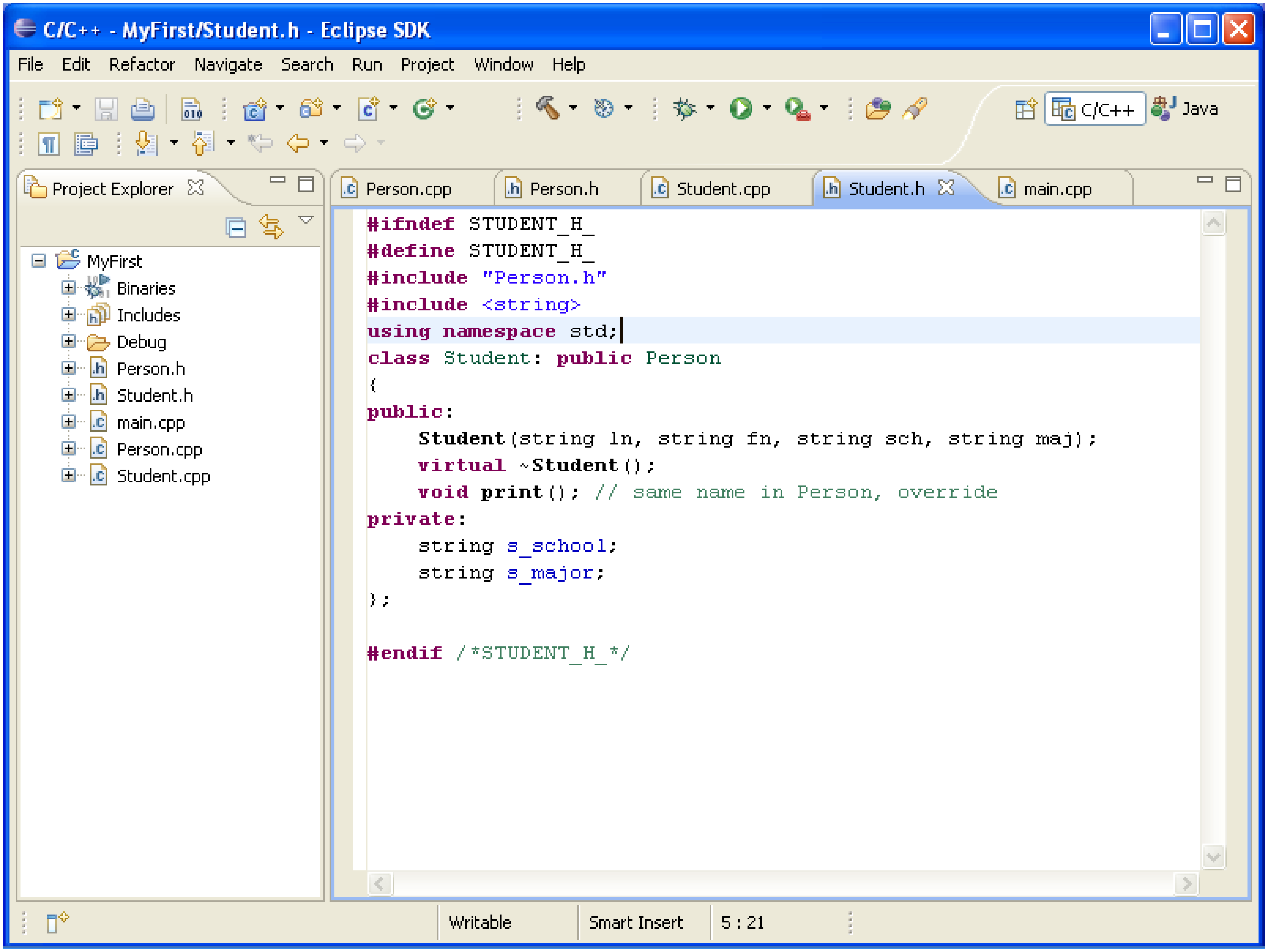
```
#include "Person.h"
#include "Student.h"
#include <iostream>
using namespace std;
int main(int argc, char * argv[])
{
    Person p1("Johnson", "Tom");
    p1.print();
    Student s1("Smith", "Mary", "Purdue", "ECE");
    s1.print();
    return 0;
}
```



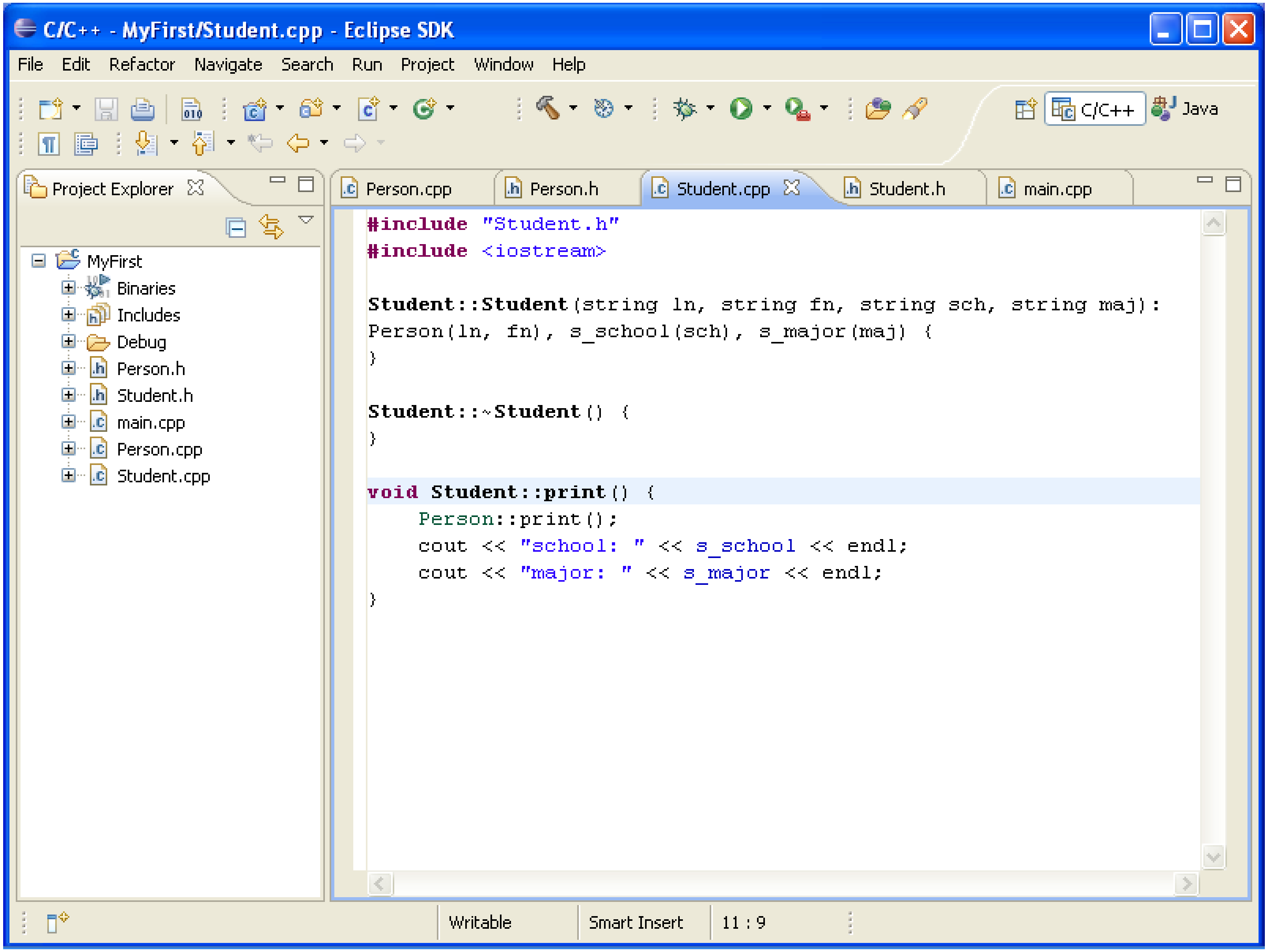
```
#ifndef PERSON_H_
#define PERSON_H_
#include <string>
using namespace std;
class Person
 {
public:
    Person(string ln, string fn);
    // use at least two letters for a variable name
    virtual ~Person();
    void print();
protected:
    const string p_lastName;
    const string p_firstName;
};

#endif /*PERSON_H_*/
```



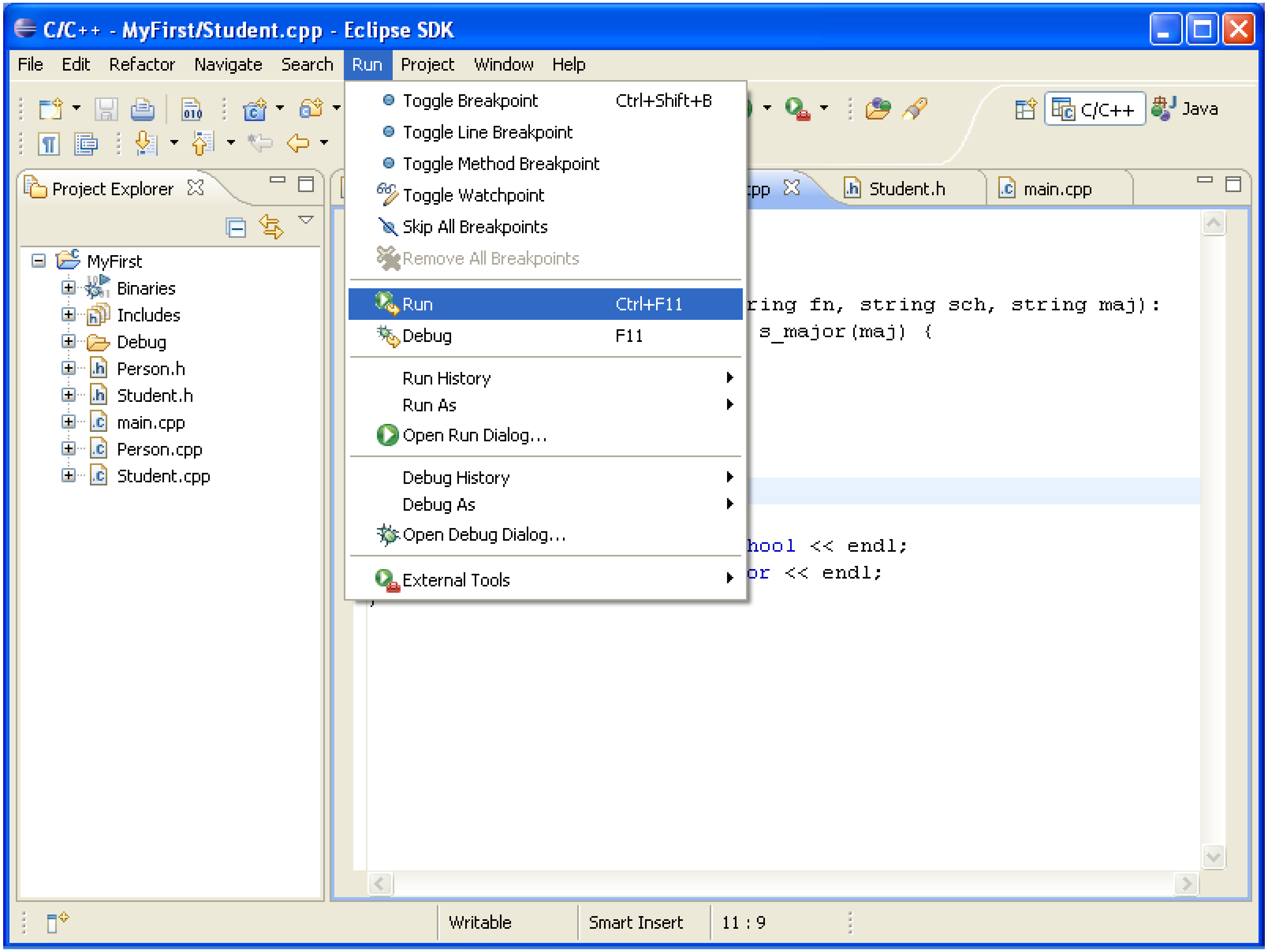
```
#include "Person.h"
#include <iostream>

Person::Person(string ln, string fn) :
    p_lastName(ln), p_firstName(fn) {
}

Person::~Person() {
}

void Person::print() {
    cout << "last name: " << p_lastName << endl;
    cout << "first name: " << p_firstName << endl;
}
```



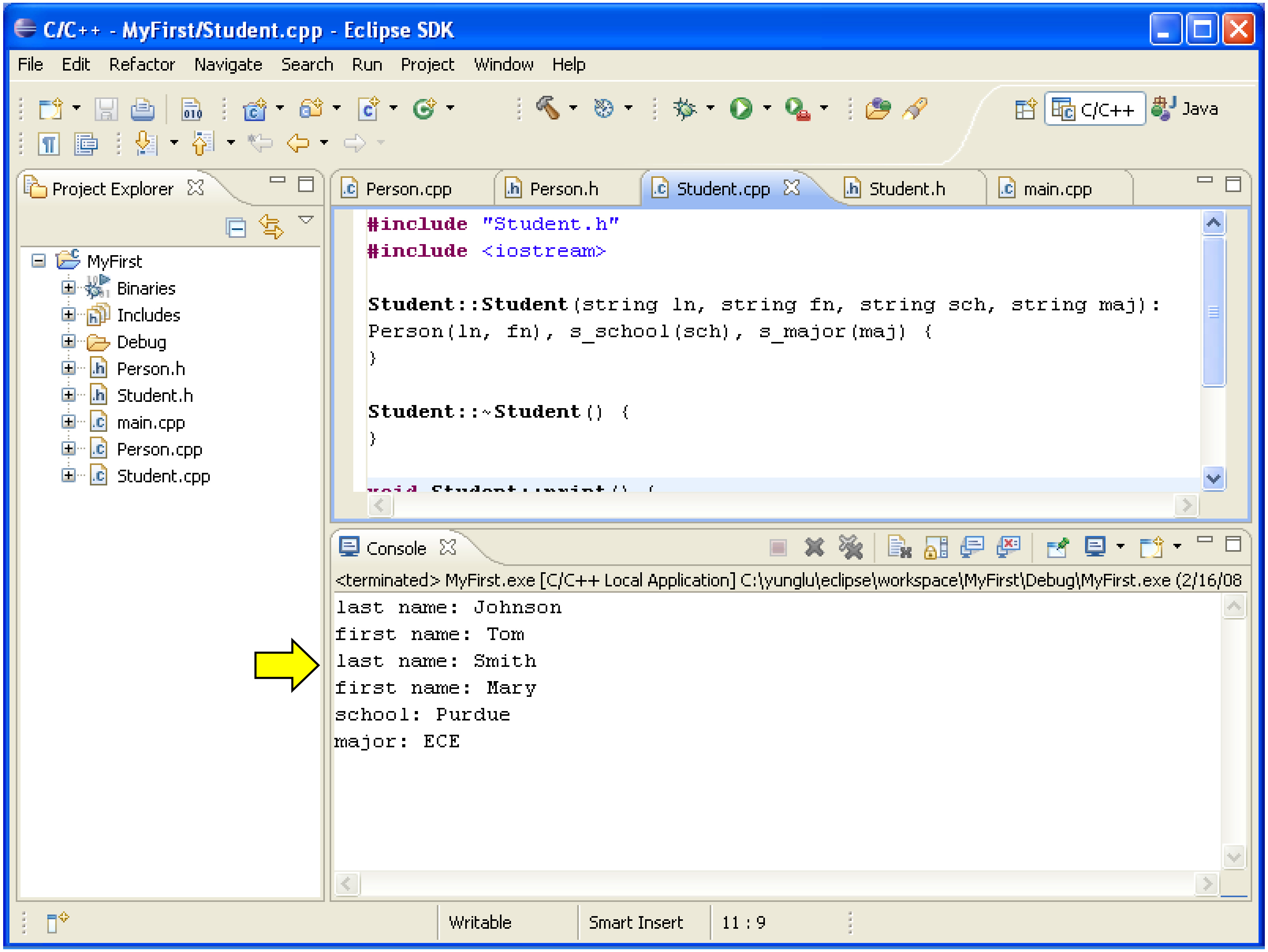
```
#ifndef STUDENT_H_
#define STUDENT_H_
#include "Person.h"
#include <string>
using namespace std;
class Student: public Person
{
public:
    Student(string ln, string fn, string sch, string maj);
    virtual ~Student();
    void print(); // same name in Person, override
private:
    string s_school;
    string s_major;
};

#endif /*STUDENT_H_*/
```



```
#include "Student.h"
#include <iostream>

Student::Student(string ln, string fn, string sch, string maj):
Person(ln, fn), s_school(sch), s_major(maj) {
}

Student::~Student() {
}

void Student::print() {
    Person::print();
    cout << "school: " << s_school << endl;
    cout << "major: " << s_major << endl;
}
```



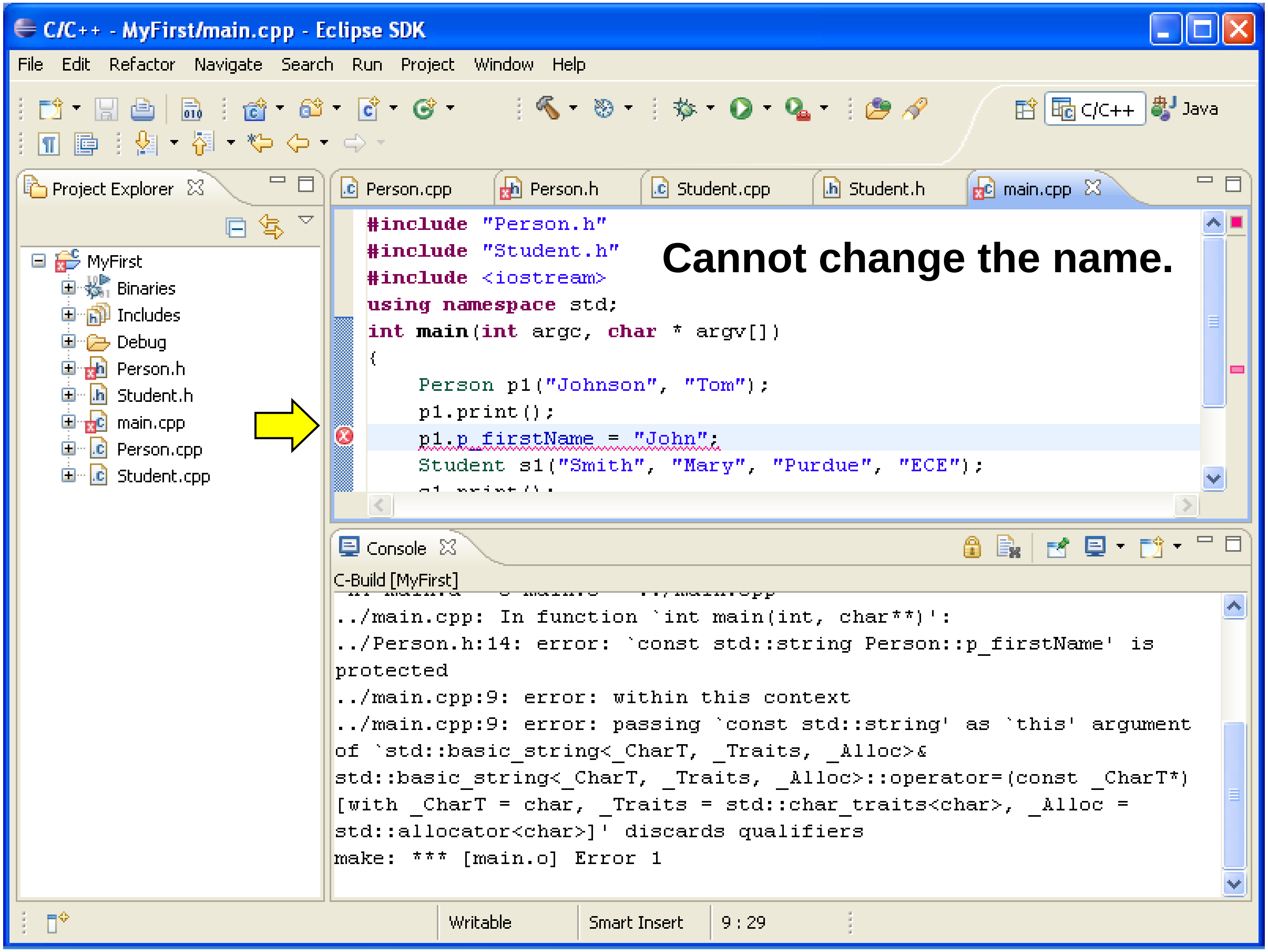
```
ring fn, string sch, string maj):
 s_major(maj) {




hool << endl;
or << endl;
```



```
#include "Student.h"
#include <iostream>

Student::Student(string ln, string fn, string sch, string maj):
Person(ln, fn), s_school(sch), s_major(maj) {
}

Student::~Student() {
}

void Student::print() {
```

<terminated> MyFirst.exe [C/C++ Local Application] C:\yunglu\eclipse\workspace\MyFirst\Debug\MyFirst.exe (2/16/08

```
last name: Johnson
first name: Tom
last name: Smith
first name: Mary
school: Purdue
major: ECE
```

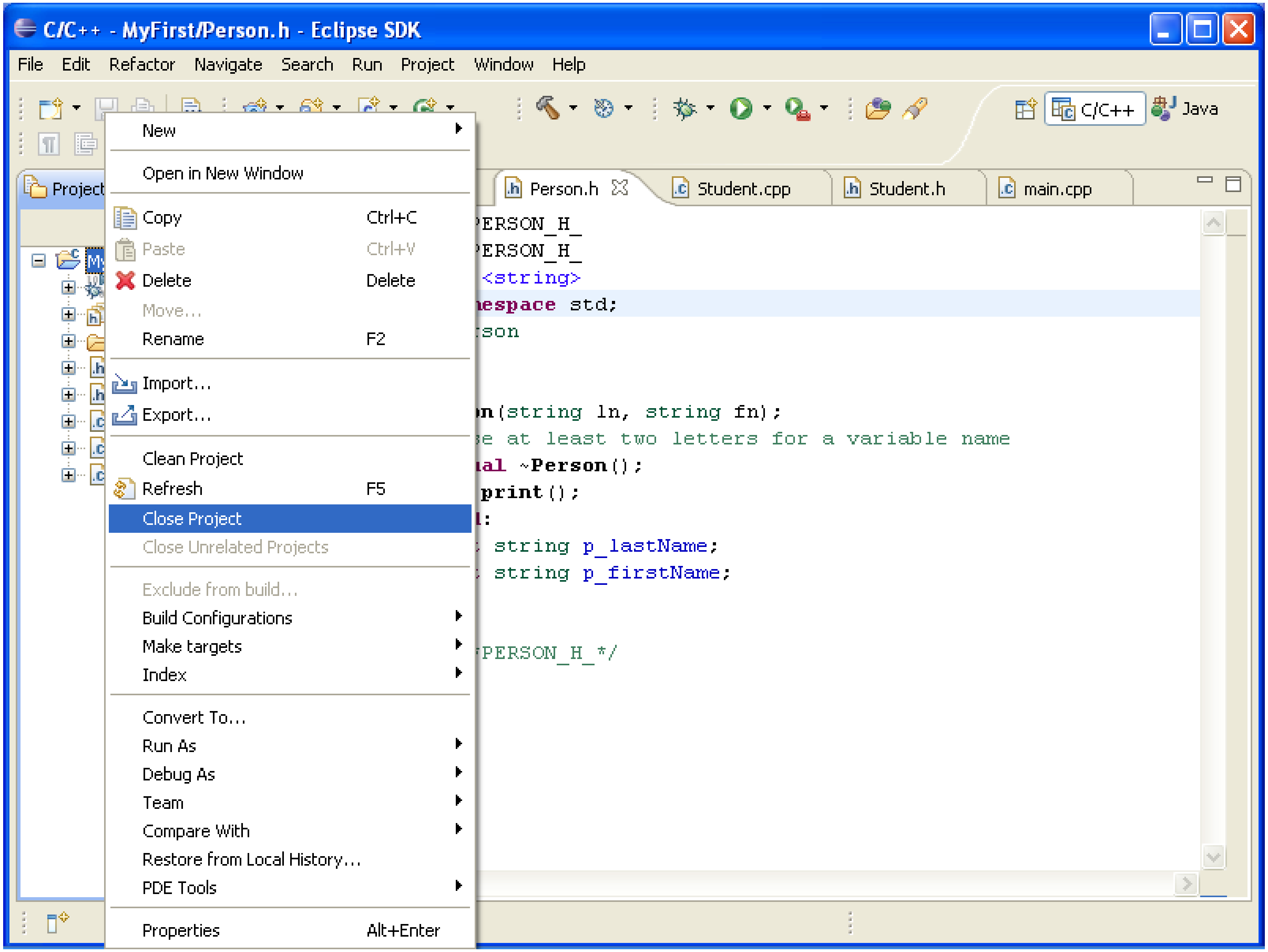


Cannot change the name.

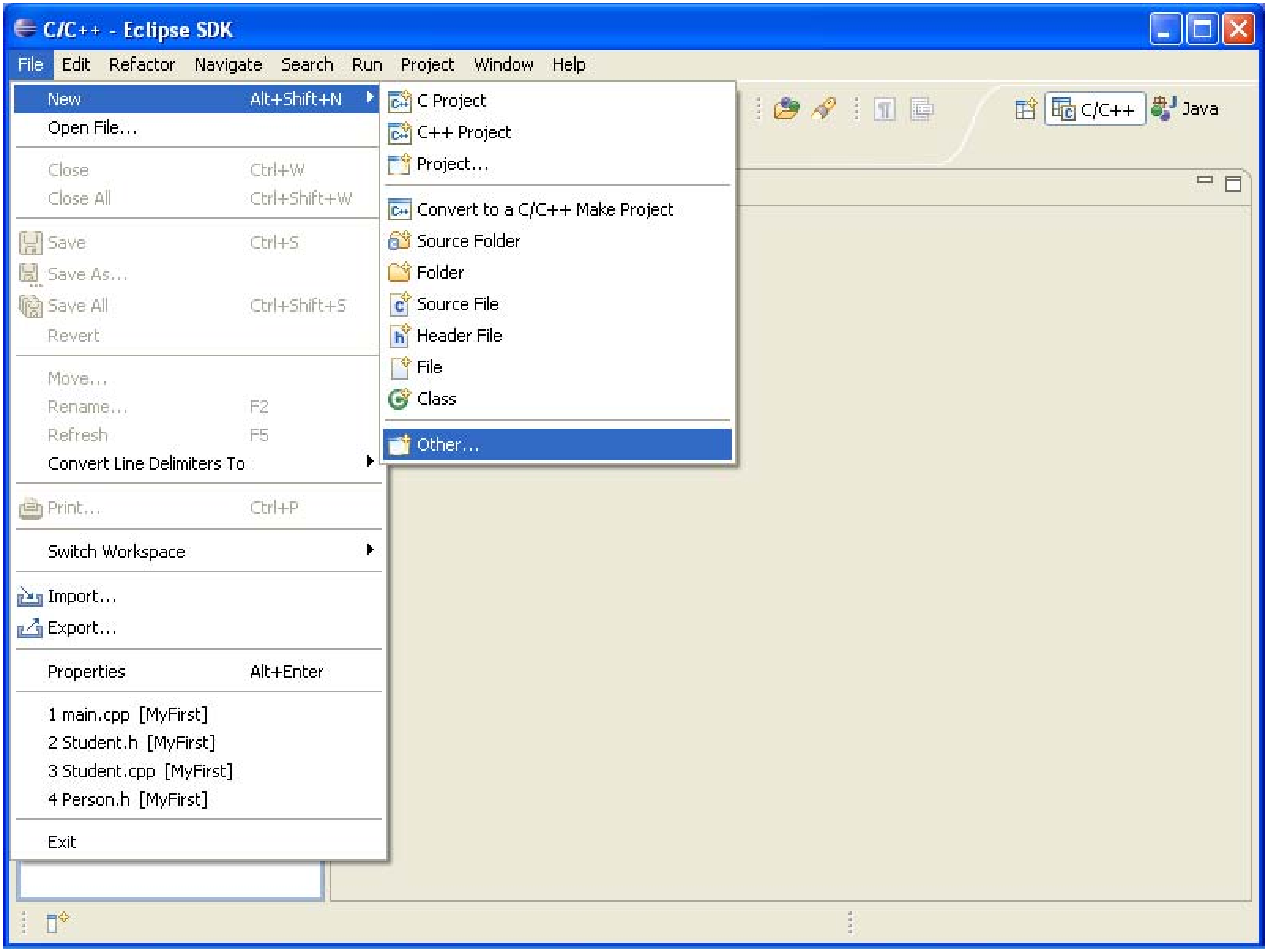
```
#include "Person.h"
#include "Student.h"
#include <iostream>
using namespace std;
int main(int argc, char * argv[])
{
    Person p1("Johnson", "Tom");
    p1.print();
    p1.p_firstName = "John";
    Student s1("Smith", "Mary", "Purdue", "ECE");
    s1.print();
```

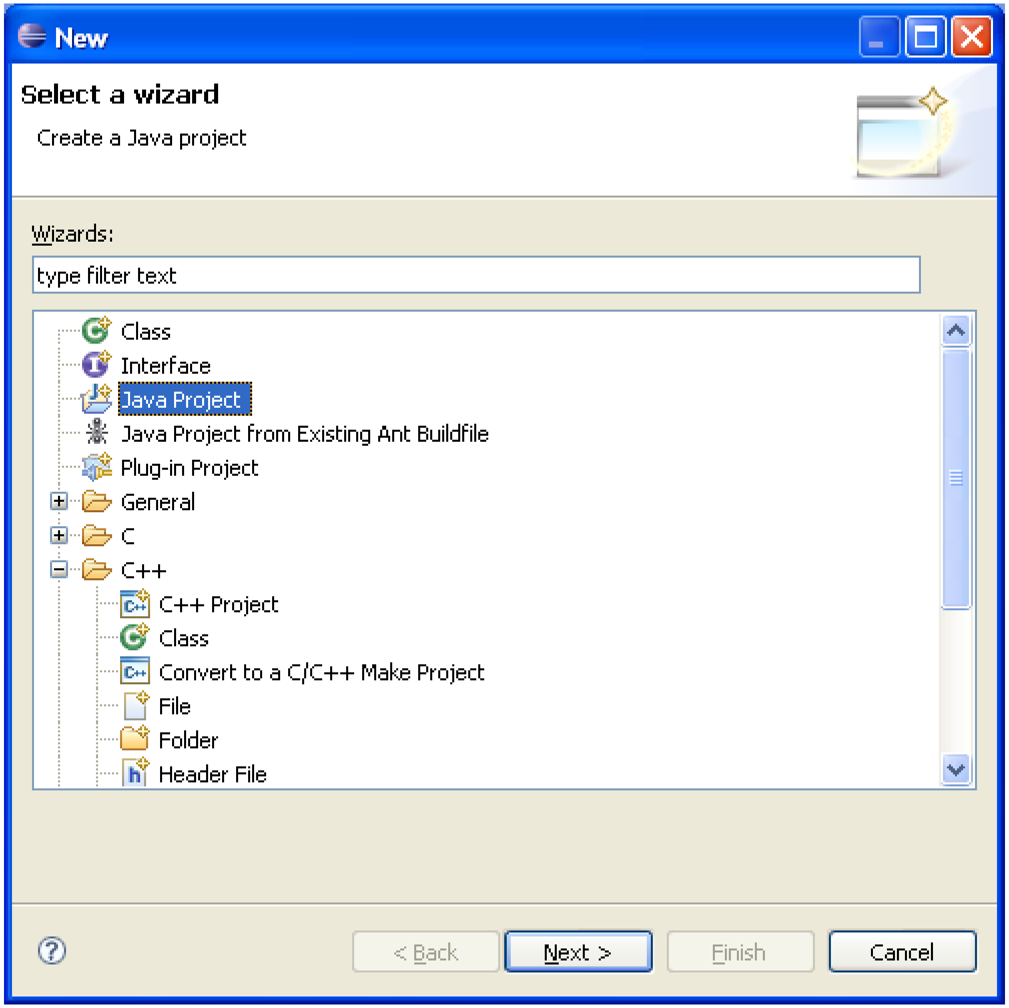
```
../main.cpp: In function `int main(int, char**)':
../Person.h:14: error: `const std::string Person::p_firstName' is
protected
../main.cpp:9: error: within this context
../main.cpp:9: error: passing `const std::string' as `this' argument
of `std::basic_string<_CharT, _Traits, _Alloc>&
std::basic_string<_CharT, _Traits, _Alloc>::operator=(const _CharT*)
[with _CharT = char, _Traits = std::char_traits<char>, _Alloc =
std::allocator<char>]' discards qualifiers
make: *** [main.o] Error 1
```

Develop Java Projects in Eclipse



```
ERSON_H_
ERSON_H_
 <string>
espace std;
son


n(string ln, string fn);
e at least two letters for a variable name
al ~Person();
print();
:
  string p_lastName;
  string p_firstName;


PERSON_H_*/
```


C/C++ - Eclipse SDK
File
Edit
Refactor
Navigate
Search
Run
Project
Window
Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Import...
Export...
Properties Alt+Enter
1 main.cpp [MyFirst]
2 Student.h [MyFirst]
3 Student.cpp [MyFirst]
4 Person.h [MyFirst]
Exit
C Project
C++ Project
Project...
Convert to a C/C++ Make Project
Source Folder
Folder
Source File
Header File
File
Class
Other...
C/C++
Java

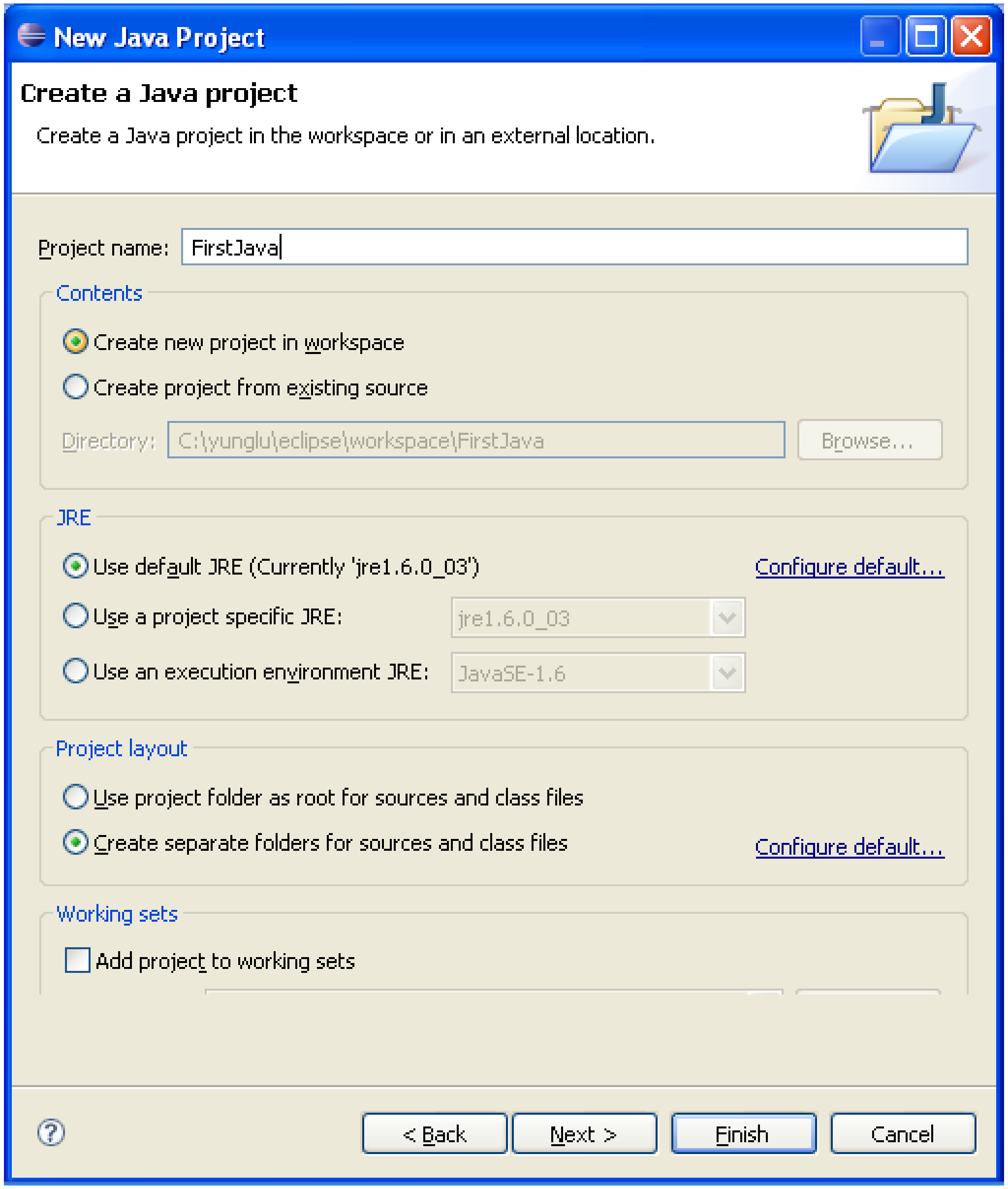


New

Select a wizard

Create a Java project

Wizards: type filter text



New Java Project

Create a Java project

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

Project name: FirstJava

Contents

● Create new project in workspace

○ Create project from existing source

Directory: C:\yunglu\eclipse\workspace\FirstJava

JRE

● Use default JRE (Currently 'jre1.6.0_03')

○ Use a project specific JRE: jre1.6.0_03

○ Use an execution environment JRE: JavaSE-1.6

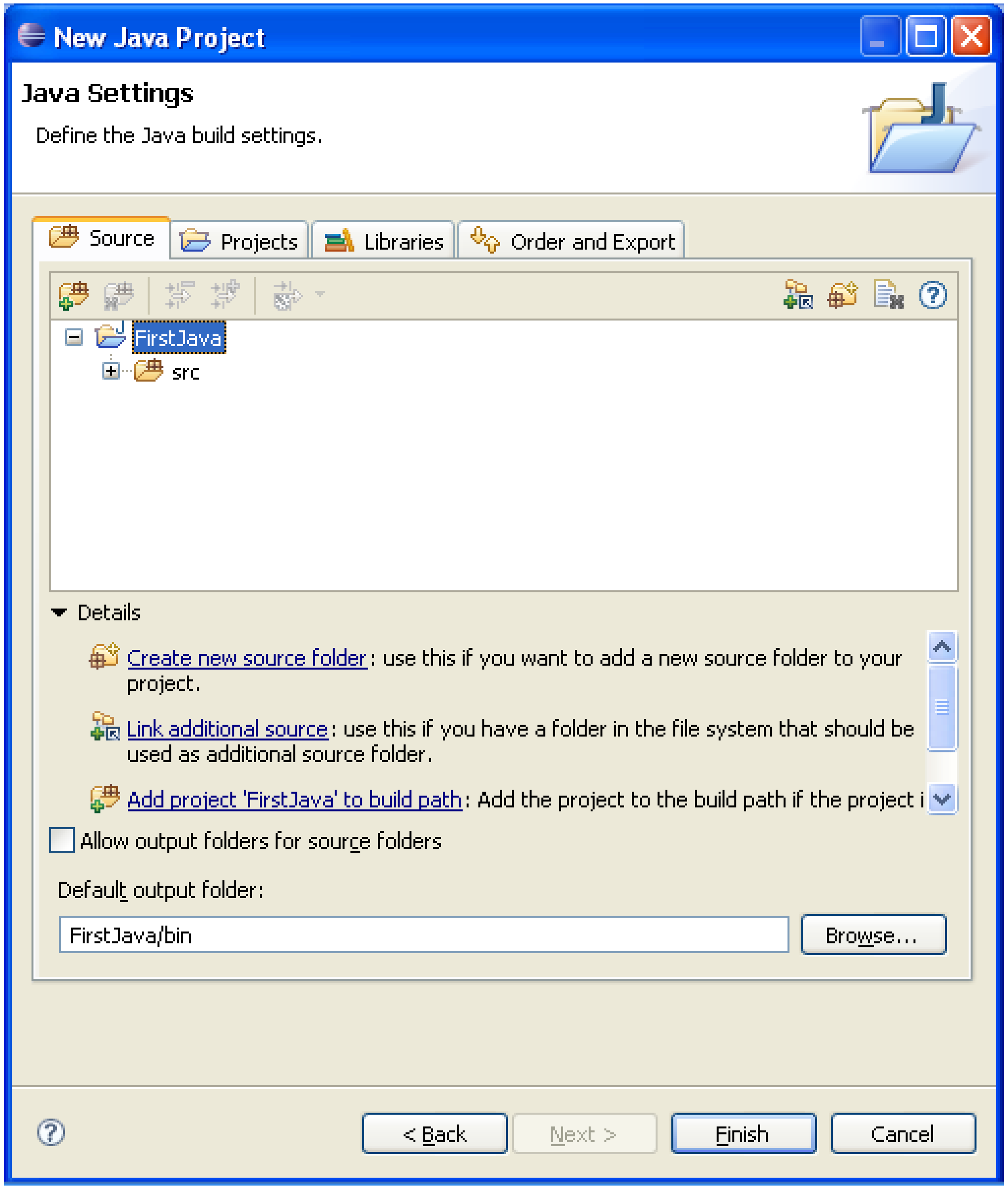
Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

□ Add project to working sets



Java Settings

Define the Java build settings.

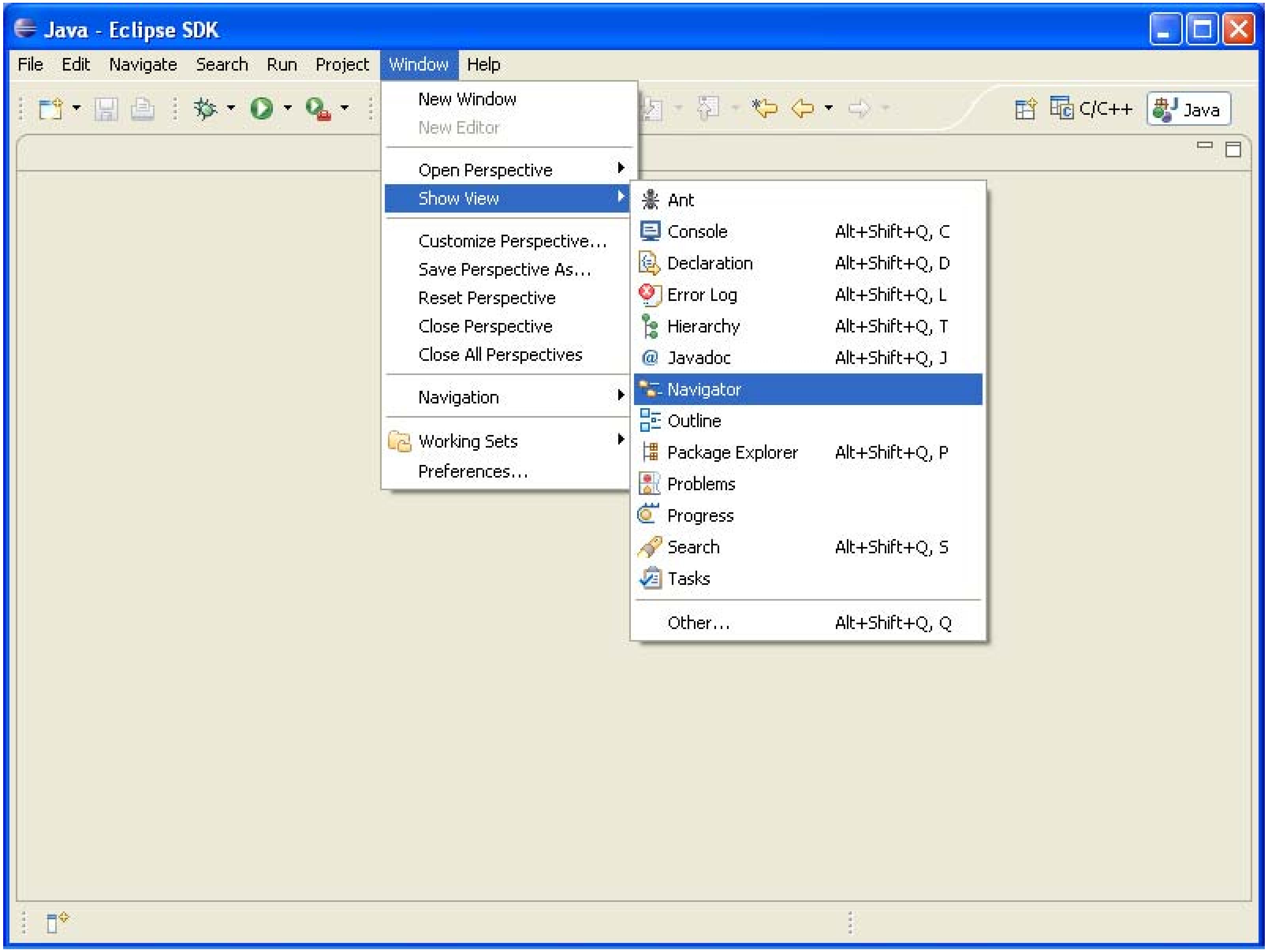
Create new source folder: use this if you want to add a new source folder to your project.

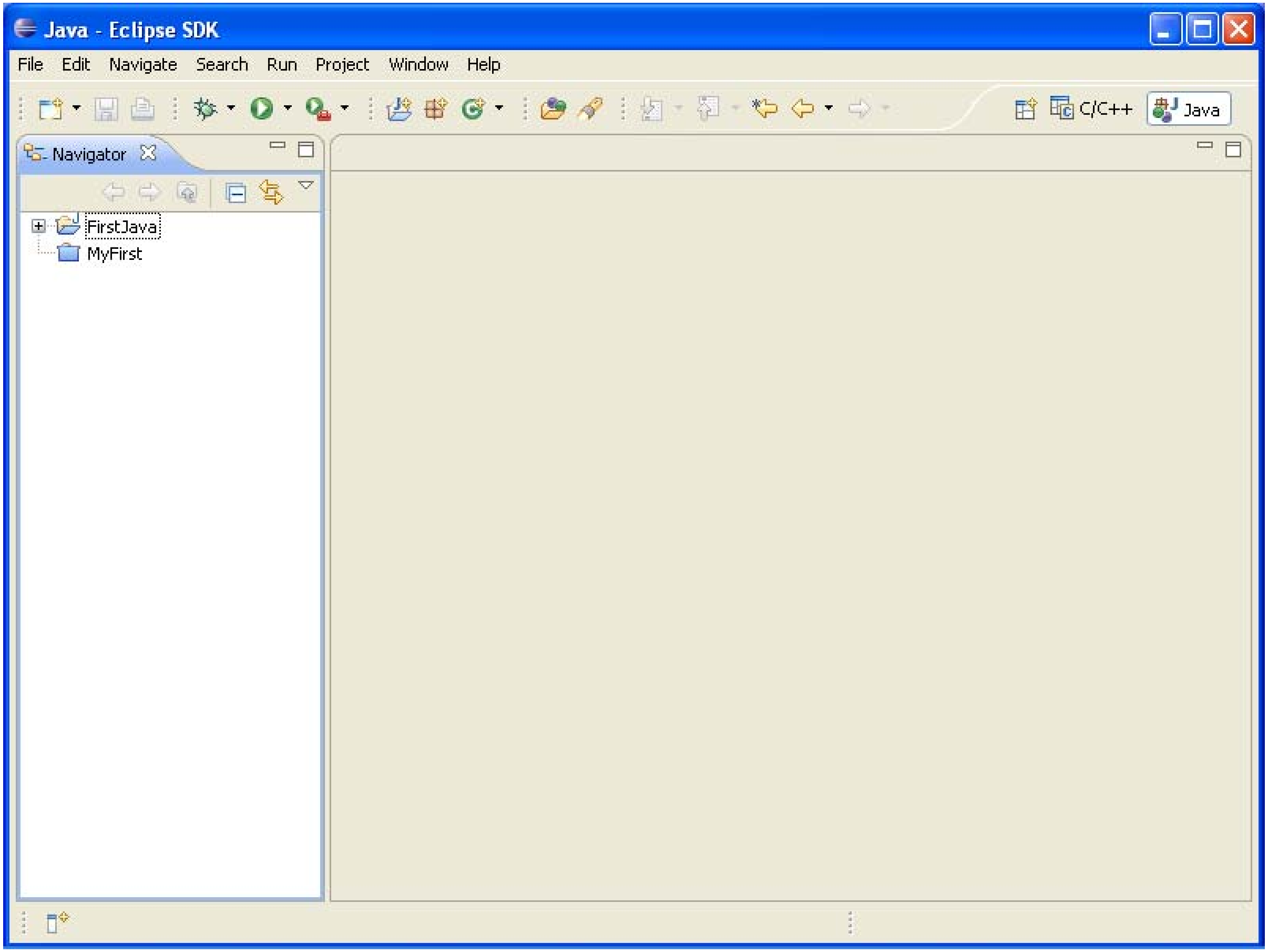
Link additional source: use this if you have a folder in the file system that should be used as additional source folder.

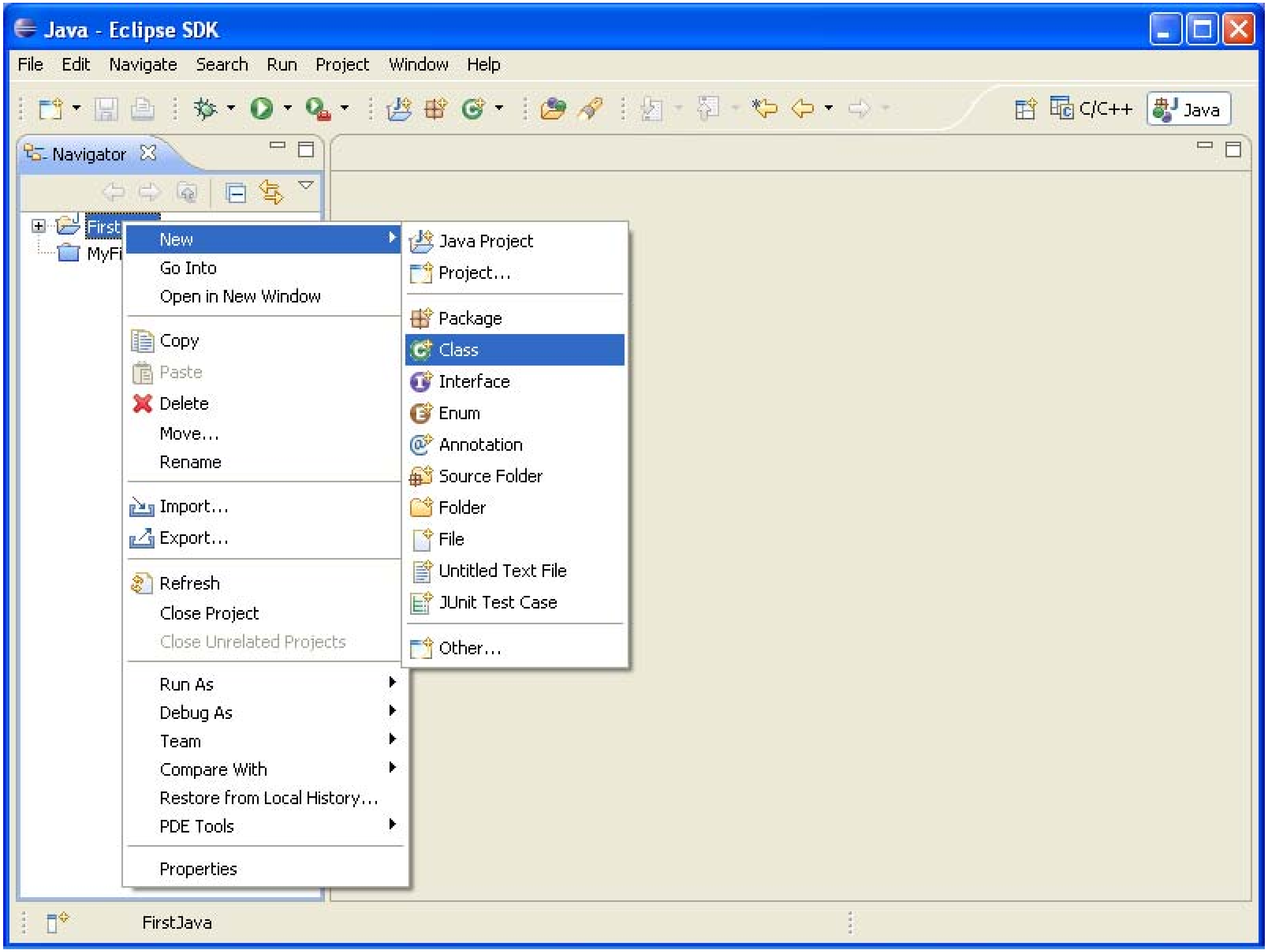
Add project 'FirstJava' to build path: Add the project to the build path if the project i

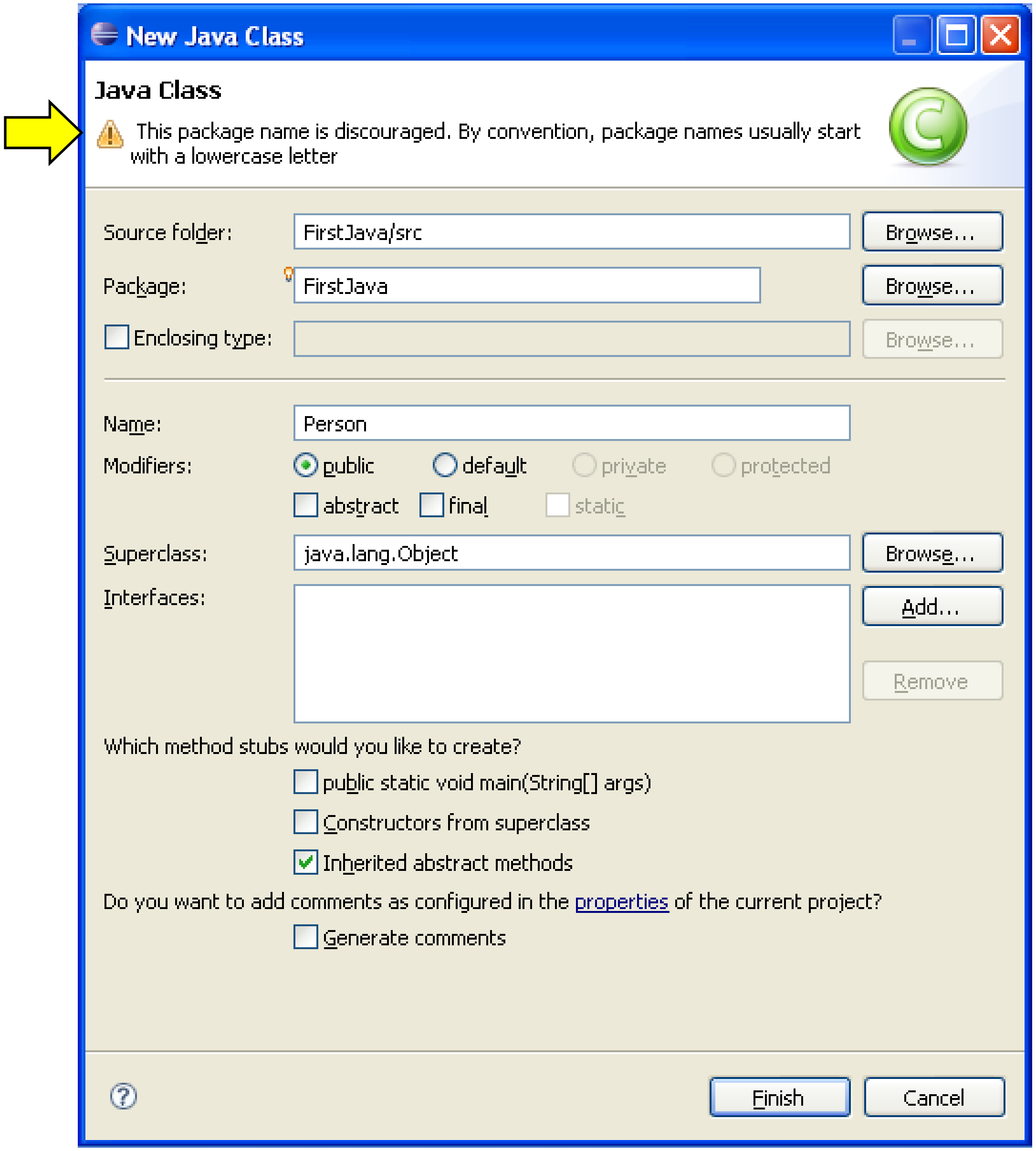
☐ Allow output folders for source folders

Default output folder: FirstJava/bin


Java - Eclipse SDK
File
Edit
Navigate
Search
Run
Project
Window
Help
C/C++
Java
New Window
New Editor
Open Perspective
Show View
Customize Perspective...
Save Perspective As...
Reset Perspective
Close Perspective
Close All Perspectives
Navigation
Working Sets
Preferences...
Ant
Console Alt+Shift+Q, C
Declaration Alt+Shift+Q, D
Error Log Alt+Shift+Q, L
Hierarchy Alt+Shift+Q, T
Javadoc Alt+Shift+Q, J
Navigator
Outline
Package Explorer Alt+Shift+Q, P
Problems
Progress
Search Alt+Shift+Q, S
Tasks
Other... Alt+Shift+Q, Q


Java - Eclipse SDK
File
Edit
Navigate
Search
Run
Project
Window
Help
C/C++
Java
Navigator
FirstJava
MyFirst


Java - Eclipse SDK
File Edit Navigate Search Run Project Window Help
C/C++
Java
Navigator
First
MyFi
New
Go Into
Open in New Window
Copy
Paste
Delete
Move...
Rename
Import...
Export...
Refresh
Close Project
Close Unrelated Projects
Run As
Debug As
Team
Compare With
Restore from Local History...
PDE Tools
Properties
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Folder
File
Untitled Text File
JUnit Test Case
Other...
FirstJava



Java Class

⚠ This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: FirstJava/src

Package: FirstJava

□ Enclosing type:

Name: Person

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

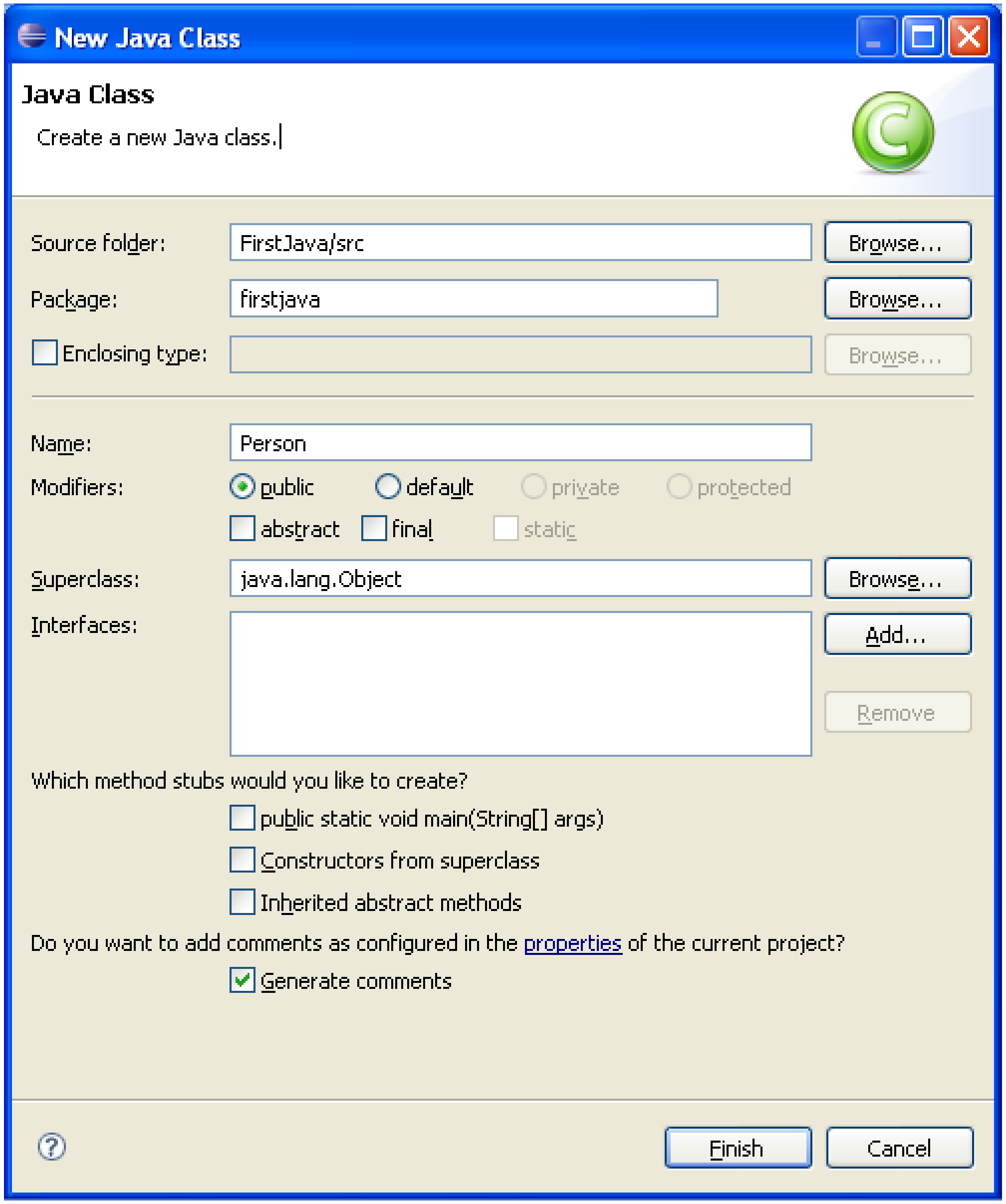
□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

□ Generate comments



Java Class

Create a new Java class.

Source folder: FirstJava/src

Package: firstjava

□ Enclosing type:

Name: Person

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

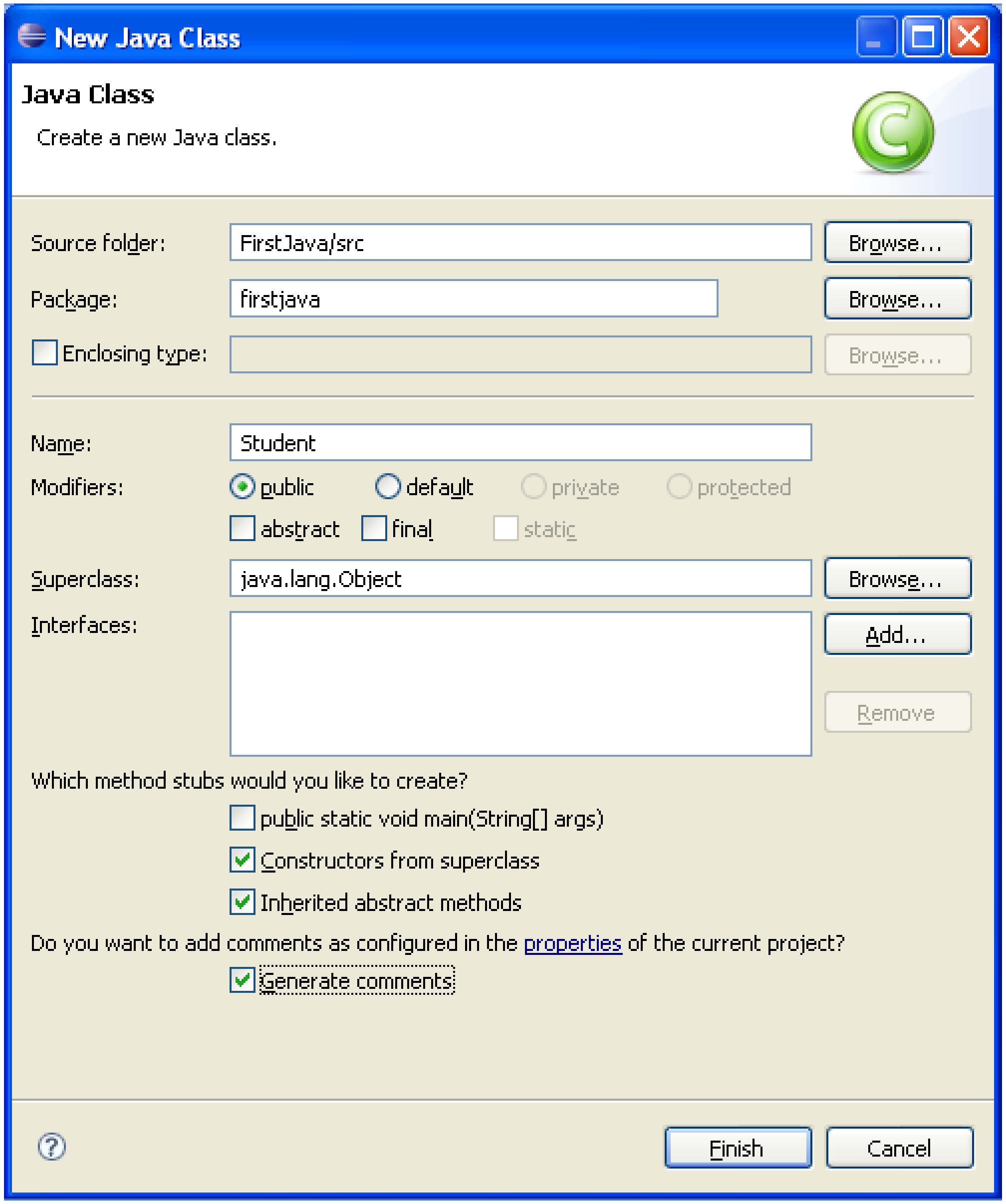
□ public static void main(String[] args)

□ Constructors from superclass

□ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

☑ Generate comments



Java Class

Create a new Java class.

Source folder: FirstJava/src

Package: firstjava

□ Enclosing type:

Name: Student

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

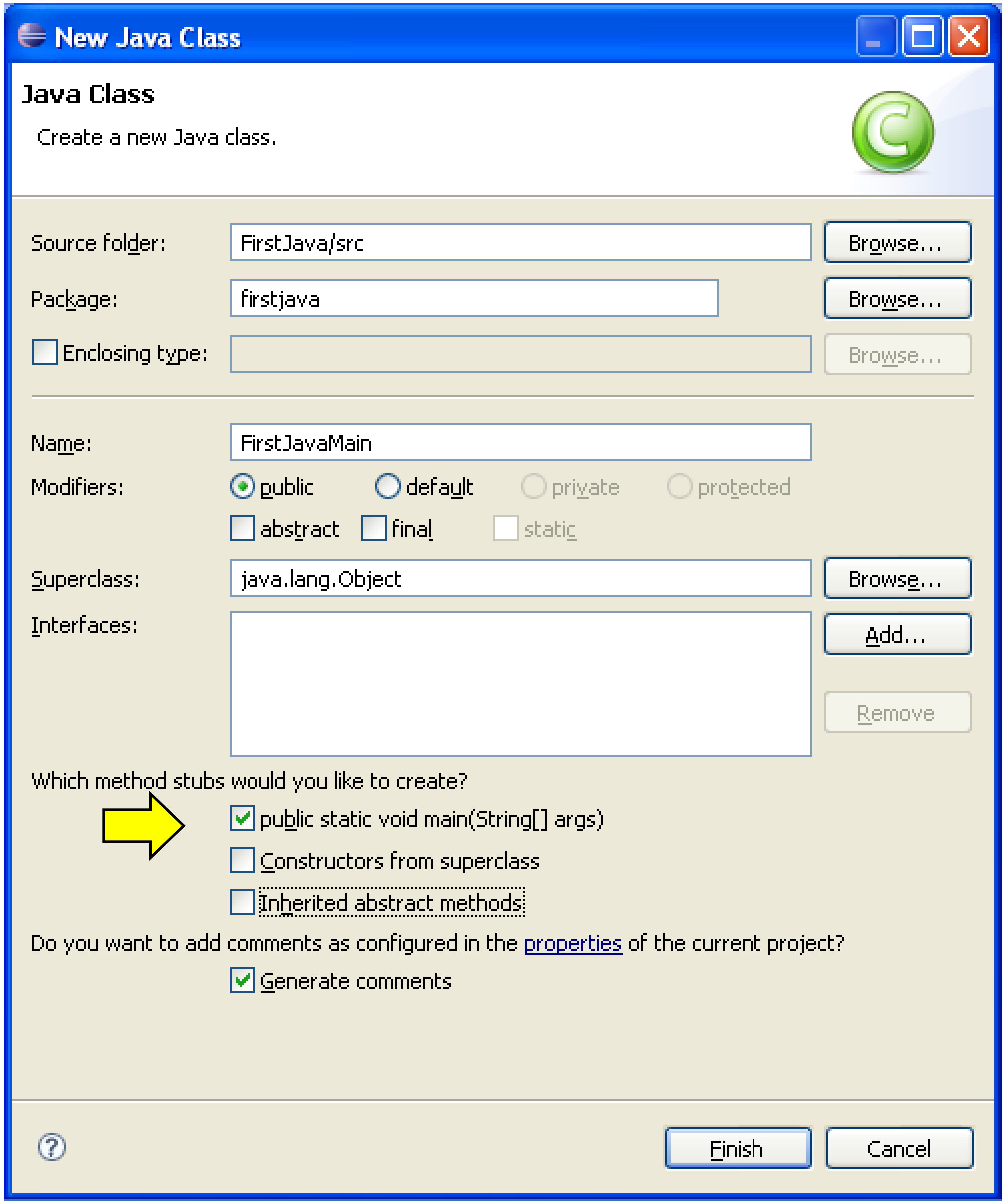
□ public static void main(String[] args)

☑ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

☑ Generate comments



New Java Class

Java Class

Create a new Java class.

Source folder: FirstJava/src

Package: firstjava

□ Enclosing type:

Name: FirstJavaMain

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

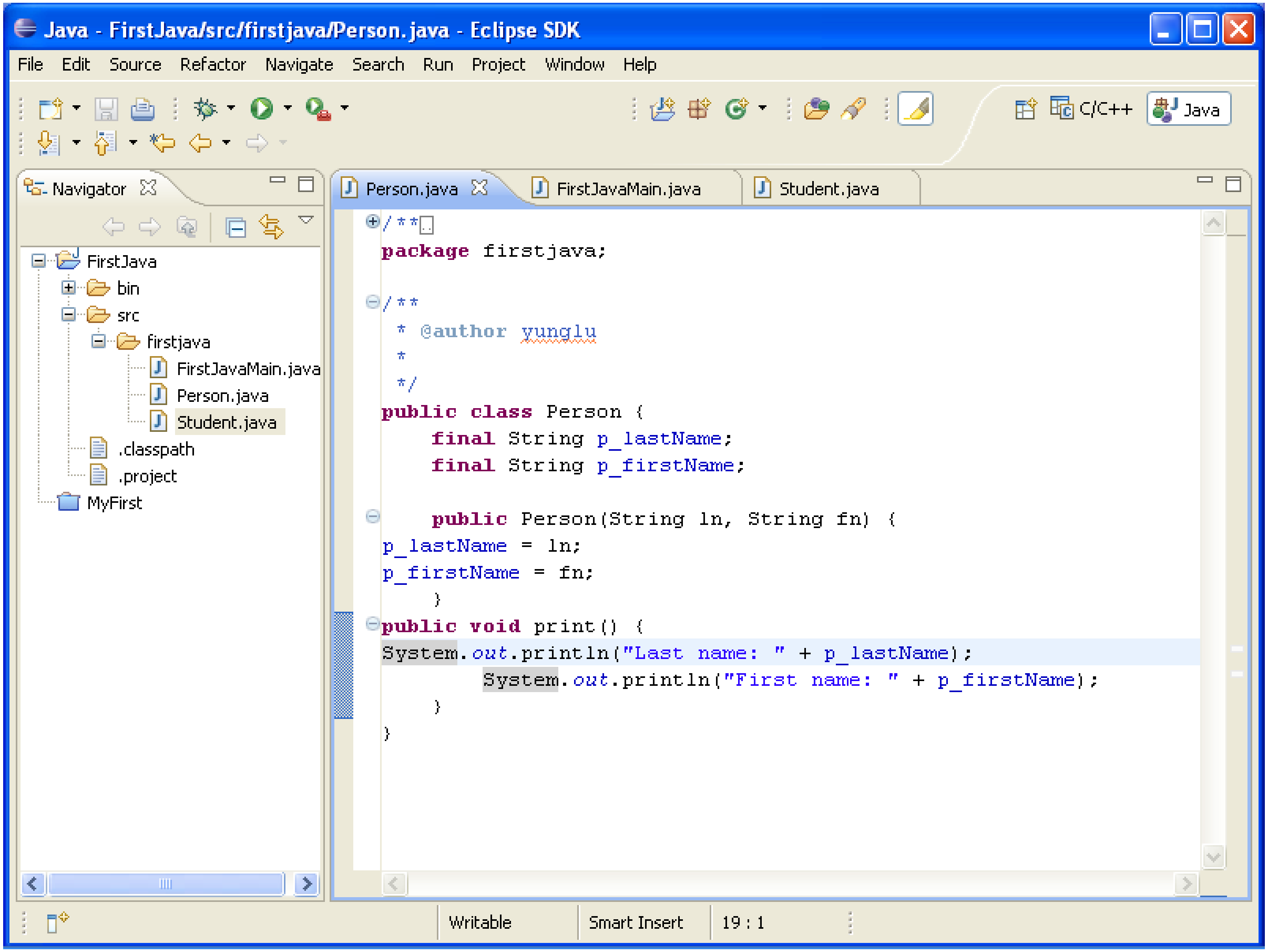
☑ public static void main(String[] args)

□ Constructors from superclass

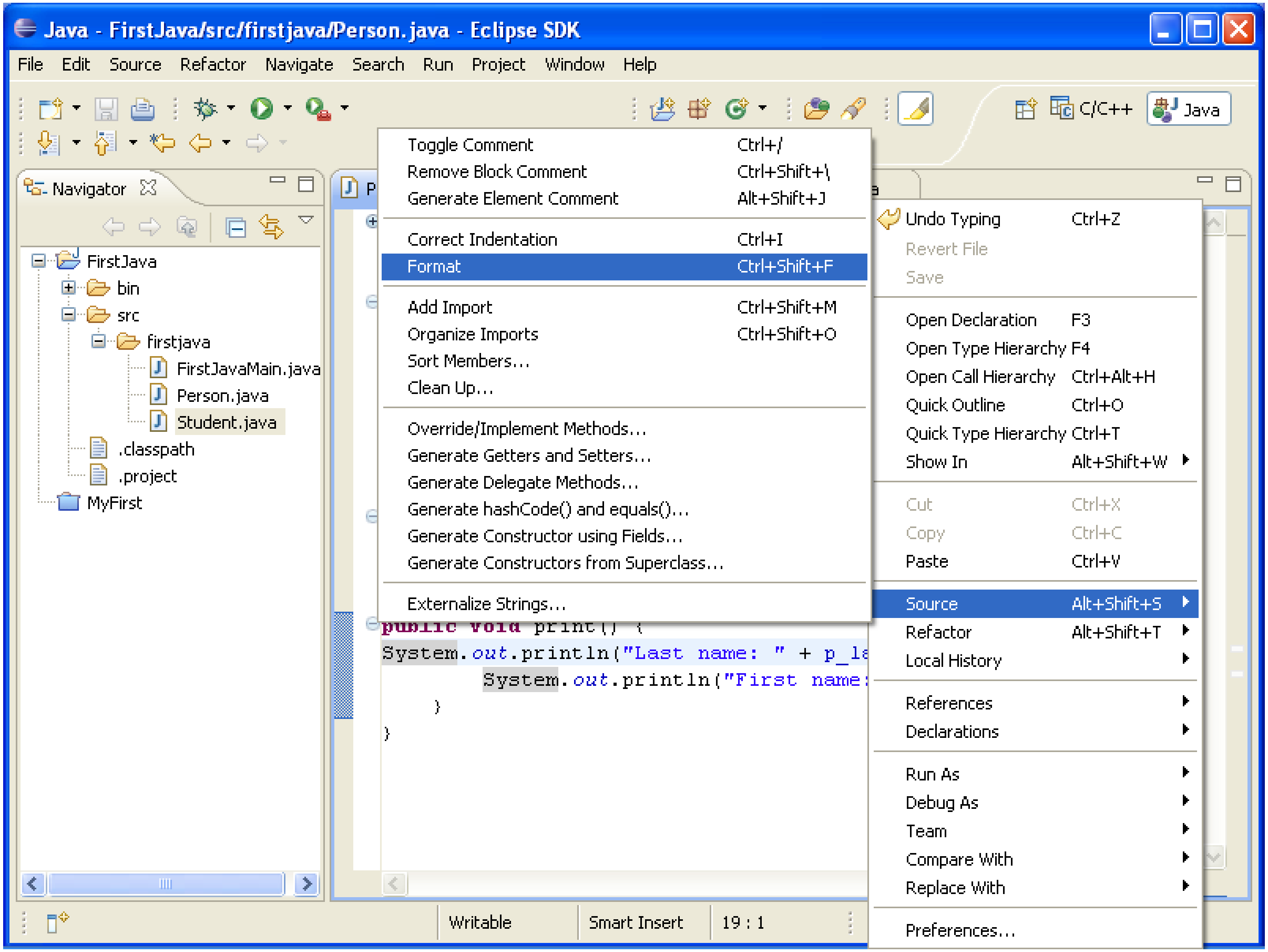
□ Inherited abstract methods

Do you want to add comments as configured in the properties of the current project?

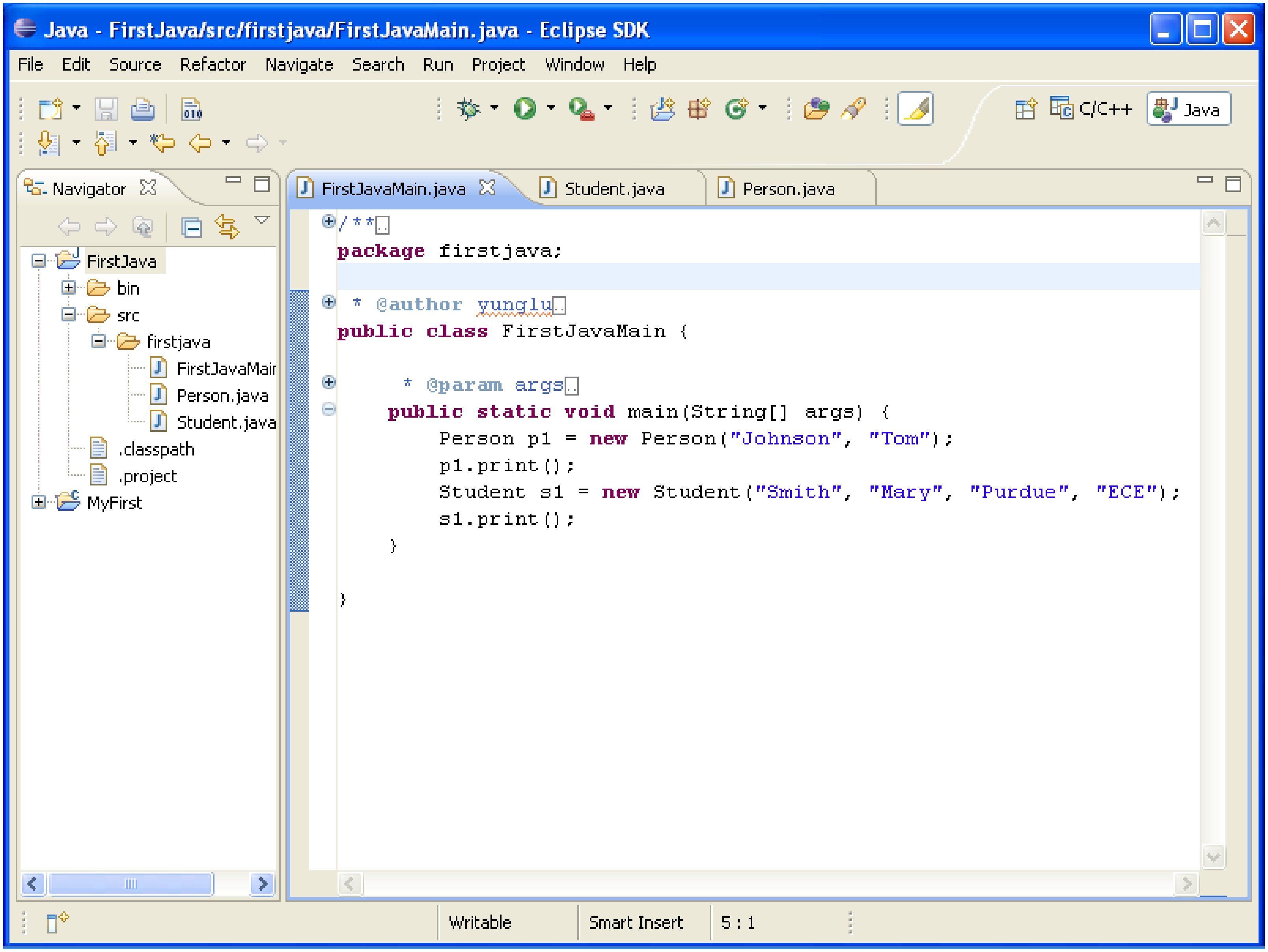
☑ Generate comments



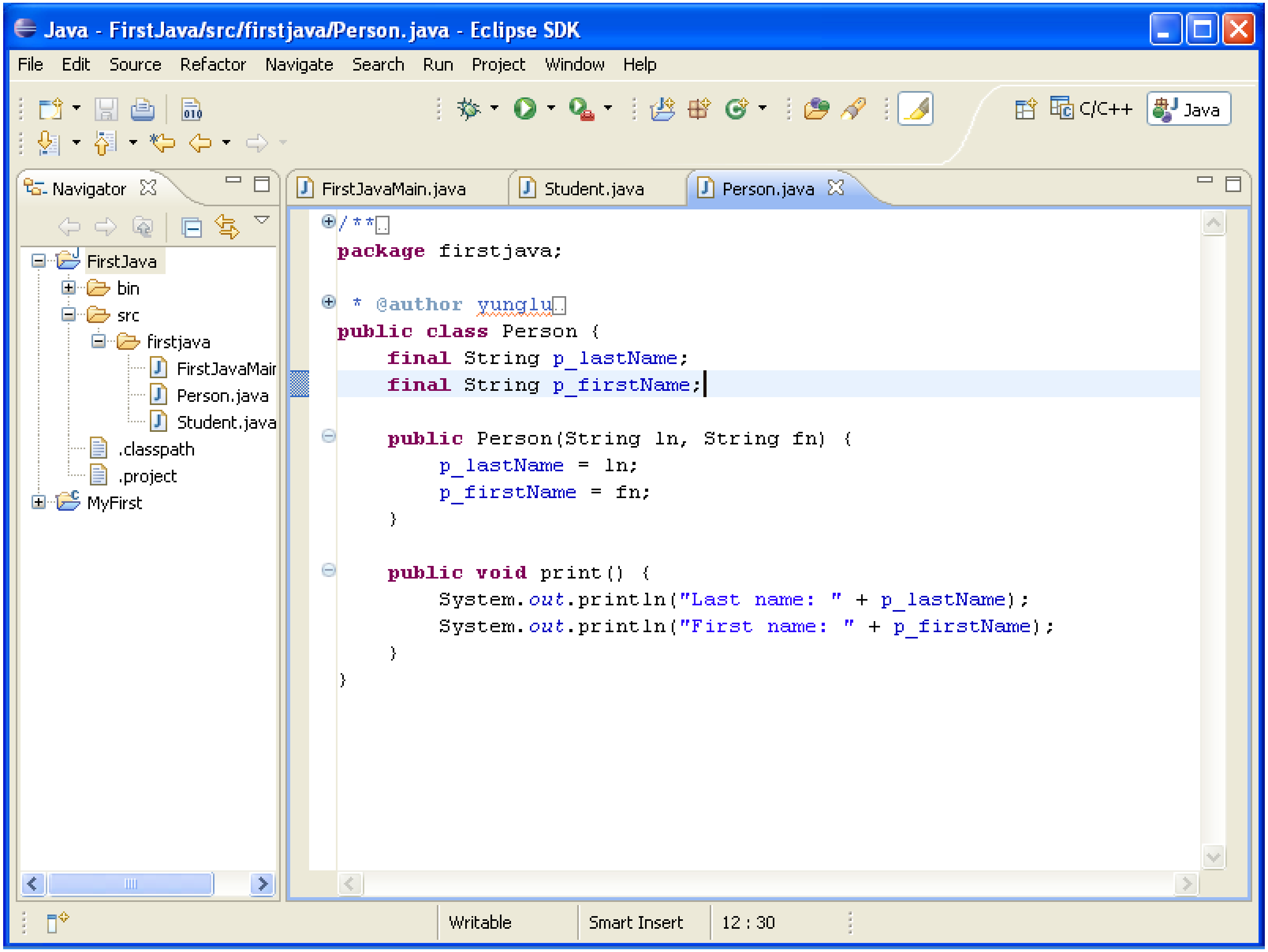
```
/**□
package firstjava;

/**
 * @author yunglu
 *
 */
public class Person {
    final String p_lastName;
    final String p_firstName;

    public Person(String ln, String fn) {
p_lastName = ln;
p_firstName = fn;
    }
public void print() {
System.out.println("Last name: " + p_lastName);
        System.out.println("First name: " + p_firstName);
    }
}
```



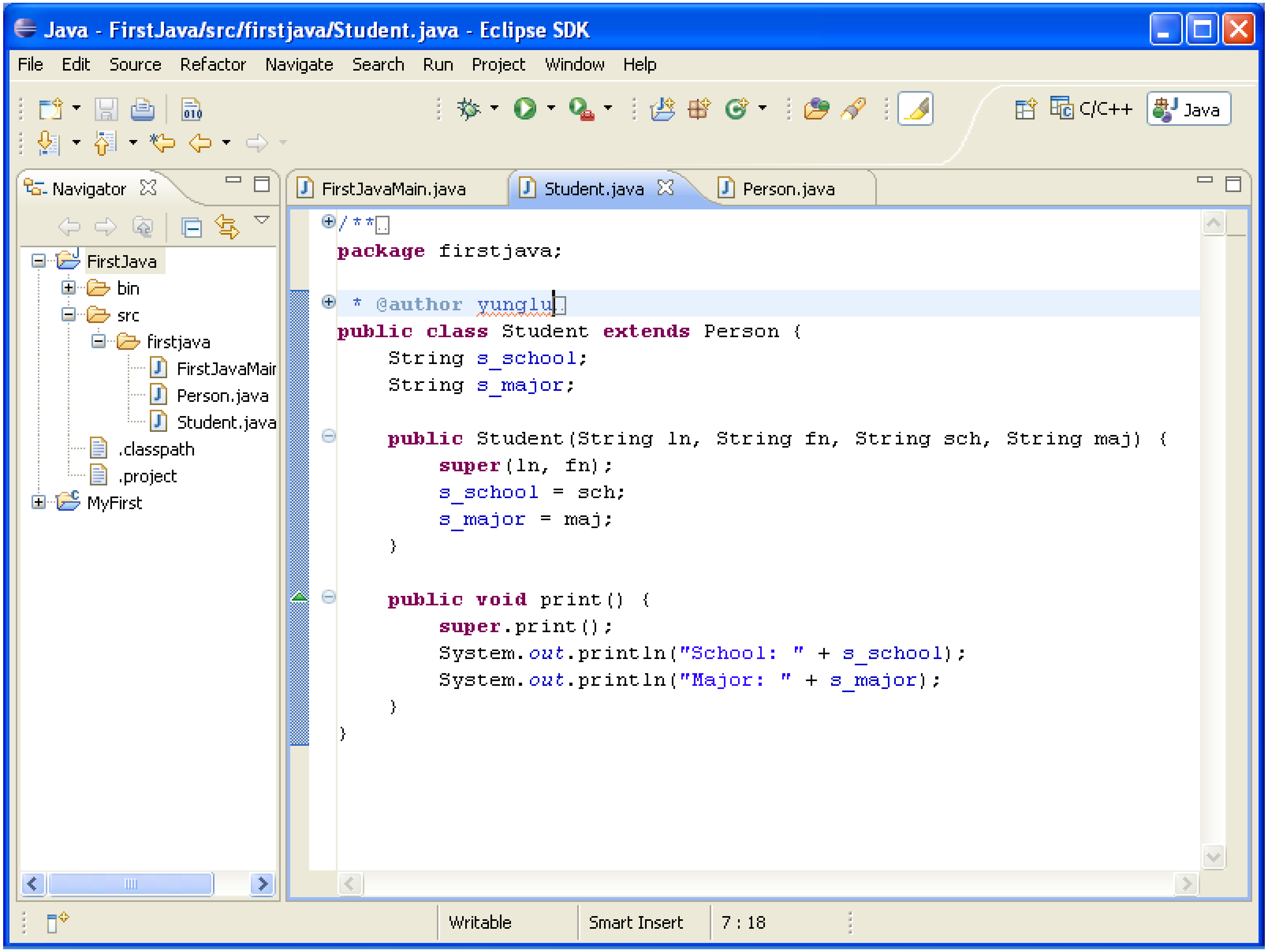
```
public void print() {
System.out.println("Last name: " + p_la
        System.out.println("First name:
    }
}
```



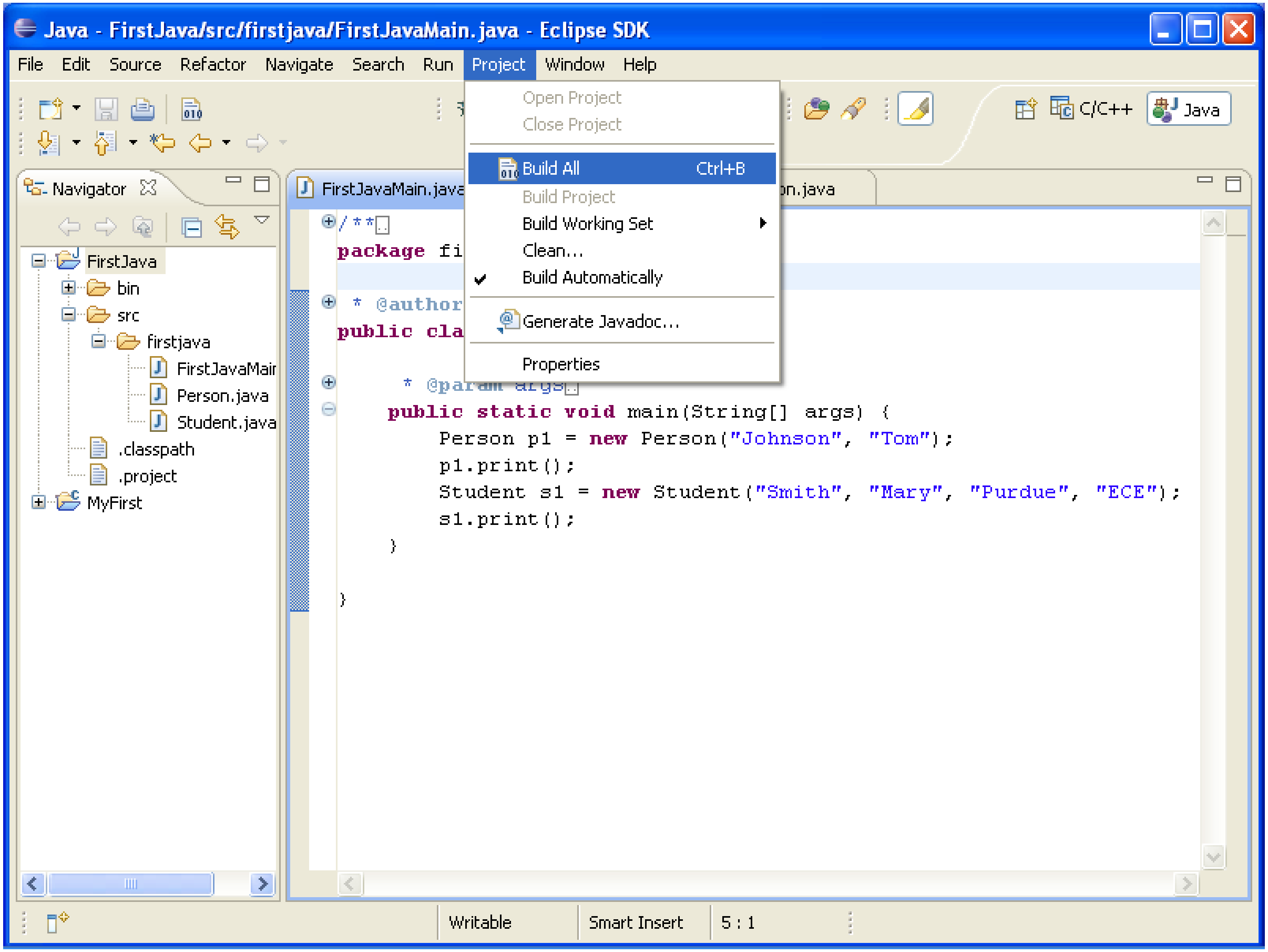
```
/**
package firstjava;

 * @author yunglu
public class FirstJavaMain {

	 * @param args
	public static void main(String[] args) {
		Person p1 = new Person("Johnson", "Tom");
		p1.print();
		Student s1 = new Student("Smith", "Mary", "Purdue", "ECE");
		s1.print();
	}

}
```



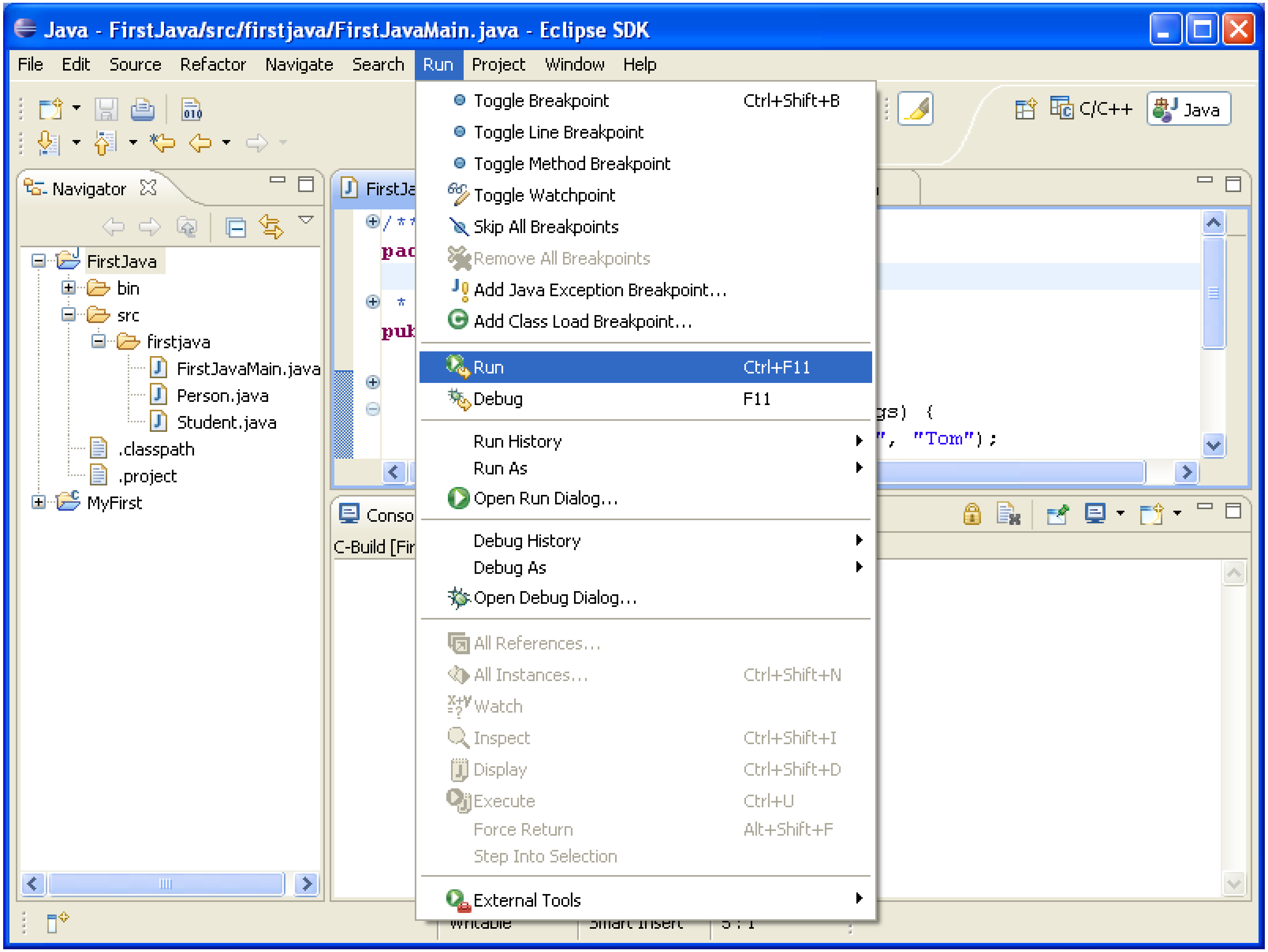
```
/**
package firstjava;

 * @author yunglu
public class Person {
     final String p_lastName;
     final String p_firstName;

     public Person(String ln, String fn) {
          p_lastName = ln;
          p_firstName = fn;
     }

     public void print() {
          System.out.println("Last name: " + p_lastName);
          System.out.println("First name: " + p_firstName);
     }
}
```

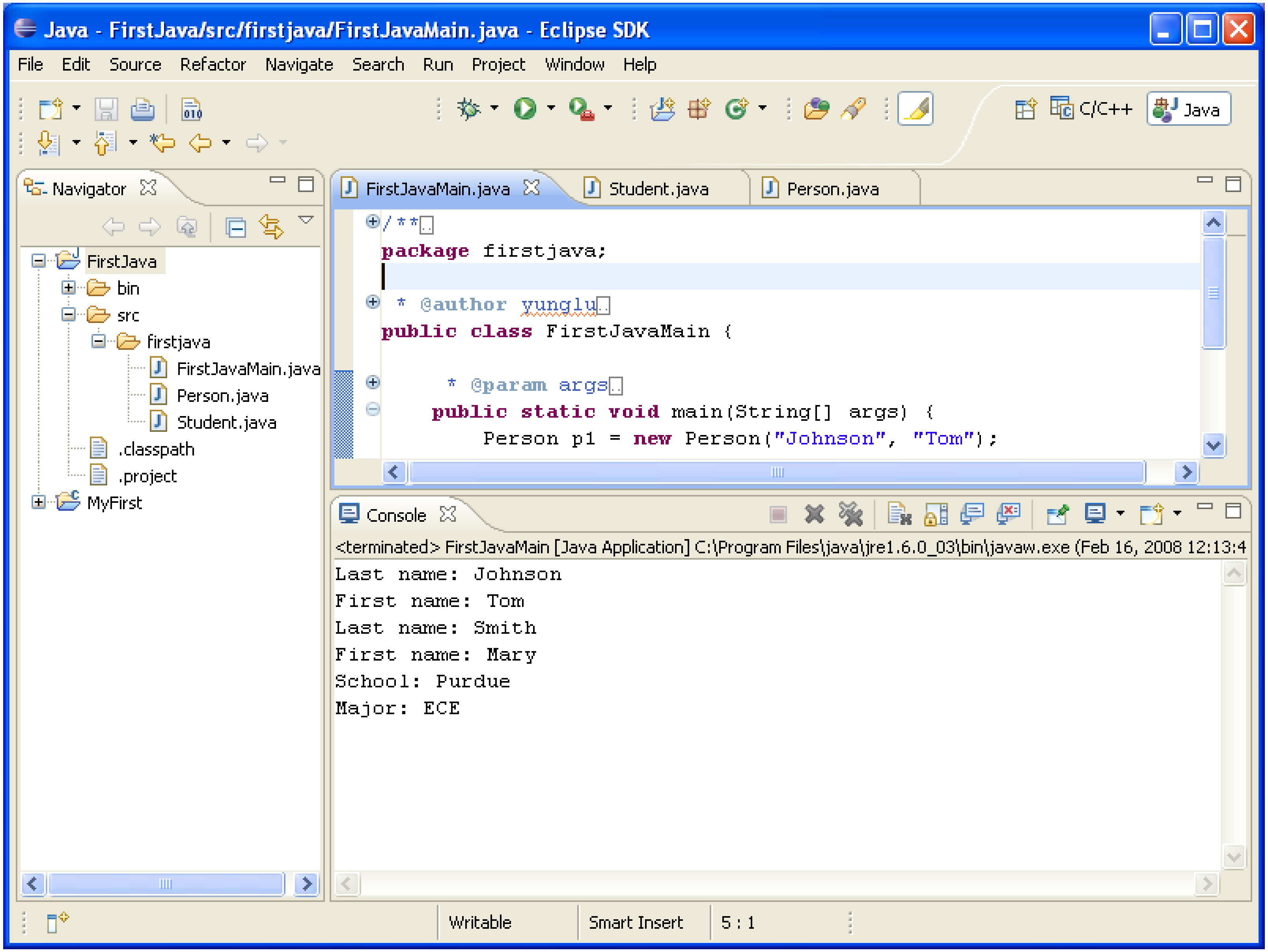


```
/**
package firstjava;

 * @author yunglu
public class Student extends Person {
    String s_school;
    String s_major;

    public Student(String ln, String fn, String sch, String maj) {
        super(ln, fn);
        s_school = sch;
        s_major = maj;
    }

    public void print() {
        super.print();
        System.out.println("School: " + s_school);
        System.out.println("Major: " + s_major);
    }
}
```



```
/**
package fi

 * @author
public cla

     * @param args
    public static void main(String[] args) {
        Person p1 = new Person("Johnson", "Tom");
        p1.print();
        Student s1 = new Student("Smith", "Mary", "Purdue", "ECE");
        s1.print();
    }

}
```

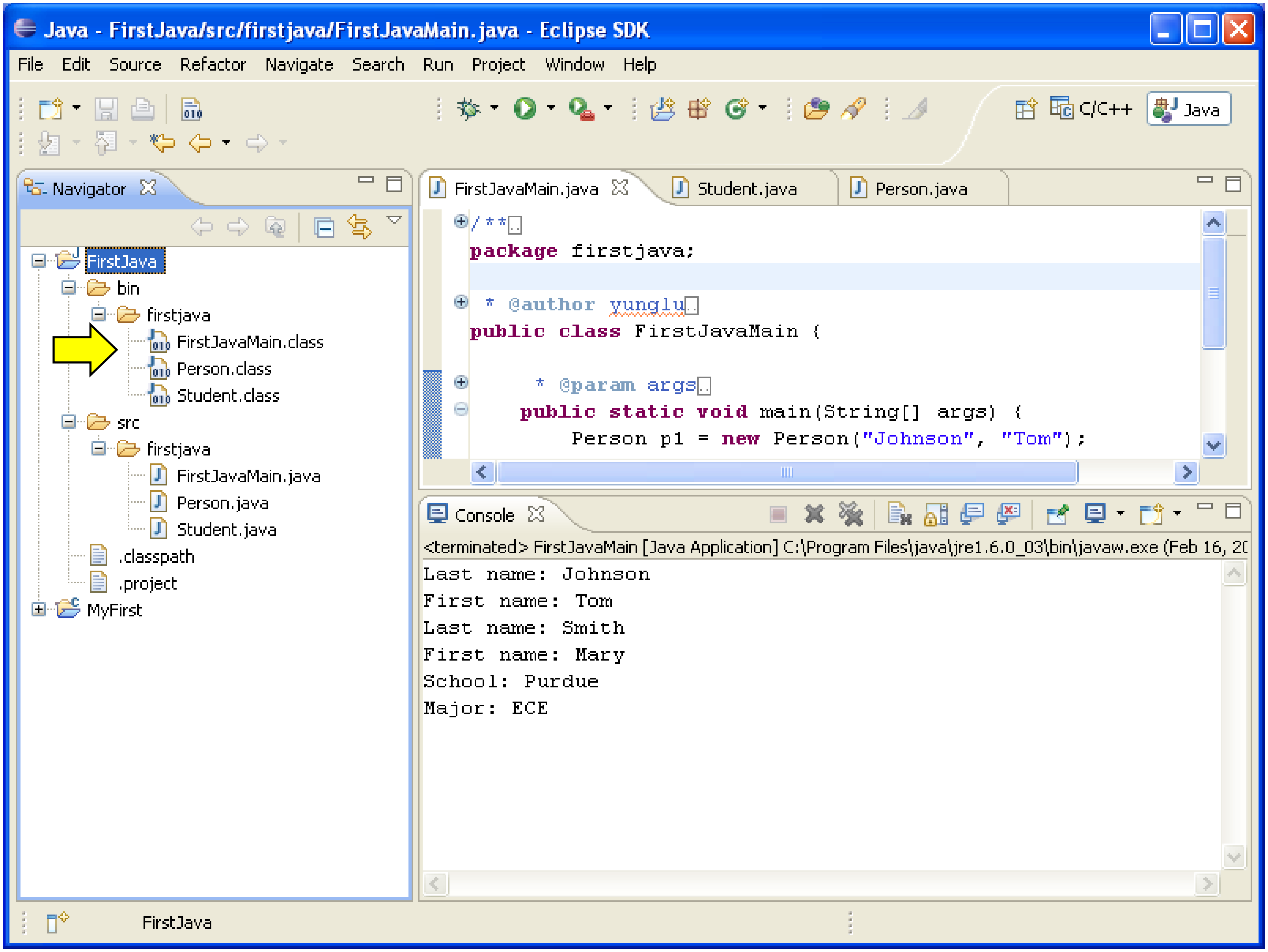

Java - FirstJava/src/firstjava/FirstJavaMain.java - Eclipse SDK
File Edit Source Refactor Navigate Search Run Project Window Help
C/C++
Java
Navigator
FirstJava
bin
src
firstjava
FirstJavaMain.java
Person.java
Student.java
.classpath
.project
MyFirst
Toggle Breakpoint Ctrl+Shift+B
Toggle Line Breakpoint
Toggle Method Breakpoint
Toggle Watchpoint
Skip All Breakpoints
Remove All Breakpoints
Add Java Exception Breakpoint...
Add Class Load Breakpoint...
Run Ctrl+F11
Debug F11
Run History
Run As
Open Run Dialog...
Debug History
Debug As
Open Debug Dialog...
All References...
All Instances... Ctrl+Shift+N
Watch
Inspect Ctrl+Shift+I
Display Ctrl+Shift+D
Execute Ctrl+U
Force Return Alt+Shift+F
Step Into Selection
External Tools
Conso
C-Build [Fir
gs) {
", "Tom");
Writable
Smart Insert
5 : 1



```
/**
package firstjava;

 * @author yunglu
public class FirstJavaMain {

     * @param args
    public static void main(String[] args) {
        Person p1 = new Person("Johnson", "Tom");
```

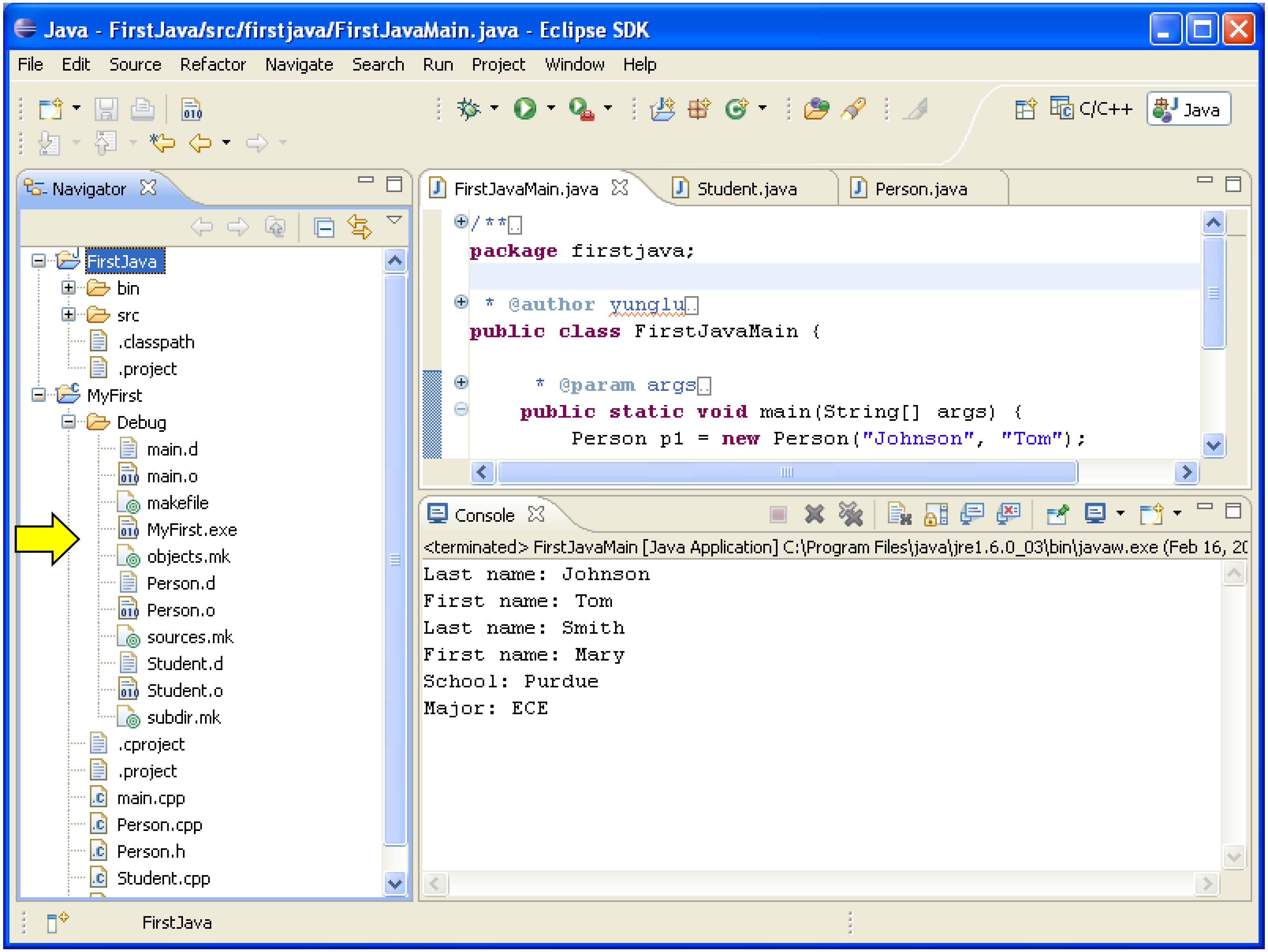
<terminated> FirstJavaMain [Java Application] C:\Program Files\java\jre1.6.0_03\bin\javaw.exe (Feb 16, 2008 12:13:4

```
Last name: Johnson
First name: Tom
Last name: Smith
First name: Mary
School: Purdue
Major: ECE
```



```
/**
package firstjava;

 * @author yunglu
public class FirstJavaMain {

     * @param args
    public static void main(String[] args) {
        Person p1 = new Person("Johnson", "Tom");
```

```
<terminated> FirstJavaMain [Java Application] C:\Program Files\java\jre1.6.0_03\bin\javaw.exe (Feb 16, 2(
Last name: Johnson
First name: Tom
Last name: Smith
First name: Mary
School: Purdue
Major: ECE
```



```
/**
package firstjava;

 * @author yunglu
public class FirstJavaMain {

     * @param args
    public static void main(String[] args) {
        Person p1 = new Person("Johnson", "Tom");
```

<terminated> FirstJavaMain [Java Application] C:\Program Files\java\jre1.6.0_03\bin\javaw.exe (Feb 16, 20

```
Last name: Johnson
First name: Tom
Last name: Smith
First name: Mary
School: Purdue
Major: ECE
```

C++ and Java Syntax

c++	Java
int main(int argc, char * argv[])	public static void main(String[] args) {
Person p1("Johnson", "Tom");	Person p1 = new Person("Johnson", "Tom");
p1.print();	p1.print();

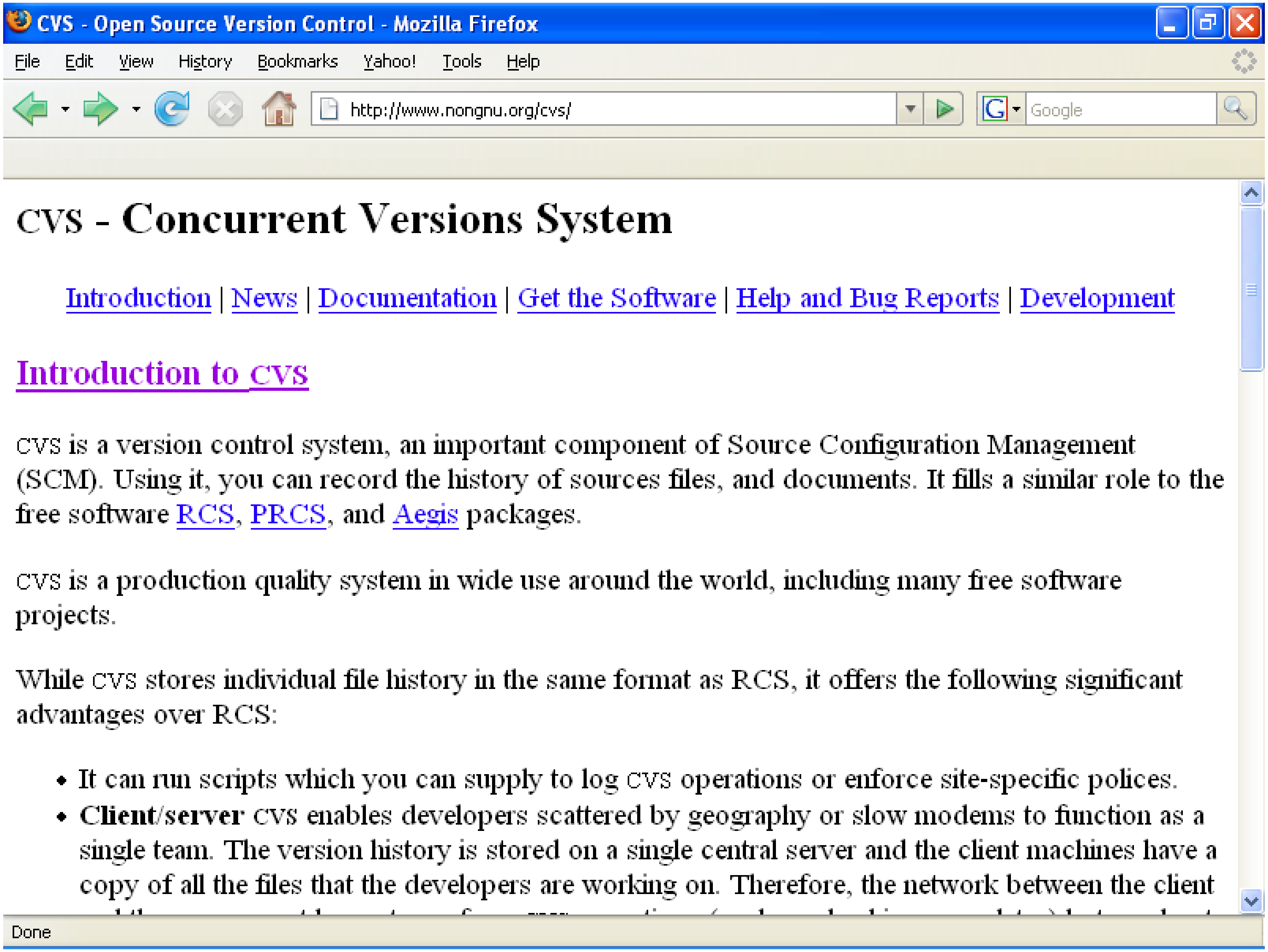
class Person
{
public:
Person(string ln, string fn);
	public class Person {
	public Person(String ln, String fn) {

const string p_lastName;	final String p_lastName;
class Student: public Person	class Student extends Person

Version Control

Every assignment and every lab exercise **must** be submitted using the **CVS repository**.

Submission of the source code only will **not be graded**.


CVS - Open Source Version Control - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://www.nongnu.org/cvs/
Google
Done

CVS - Concurrent Versions System

Introduction | News | Documentation | Get the Software | Help and Bug Reports | Development

Introduction to CVS

CVS is a version control system, an important component of Source Configuration Management (SCM). Using it, you can record the history of sources files, and documents. It fills a similar role to the free software RCS, PRCS, and Aegis packages.

CVS is a production quality system in wide use around the world, including many free software projects.

While CVS stores individual file history in the same format as RCS, it offers the following significant advantages over RCS:

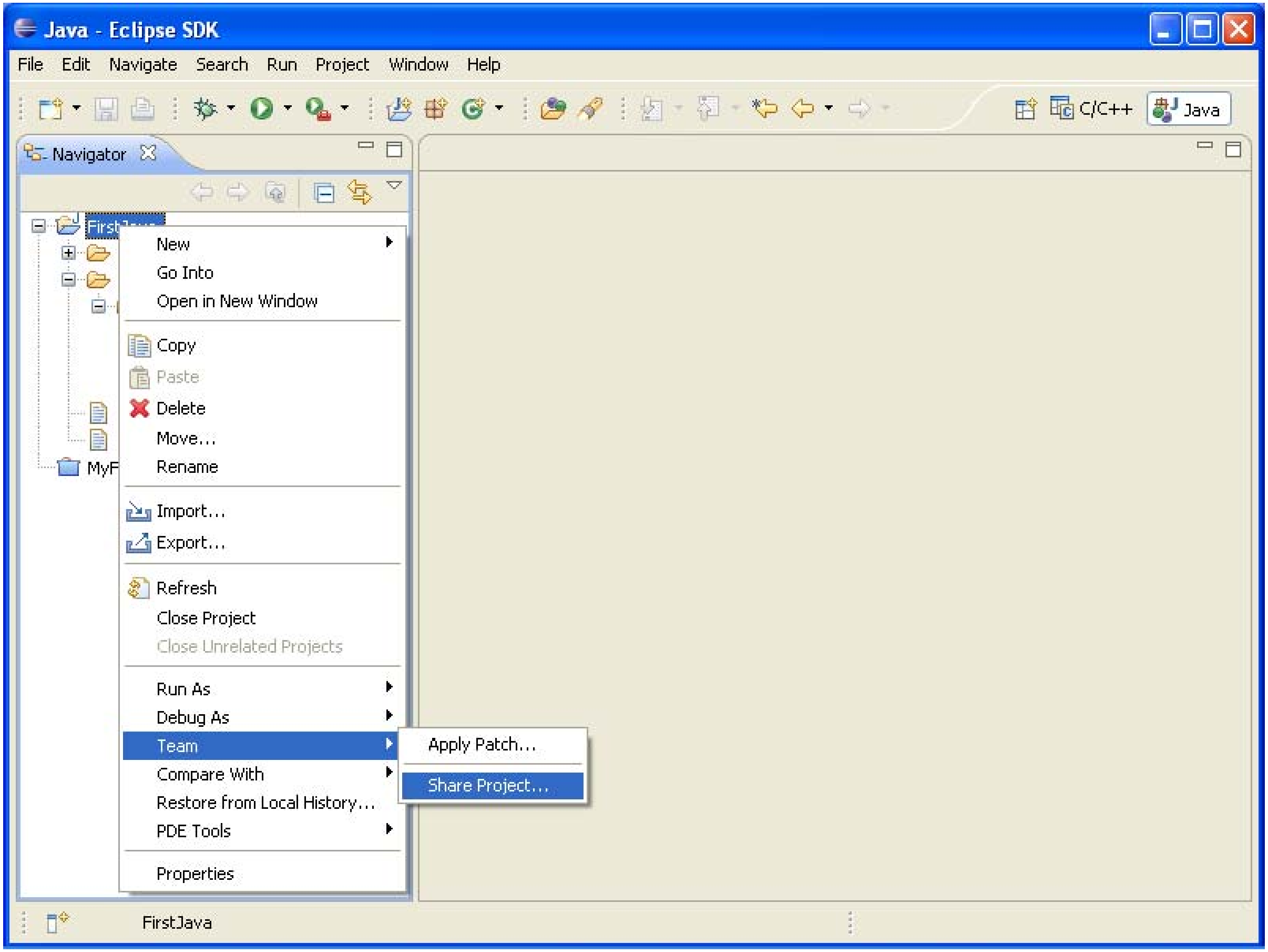
- It can run scripts which you can supply to log CVS operations or enforce site-specific polices.
- **Client/server** CVS enables developers scattered by geography or slow modems to function as a single team. The version history is stored on a single central server and the client machines have a copy of all the files that the developers are working on. Therefore, the network between the client

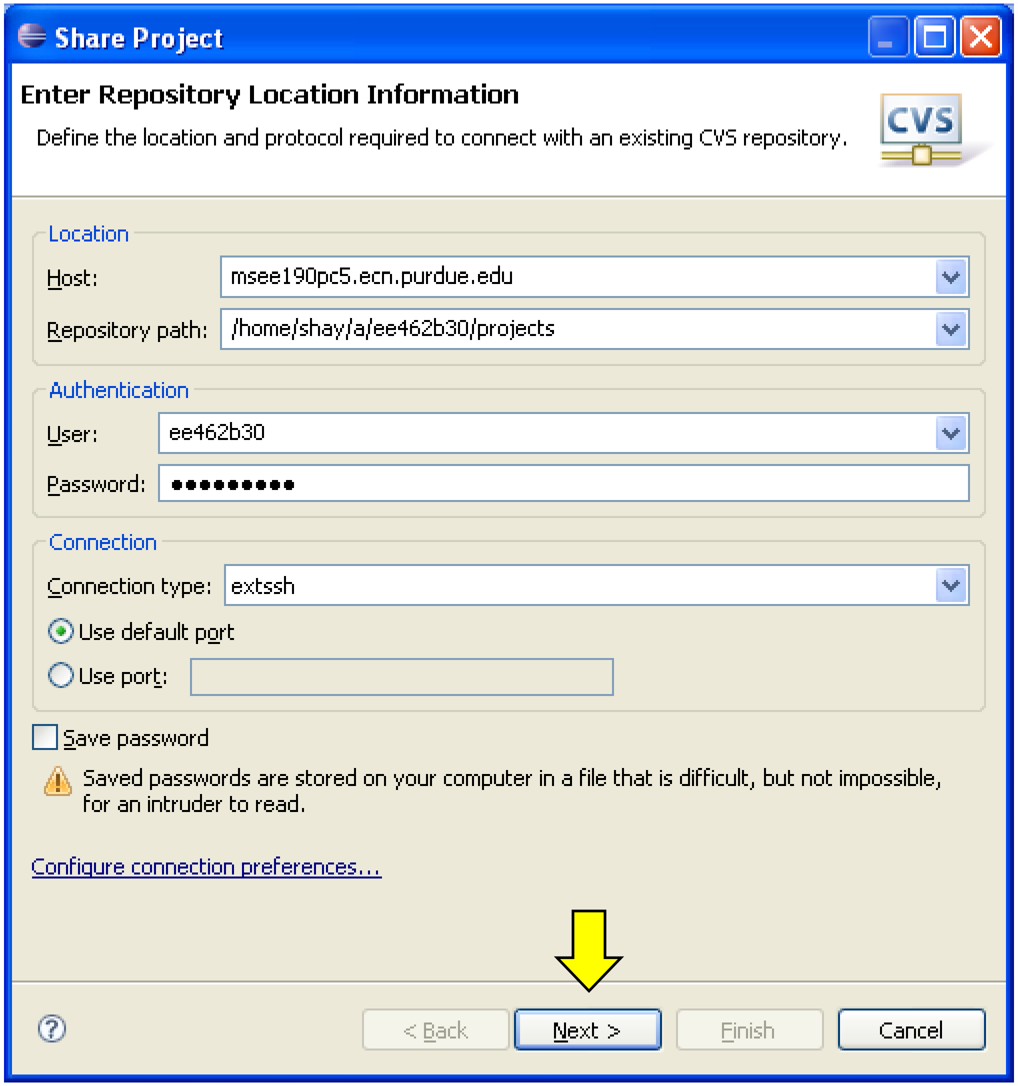
Update and Commit

- update = pull the changes made by your teammates from the repository
- commit = push your changes to the repository so that your teammates can see
- update and commit **often** ⇒ keep your files and the repository "in sync"
- focus on one task at a time, finish it and commit it
- commit after adding a feature or fixing a bug
- use meaningful comments to indicate the progression of the project

Prepare a Repository

- enter your 462 account
- make a directory called "projects"
- enter the "projects" directory
- make a directory called "CVSROOT"

Java - Eclipse SDK
File Edit Navigate Search Run Project Window Help
C/C++
Java
Navigator
FirstJava
MyF
New
Go Into
Open in New Window
Copy
Paste
Delete
Move...
Rename
Import...
Export...
Refresh
Close Project
Close Unrelated Projects
Run As
Debug As
Team
Compare With
Restore from Local History...
PDE Tools
Properties
Apply Patch...
Share Project...
FirstJava



Enter Repository Location Information

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

Location

Host: msee190pc5.ecn.purdue.edu

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

Authentication

User: ee462b30

Password: ●●●●●●●●●●

Connection

Connection type: extssh

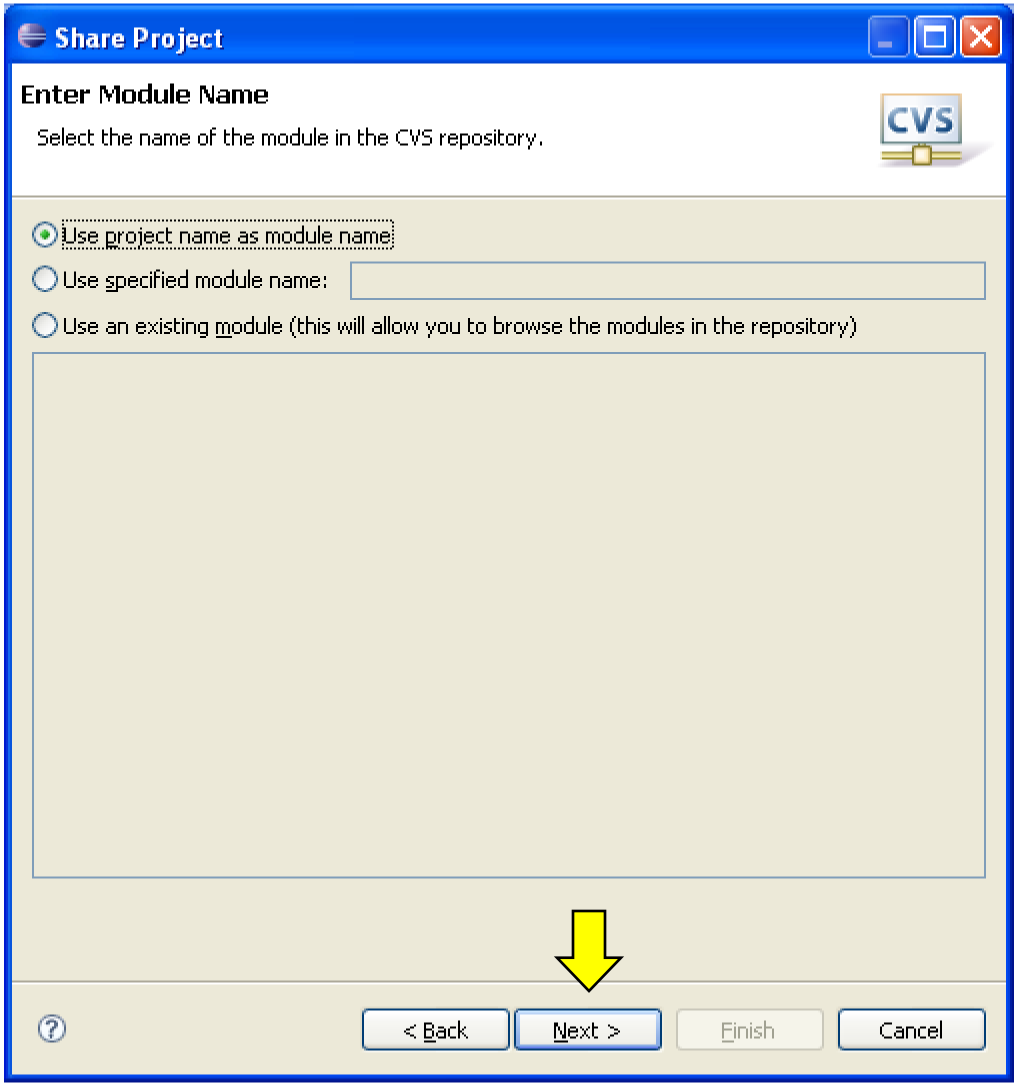
● Use default port

○ 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...



Share Project

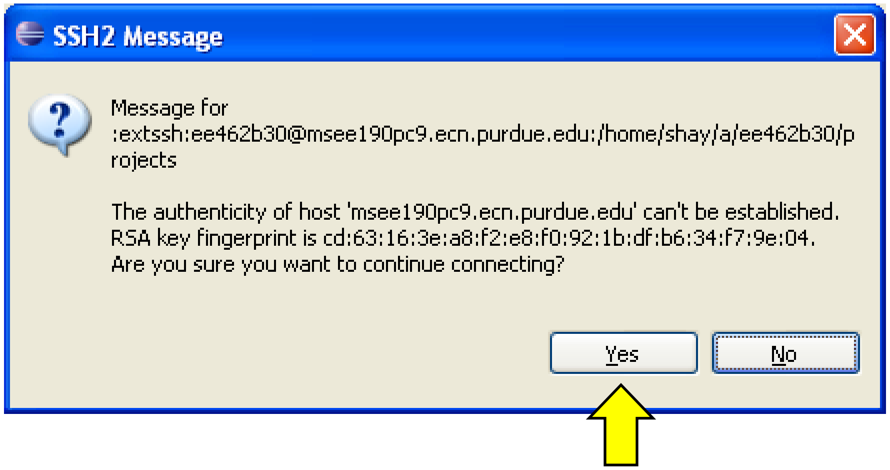
Enter Module Name

Select the name of the module in the CVS repository.

● Use project name as module name

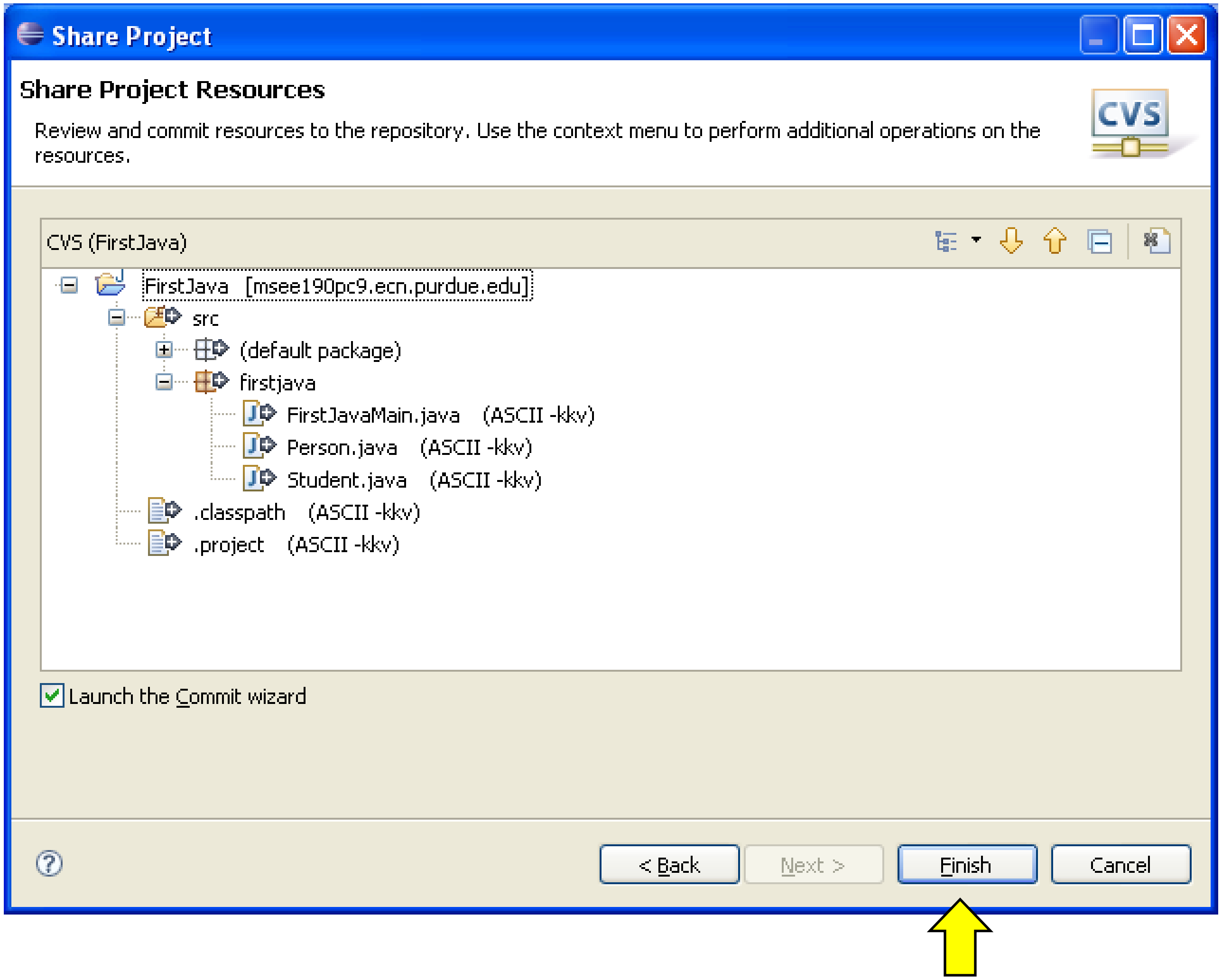
○ Use specified module name:

○ Use an existing module (this will allow you to browse the modules in the repository)



Message for
:extssh:ee462b30@msee190pc9.ecn.purdue.edu:/home/shay/a/ee462b30/projects

The authenticity of host 'msee190pc9.ecn.purdue.edu' can't be established.
RSA key fingerprint is cd:63:16:3e:a8:f2:e8:f0:92:1b:df:b6:34:f7:9e:04.
Are you sure you want to continue connecting?



Share Project

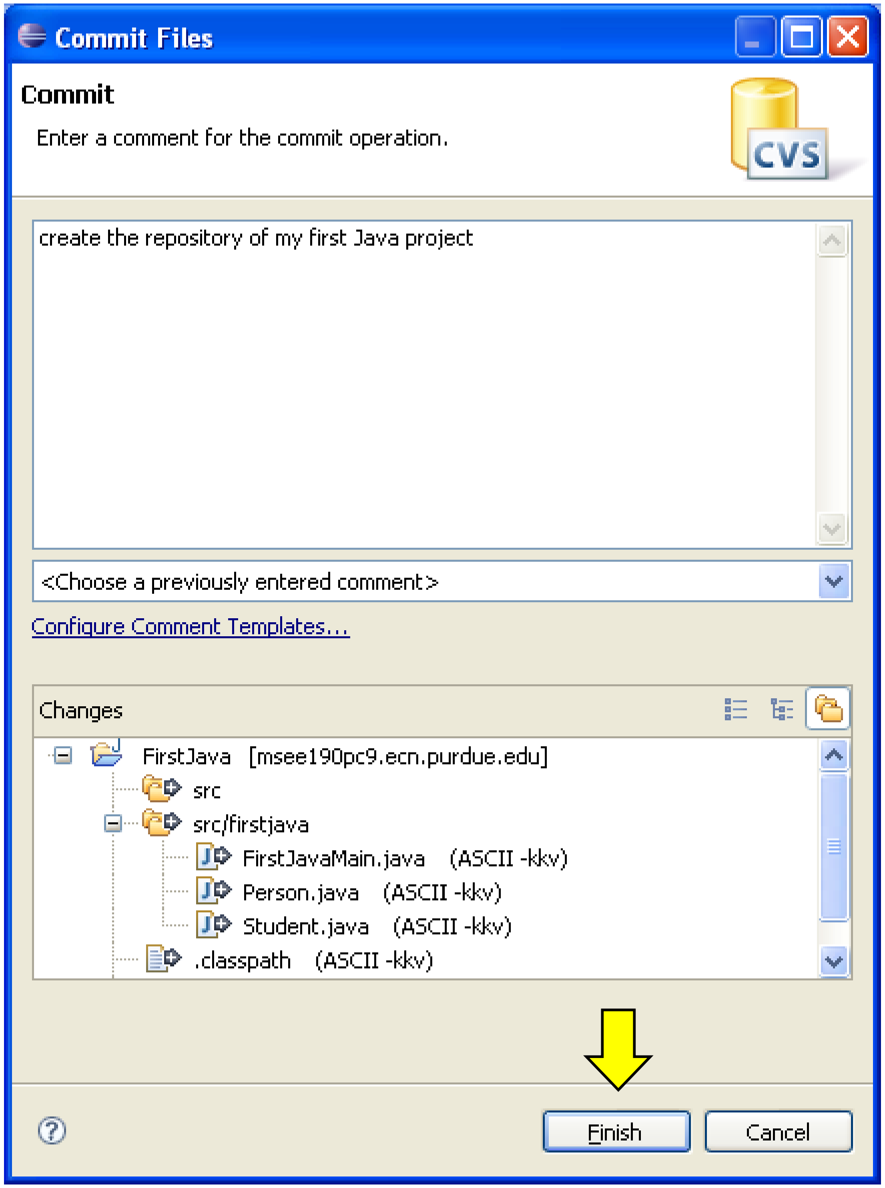
Share Project Resources

Review and commit resources to the repository. Use the context menu to perform additional operations on the resources.

CVS (FirstJava)

```
FirstJava  [msee190pc9.ecn.purdue.edu]
  src
    (default package)
    firstjava
      FirstJavaMain.java   (ASCII -kkv)
      Person.java   (ASCII -kkv)
      Student.java   (ASCII -kkv)
  .classpath   (ASCII -kkv)
  .project   (ASCII -kkv)
```

☑ Launch the Commit wizard



Commit

Enter a comment for the commit operation.

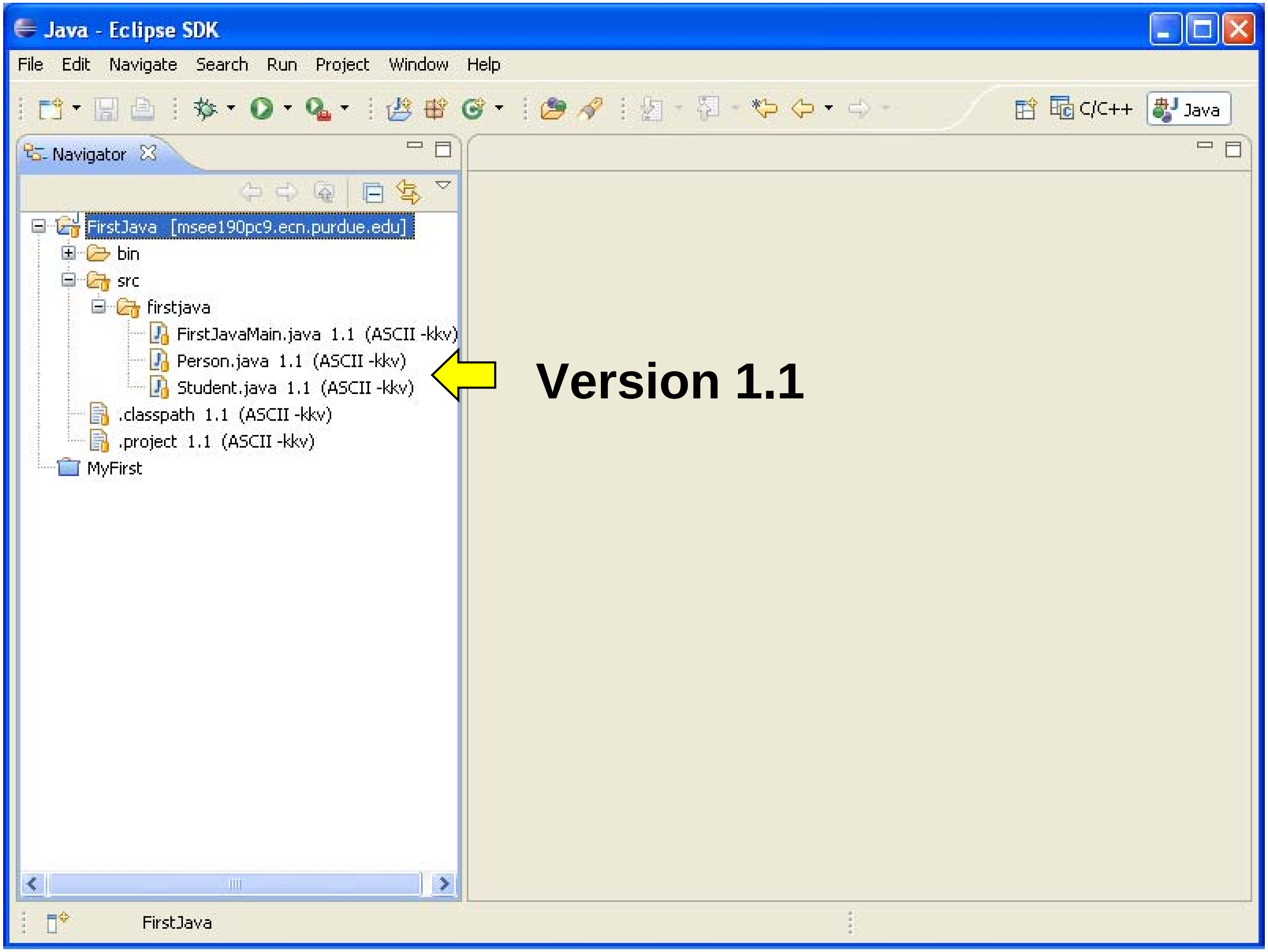
create the repository of my first Java project

<Choose a previously entered comment>

Configure Comment Templates...

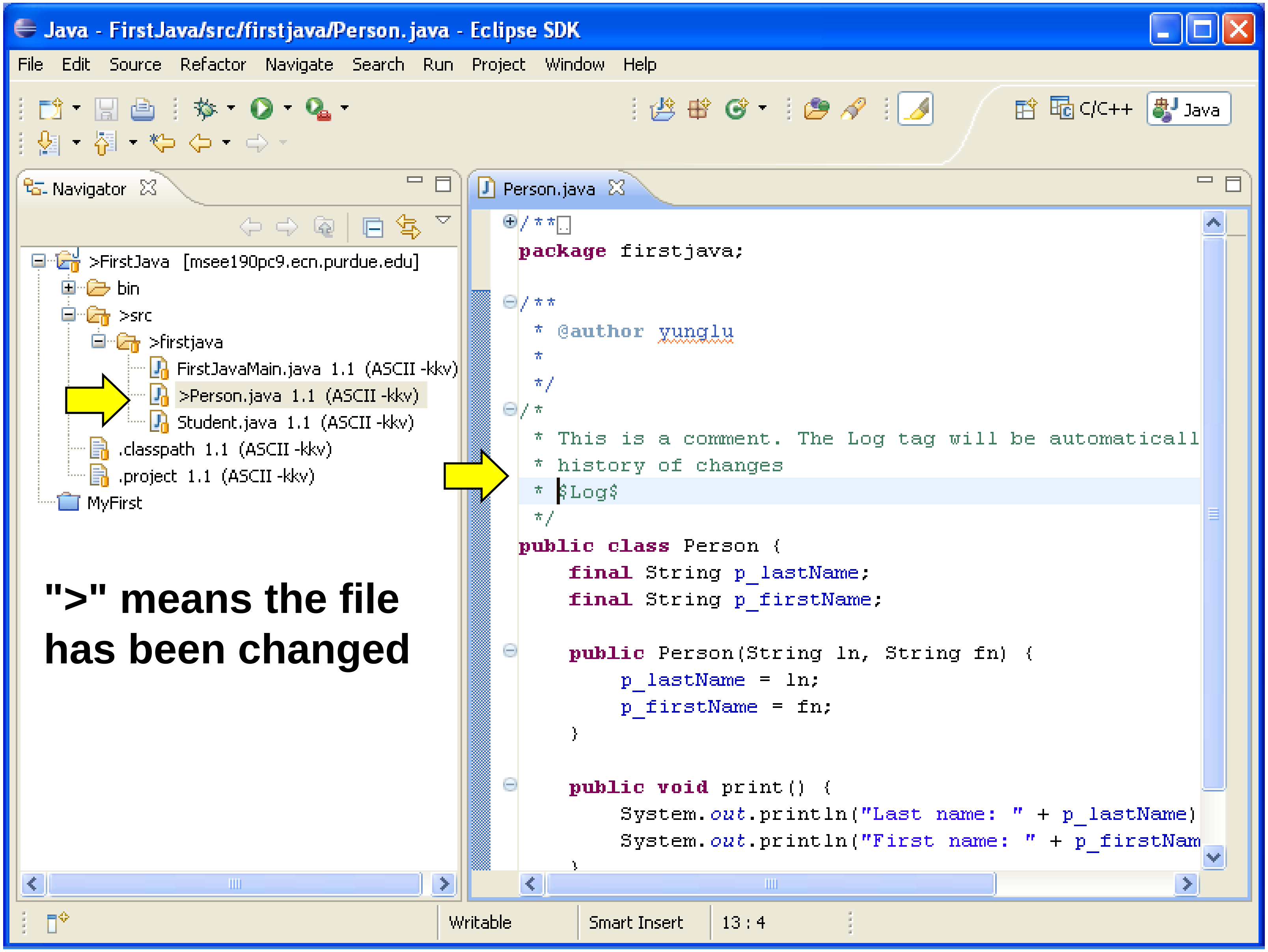
Changes

```
FirstJava  [msee190pc9.ecn.purdue.edu]
    src
    src/firstjava
        FirstJavaMain.java  (ASCII -kkv)
        Person.java  (ASCII -kkv)
        Student.java  (ASCII -kkv)
    .classpath  (ASCII -kkv)
```



Java - Eclipse SDK
File Edit Navigate Search Run Project Window Help
C/C++
Java
Navigator
FirstJava [msee190pc9.ecn.purdue.edu]
bin
src
firstjava
FirstJavaMain.java 1.1 (ASCII -kkv)
Person.java 1.1 (ASCII -kkv)
Student.java 1.1 (ASCII -kkv)
.classpath 1.1 (ASCII -kkv)
.project 1.1 (ASCII -kkv)
MyFirst
FirstJava

Version 1.1

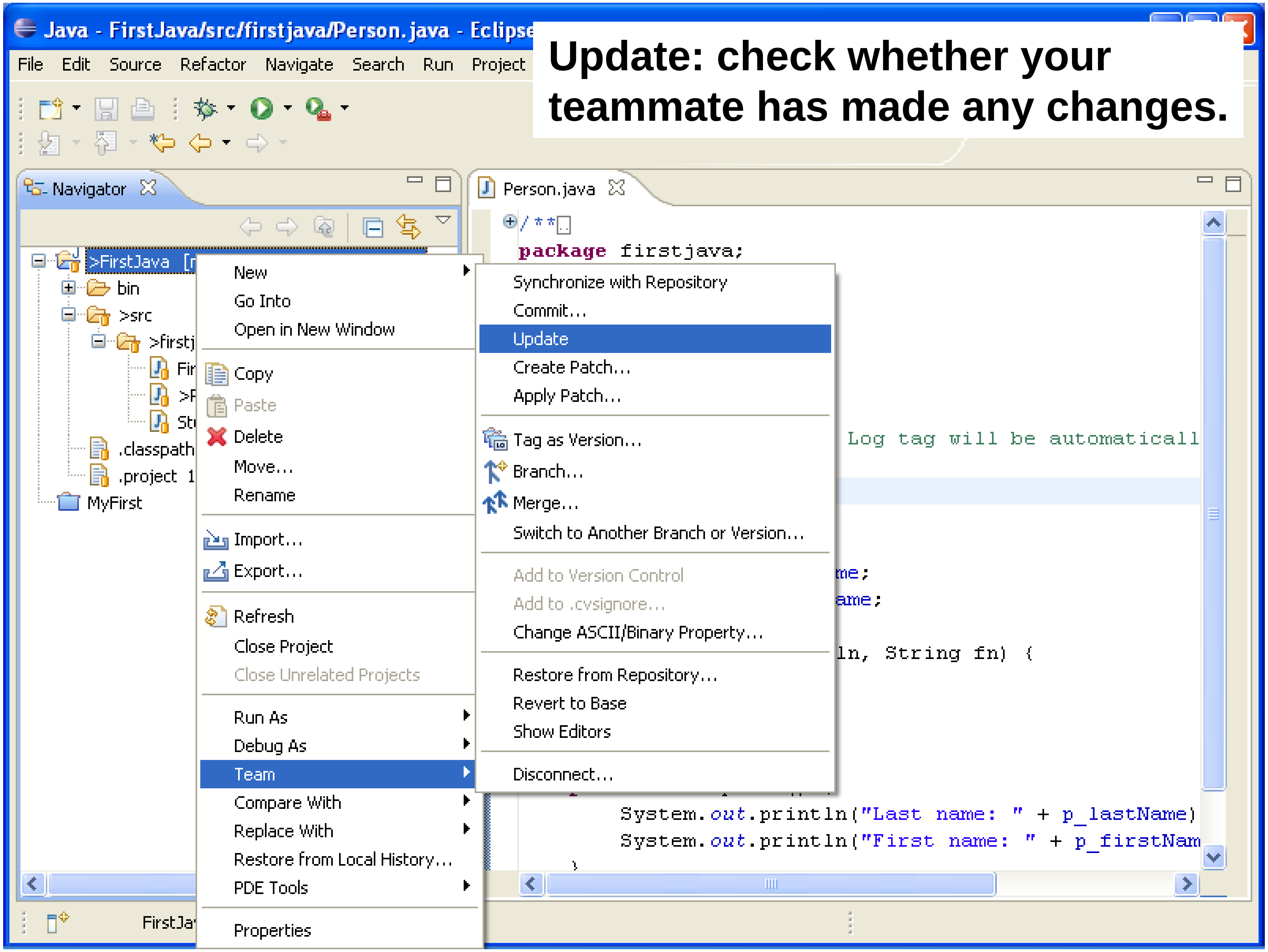


Java - FirstJava/src/firstjava/Person.java - Eclipse SDK
File Edit Source Refactor Navigate Search Run Project Window Help
C/C++
Java
Navigator
Person.java
Writable
Smart Insert
13 : 4

```
>FirstJava  [msee190pc9.ecn.purdue.edu]
    bin
    >src
        >firstjava
            FirstJavaMain.java  1.1  (ASCII -kkv)
            >Person.java  1.1  (ASCII -kkv)
            Student.java  1.1  (ASCII -kkv)
    .classpath  1.1  (ASCII -kkv)
    .project  1.1  (ASCII -kkv)
MyFirst
```

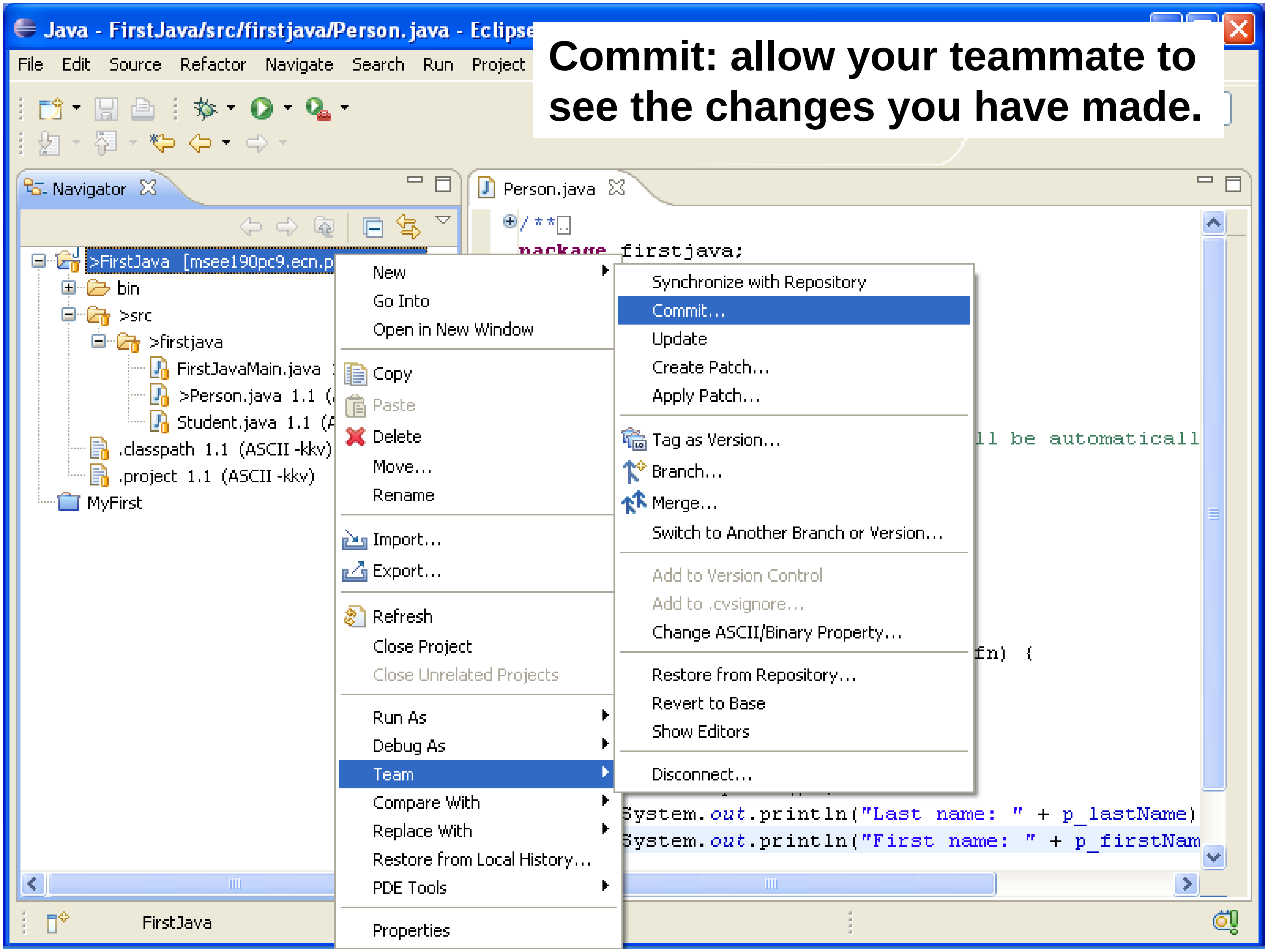
```
/**□
package firstjava;

/**
 * @author yunglu
 *
 */
/*
 * This is a comment. The Log tag will be automaticall
 * history of changes
 * $Log$
 */
public class Person {
    final String p_lastName;
    final String p_firstName;

    public Person(String ln, String fn) {
        p_lastName = ln;
        p_firstName = fn;
    }

    public void print() {
        System.out.println("Last name: " + p_lastName)
        System.out.println("First name: " + p_firstNam
    }
```

">" means the file has been changed



```
/**
package firstjava;
Log tag will be automaticall
me;
ame;
ln, String fn) {
        System.out.println("Last name: " + p_lastName)
        System.out.println("First name: " + p_firstNam
```

Update: check whether your teammate has made any changes.



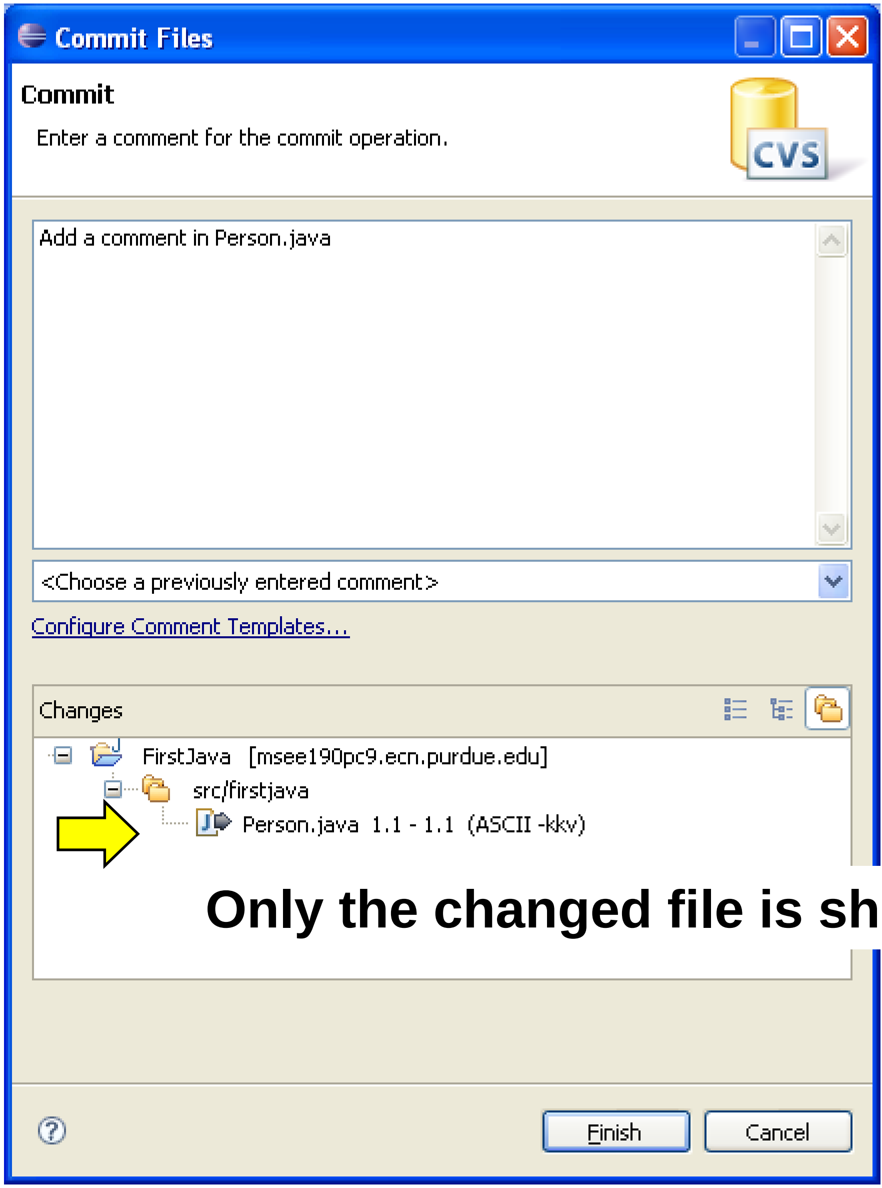
Commit: allow your teammate to see the changes you have made.

```
/**
package firstjava;
...ll be automaticall
...fn) {
System.out.println("Last name: " + p_lastName)
System.out.println("First name: " + p_firstNam
```

Always update before commit.

Otherwise, you **may overwrite** your teammate's changes.

It is recoverable but you should avoid such a problem.


Commit Files
Commit
Enter a comment for the commit operation.
CVS

Add a comment in Person.java

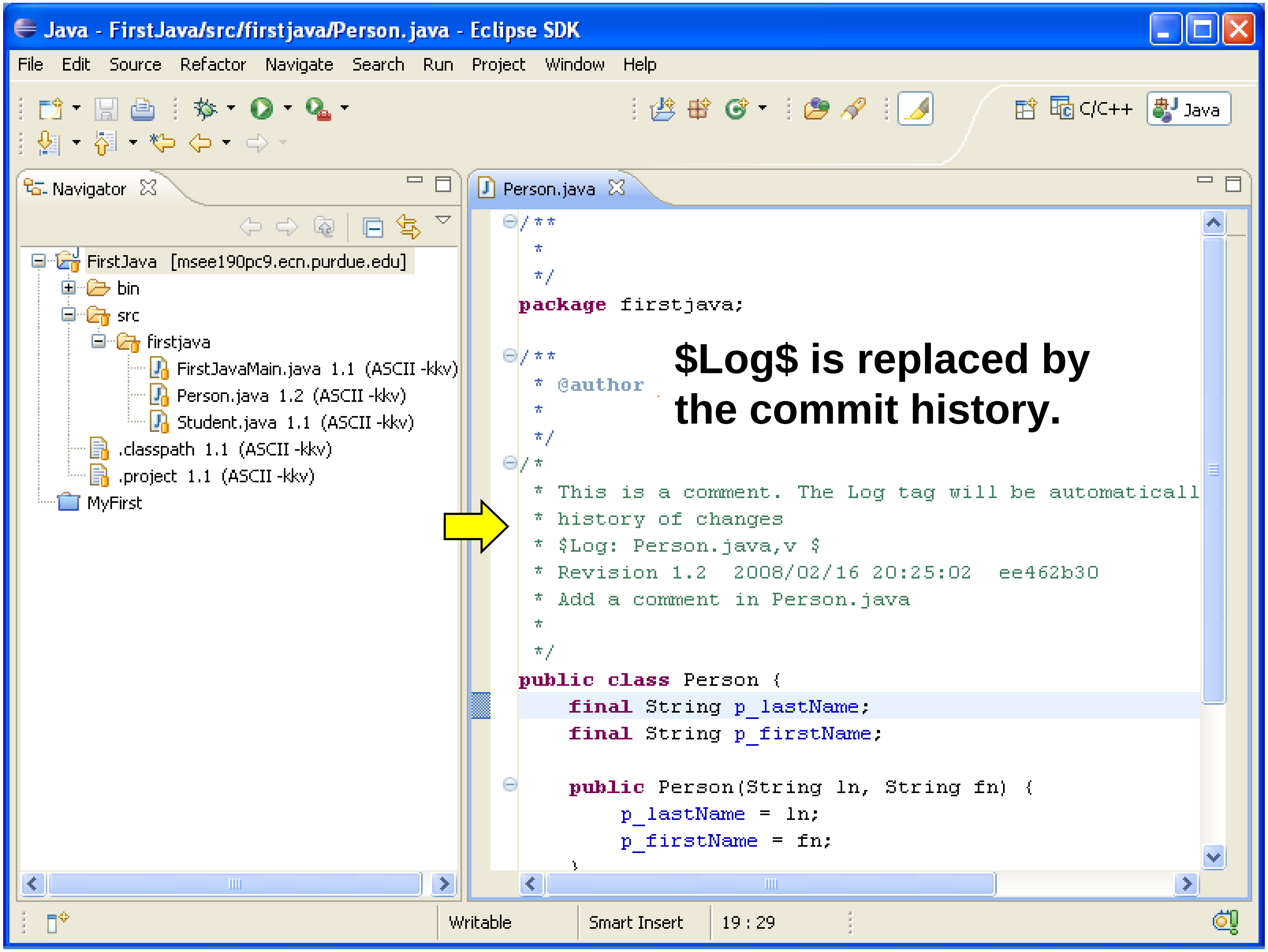
<Choose a previously entered comment>
Configure Comment Templates...
Changes

FirstJava [msee190pc9.ecn.purdue.edu]

src/firstjava

Person.java 1.1 - 1.1 (ASCII -kkv)

Only the changed file is shown.


Java - FirstJava/src/firstjava/Person.java - Eclipse SDK
File
Edit
Source
Refactor
Navigate
Search
Run
Project
Window
Help
C/C++
Java
Navigator
FirstJava [msee190pc9.ecn.purdue.edu]
bin
src
firstjava
FirstJavaMain.java 1.1 (ASCII -kkv)
Person.java 1.2 (ASCII -kkv)
Student.java 1.1 (ASCII -kkv)
.classpath 1.1 (ASCII -kkv)
.project 1.1 (ASCII -kkv)
MyFirst
Person.java
Writable
Smart Insert
19 : 29

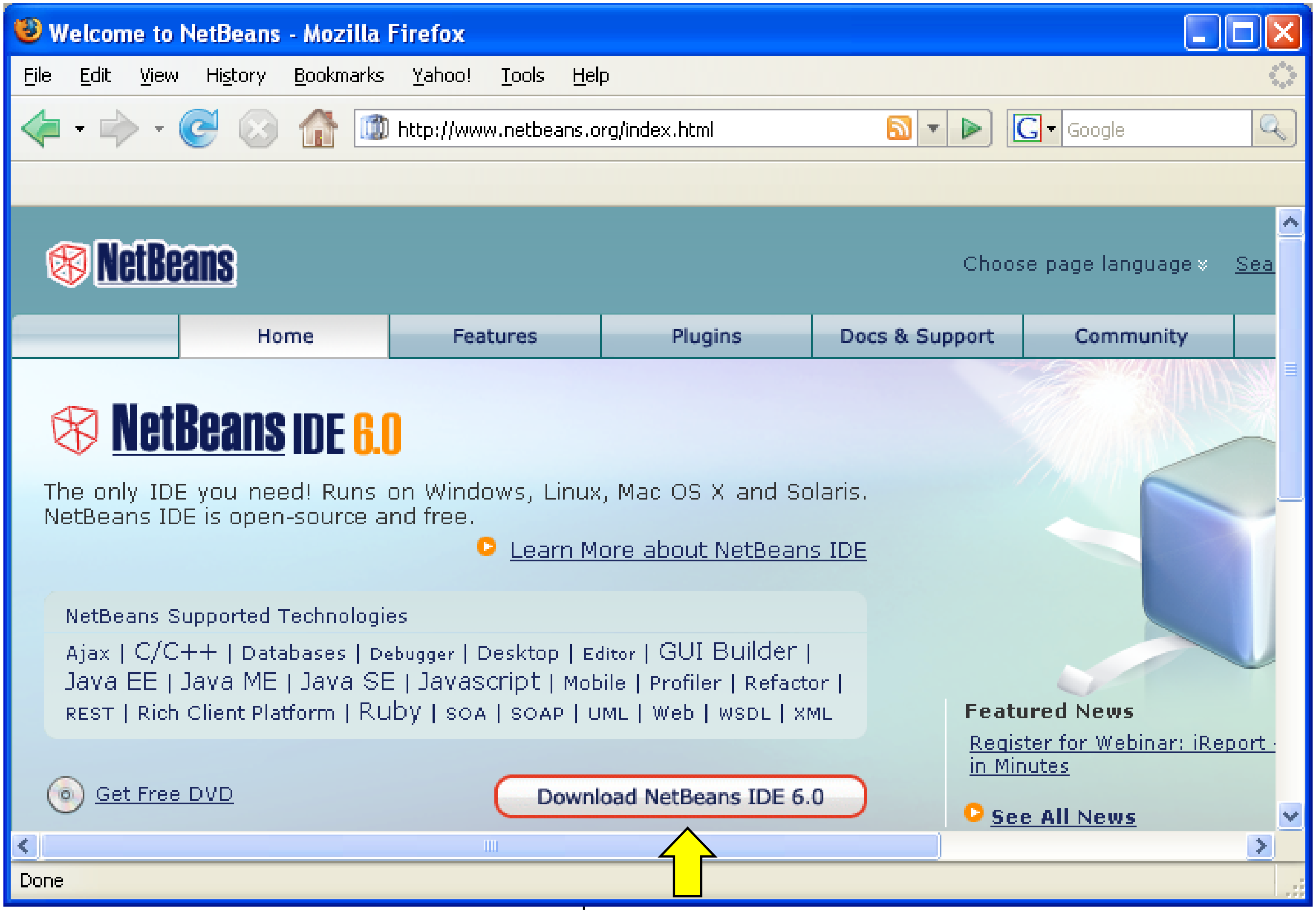
```
/**
 *
 */
package firstjava;

/**
 * @author
 *
 */
/*
 * This is a comment. The Log tag will be automaticall
 * history of changes
 * $Log: Person.java,v $
 * Revision 1.2  2008/02/16 20:25:02  ee462b30
 * Add a comment in Person.java
 *
 */
public class Person {
    final String p_lastName;
    final String p_firstName;

    public Person(String ln, String fn) {
        p_lastName = ln;
        p_firstName = fn;
    }
```

Log is replaced by the commit history.

Develop Java Projects in Netbeans

Install Netbeans at Your Own Computer

http://www.netbeans.org/



Welcome to NetBeans - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://www.netbeans.org/index.html
Google
NetBeans
Choose page language
Sea
Home
Features
Plugins
Docs & Support
Community
NetBeans IDE 6.0

The only IDE you need! Runs on Windows, Linux, Mac OS X and Solaris. NetBeans IDE is open-source and free.

Learn More about NetBeans IDE

NetBeans Supported Technologies

Ajax | C/C++ | Databases | Debugger | Desktop | Editor | GUI Builder | Java EE | Java ME | Java SE | Javascript | Mobile | Profiler | Refactor | REST | Rich Client Platform | Ruby | SOA | SOAP | UML | Web | WSDL | XML

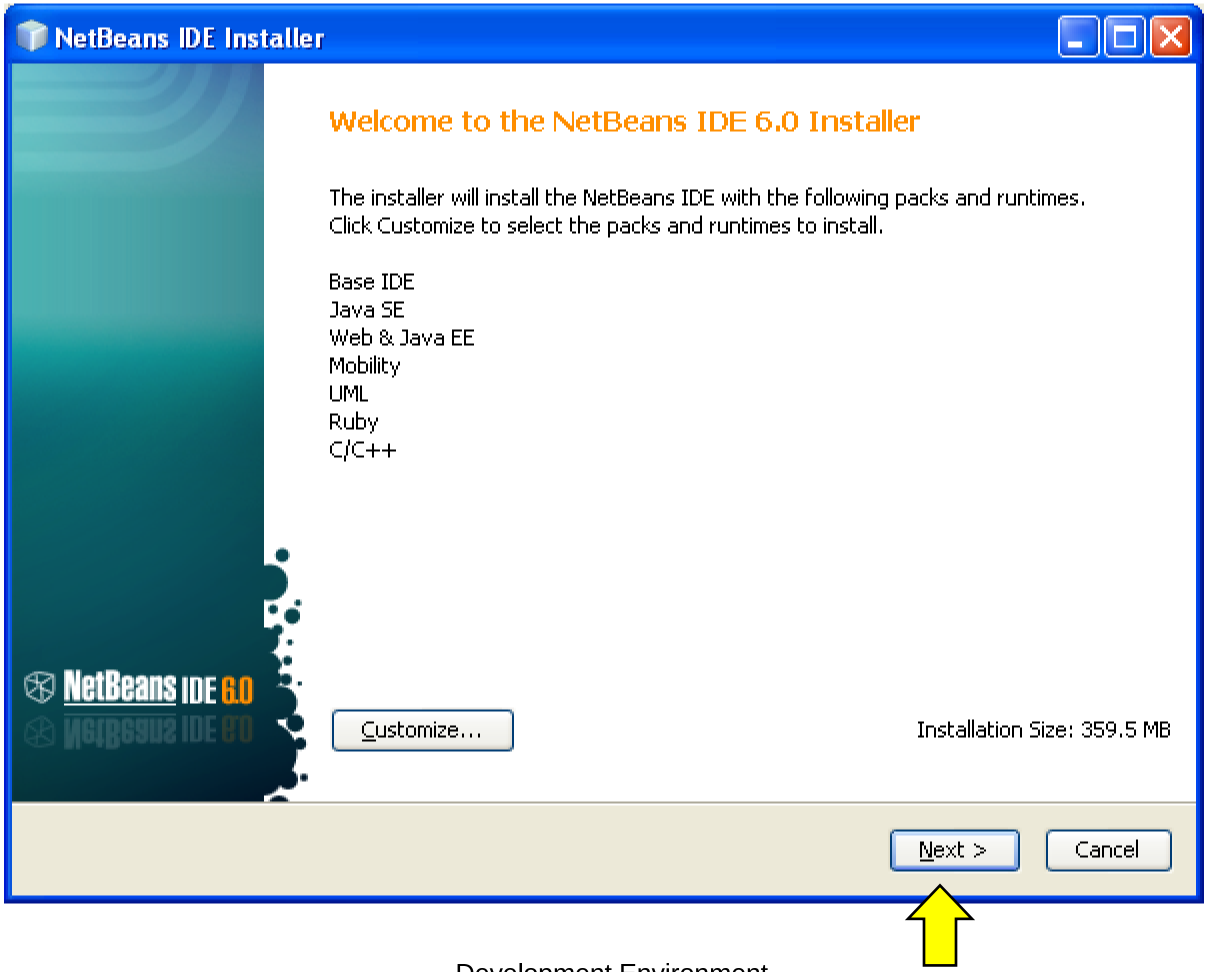
Get Free DVD

Download NetBeans IDE 6.0

Featured News

Register for Webinar: iReport in Minutes

See All News

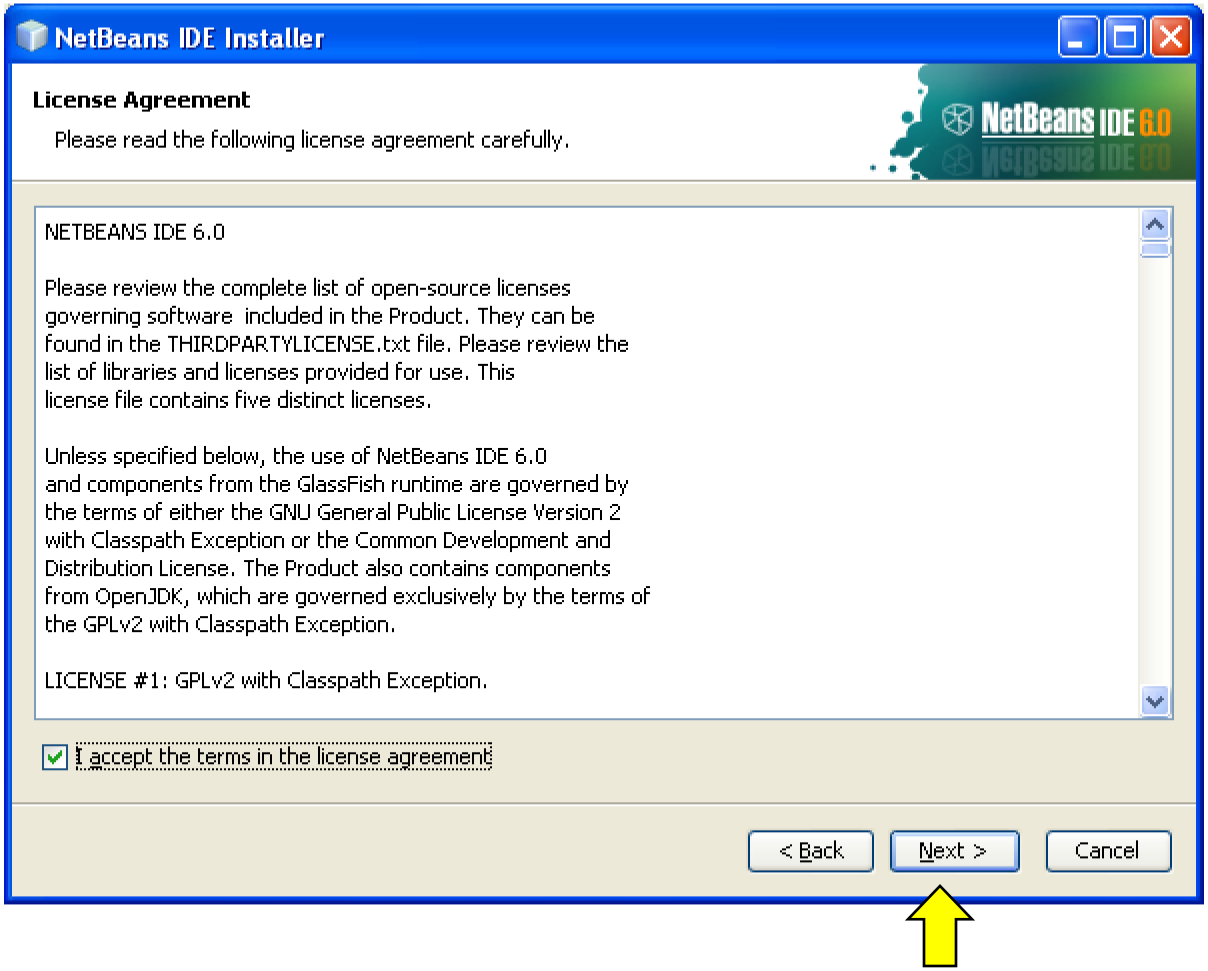


Welcome to the NetBeans IDE 6.0 Installer

The installer will install the NetBeans IDE with the following packs and runtimes.
Click Customize to select the packs and runtimes to install.

Base IDE
Java SE
Web & Java EE
Mobility
UML
Ruby
C/C++

Installation Size: 359.5 MB



License Agreement

Please read the following license agreement carefully.

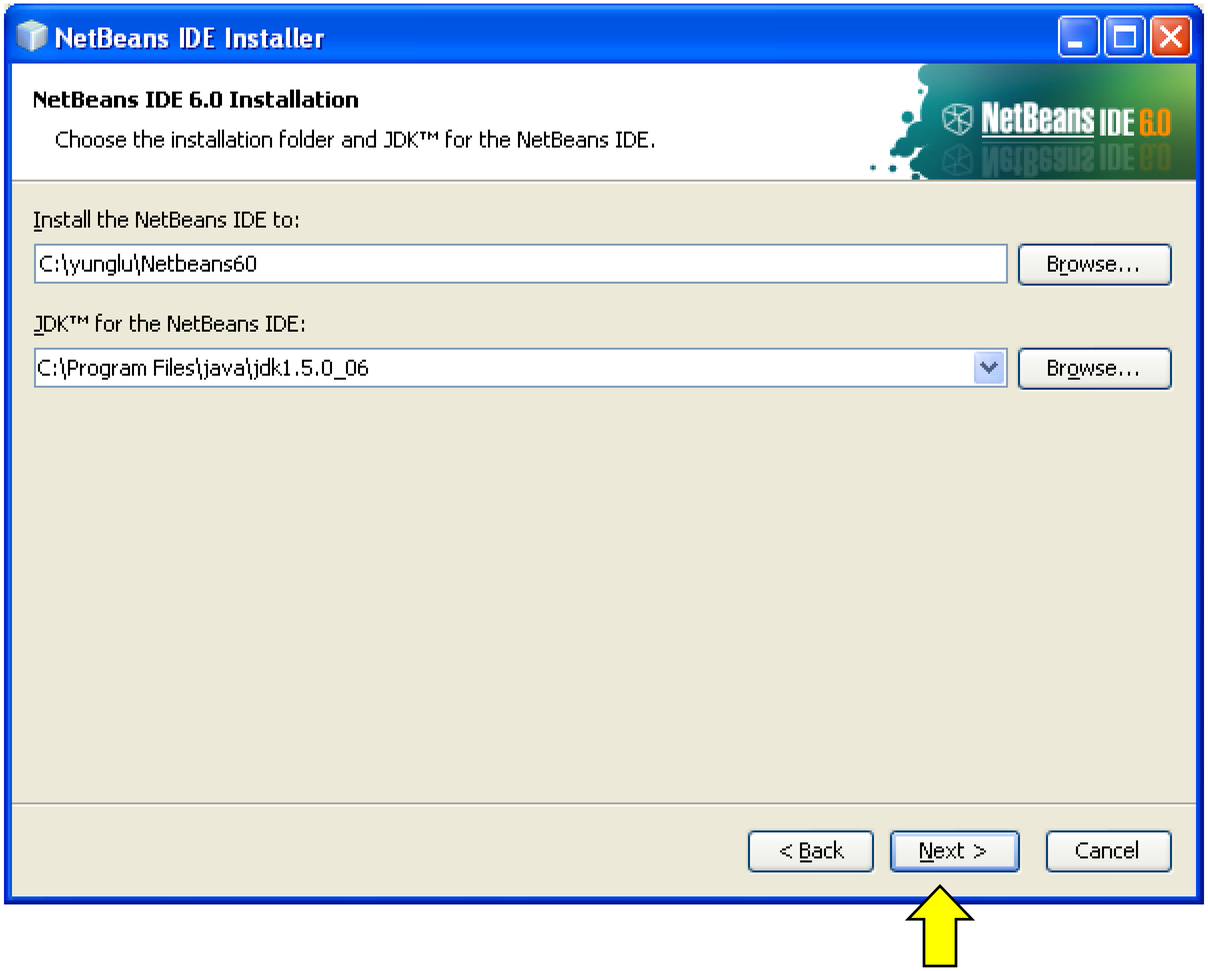
NETBEANS IDE 6.0

Please review the complete list of open-source licenses governing software included in the Product. They can be found in the THIRDPARTYLICENSE.txt file. Please review the list of libraries and licenses provided for use. This license file contains five distinct licenses.

Unless specified below, the use of NetBeans IDE 6.0 and components from the GlassFish runtime are governed by the terms of either the GNU General Public License Version 2 with Classpath Exception or the Common Development and Distribution License. The Product also contains components from OpenJDK, which are governed exclusively by the terms of the GPLv2 with Classpath Exception.

LICENSE #1: GPLv2 with Classpath Exception.

☑ I accept the terms in the license agreement



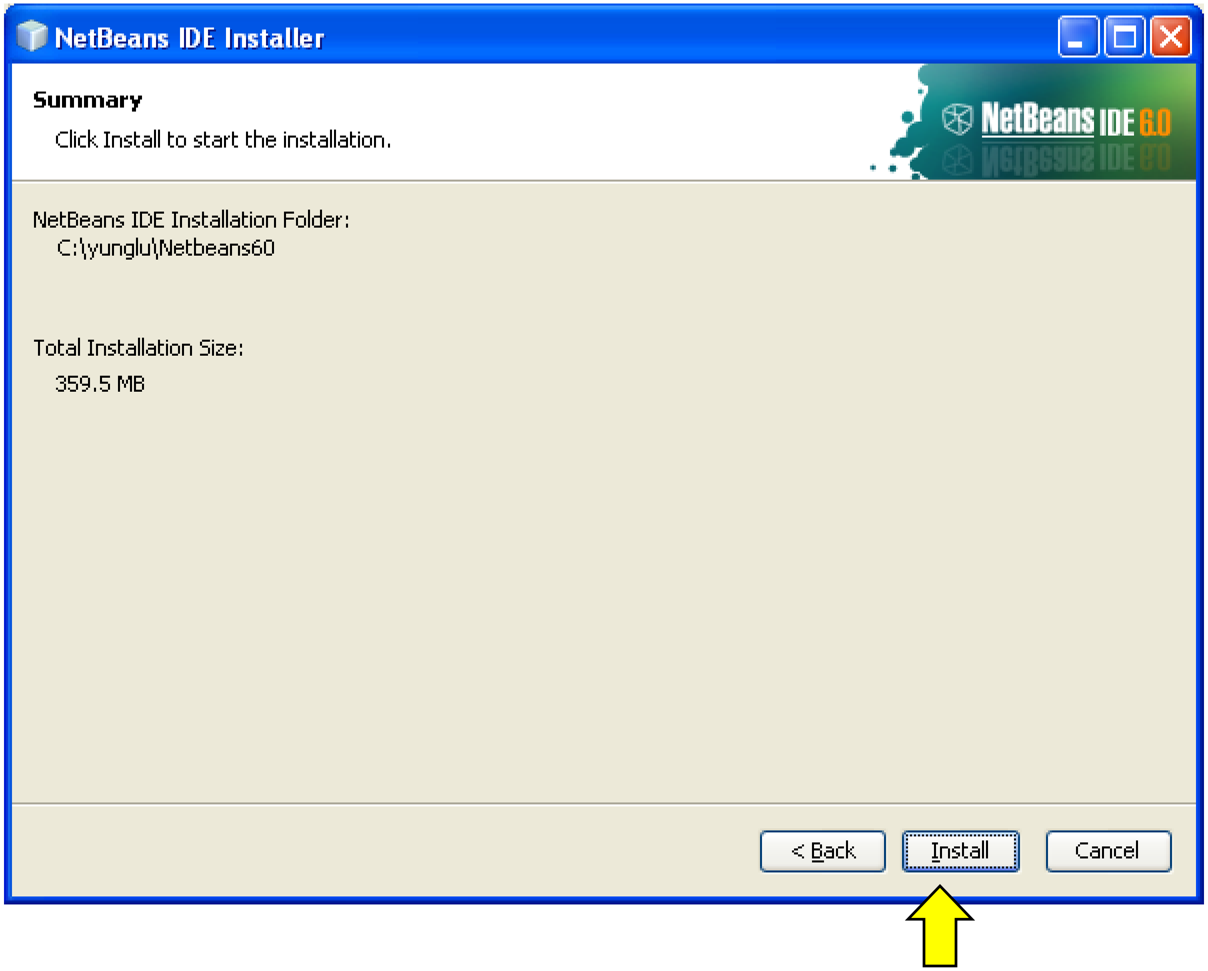
NetBeans IDE Installer

NetBeans IDE 6.0 Installation

Choose the installation folder and JDK™ for the NetBeans IDE.

Install the NetBeans IDE to: C:\yunglu\Netbeans60

JDK™ for the NetBeans IDE: C:\Program Files\java\jdk1.5.0_06

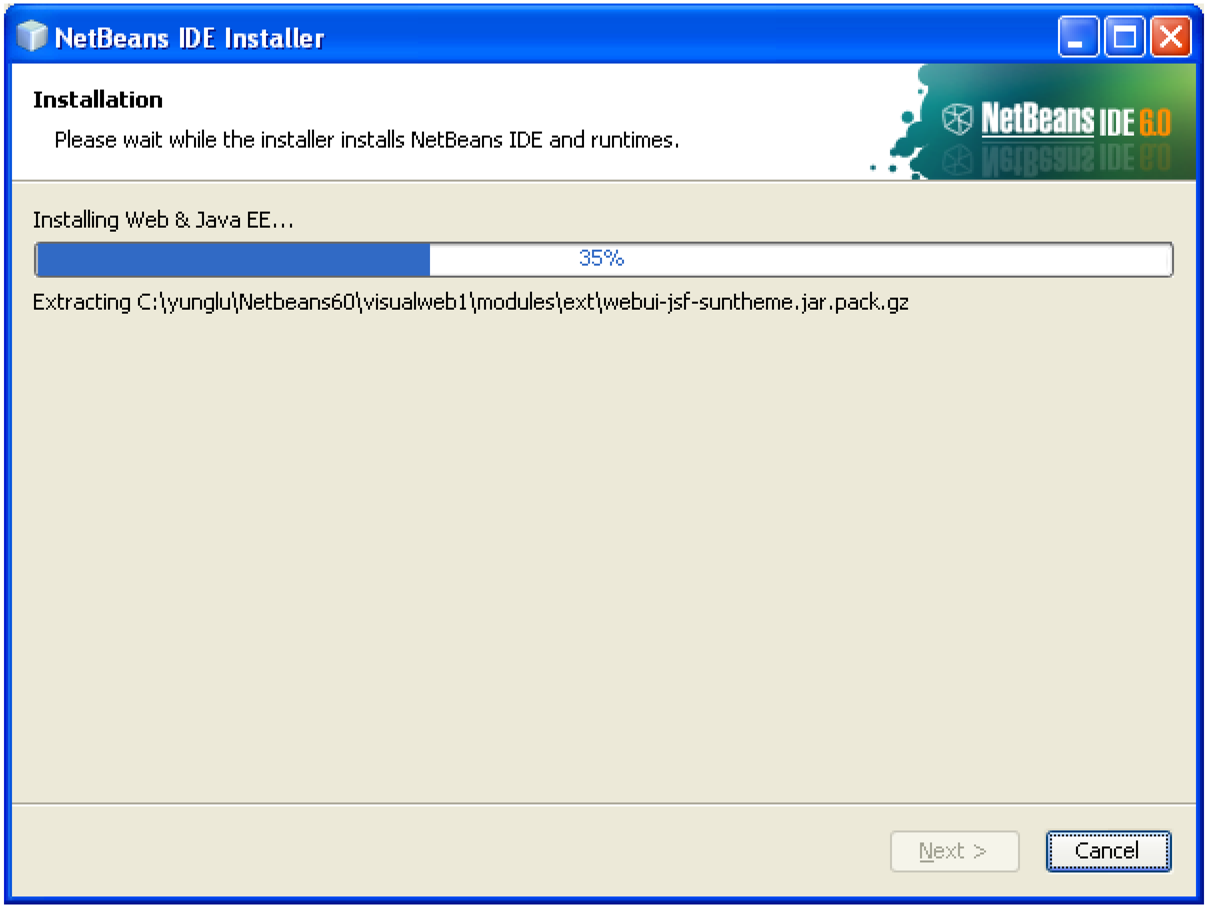


Summary

Click Install to start the installation.

NetBeans IDE Installation Folder:
C:\yunglu\Netbeans60

Total Installation Size:
359.5 MB



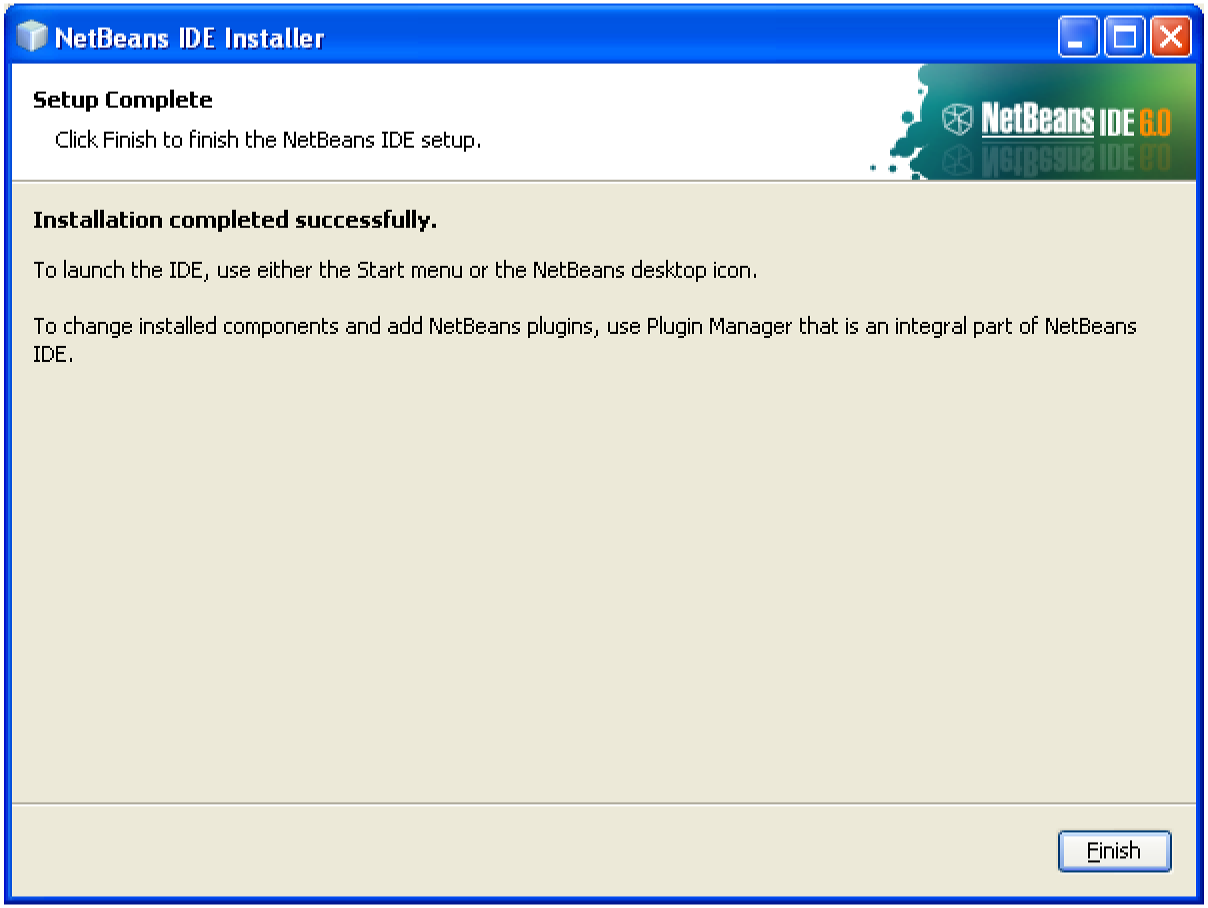
Installation

Please wait while the installer installs NetBeans IDE and runtimes.

Installing Web & Java EE...

35%

Extracting C:\yunglu\Netbeans60\visualweb1\modules\ext\webui-jsf-suntheme.jar.pack.gz



Setup Complete

Click Finish to finish the NetBeans IDE setup.

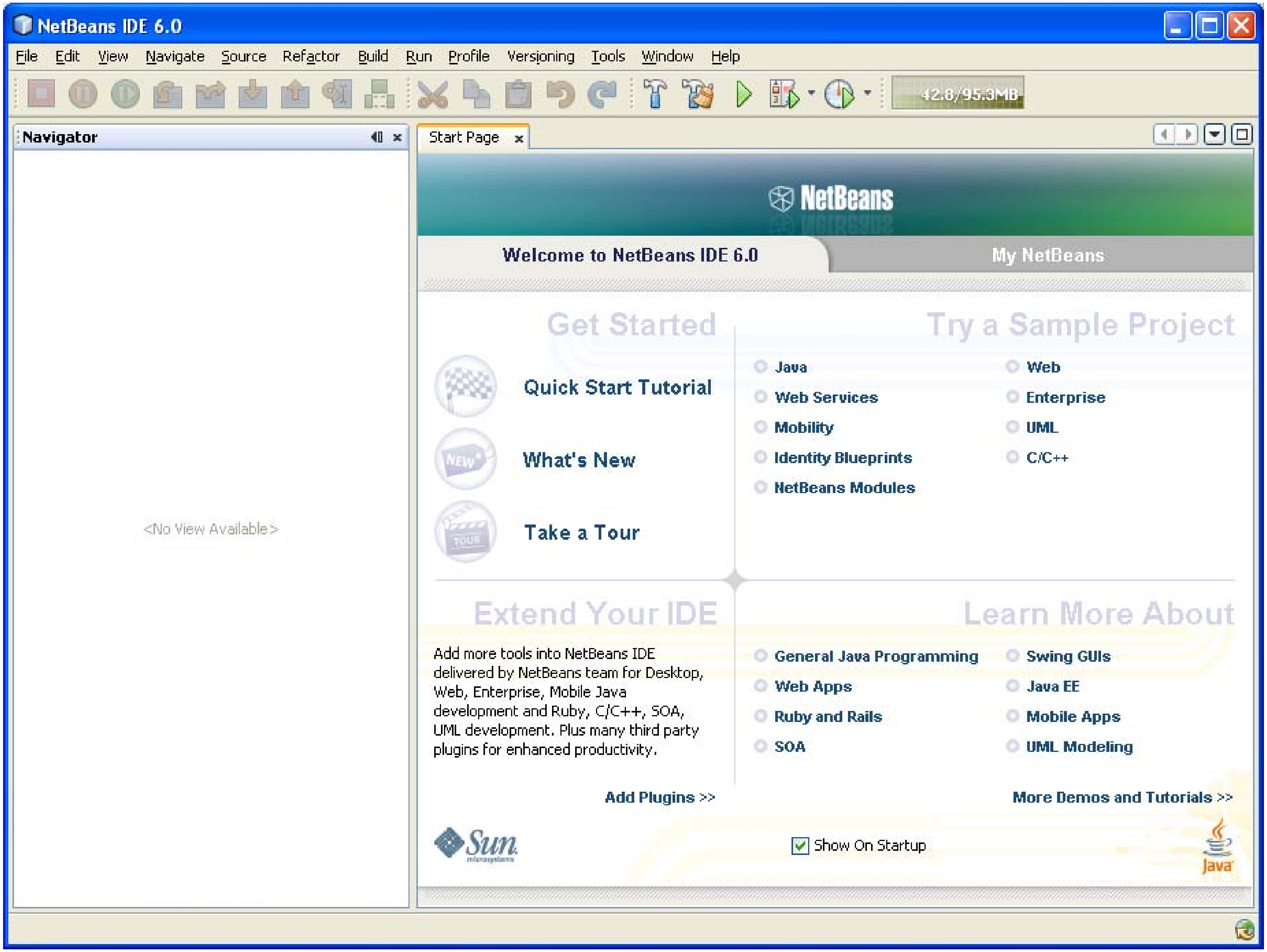
Installation completed successfully.

To launch the IDE, use either the Start menu or the NetBeans desktop icon.

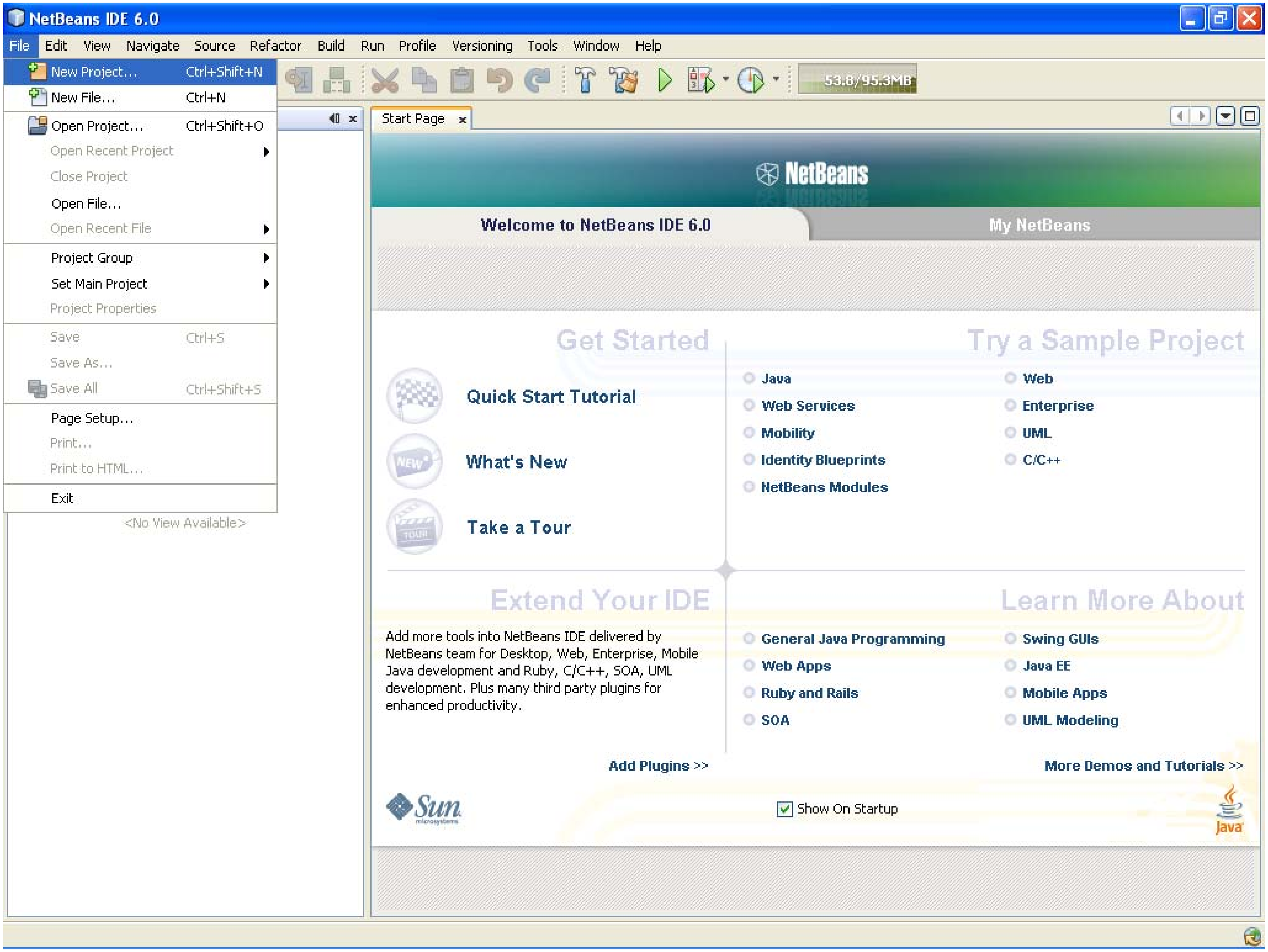
To change installed components and add NetBeans plugins, use Plugin Manager that is an integral part of NetBeans IDE.

Loading modules...

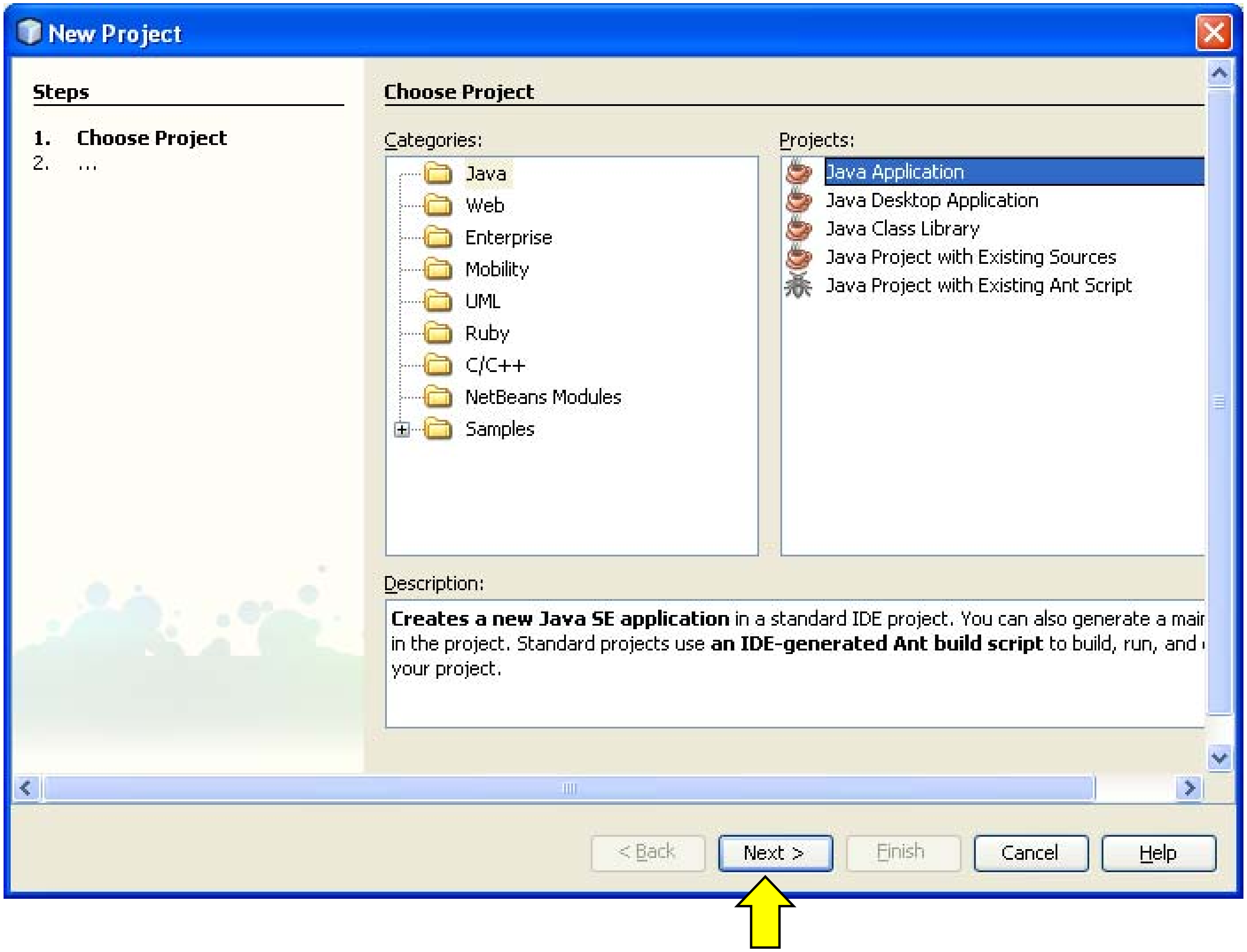
NetBeans IDE and NetBeans Platform are based on software from netbeans.org, which has been dual licensed under the Common Development and Distribution License (CDDL) and the GNU General Public License version 2 with Classpath exception. For more information, please visit www.netbeans.org.



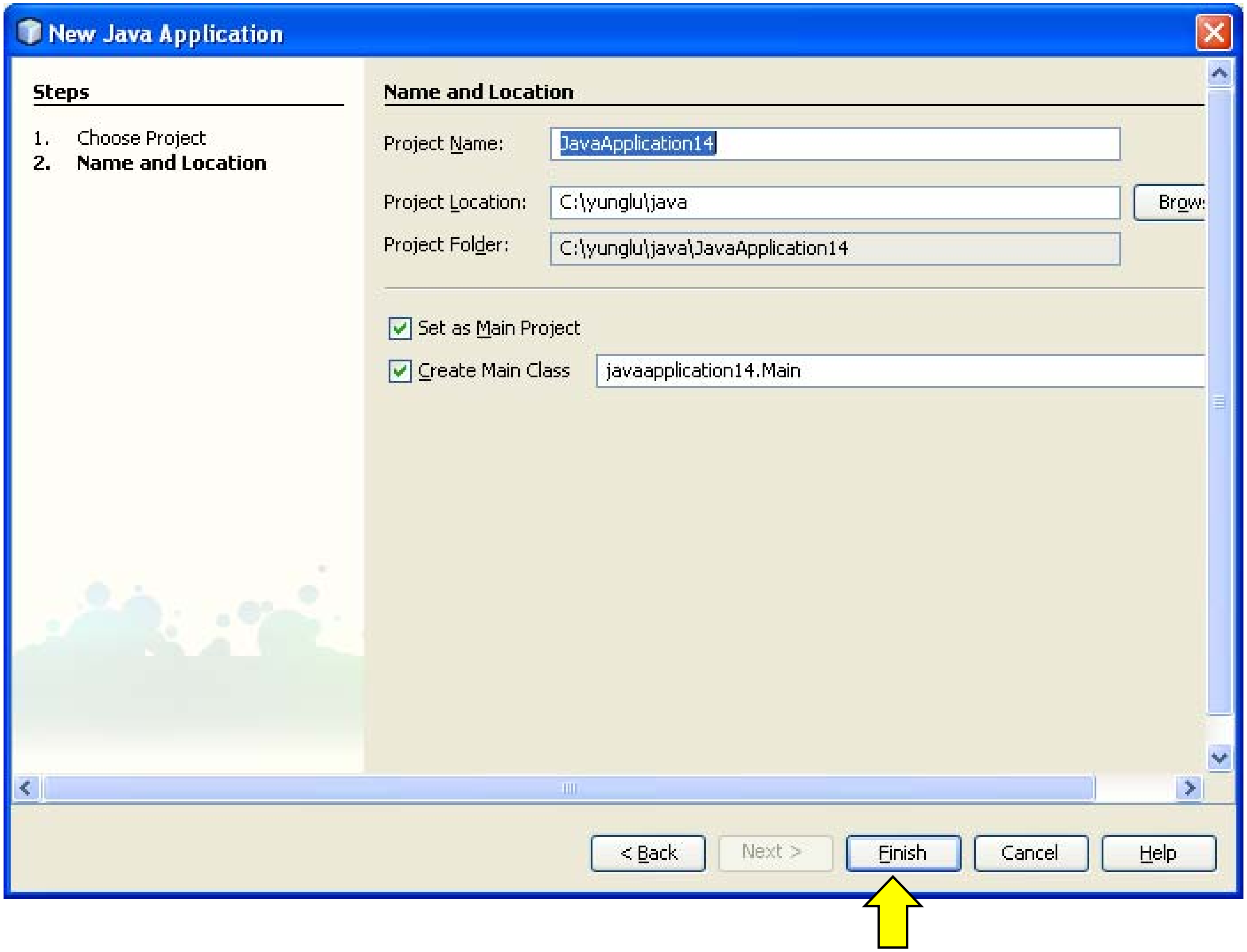
Add more tools into NetBeans IDE delivered by NetBeans team for Desktop, Web, Enterprise, Mobile Java development and Ruby, C/C++, SOA, UML development. Plus many third party plugins for enhanced productivity.



Add more tools into NetBeans IDE delivered by NetBeans team for Desktop, Web, Enterprise, Mobile Java development and Ruby, C/C++, SOA, UML development. Plus many third party plugins for enhanced productivity.



Creates a new Java SE application in a standard IDE project. You can also generate a mai
in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and
your project.



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

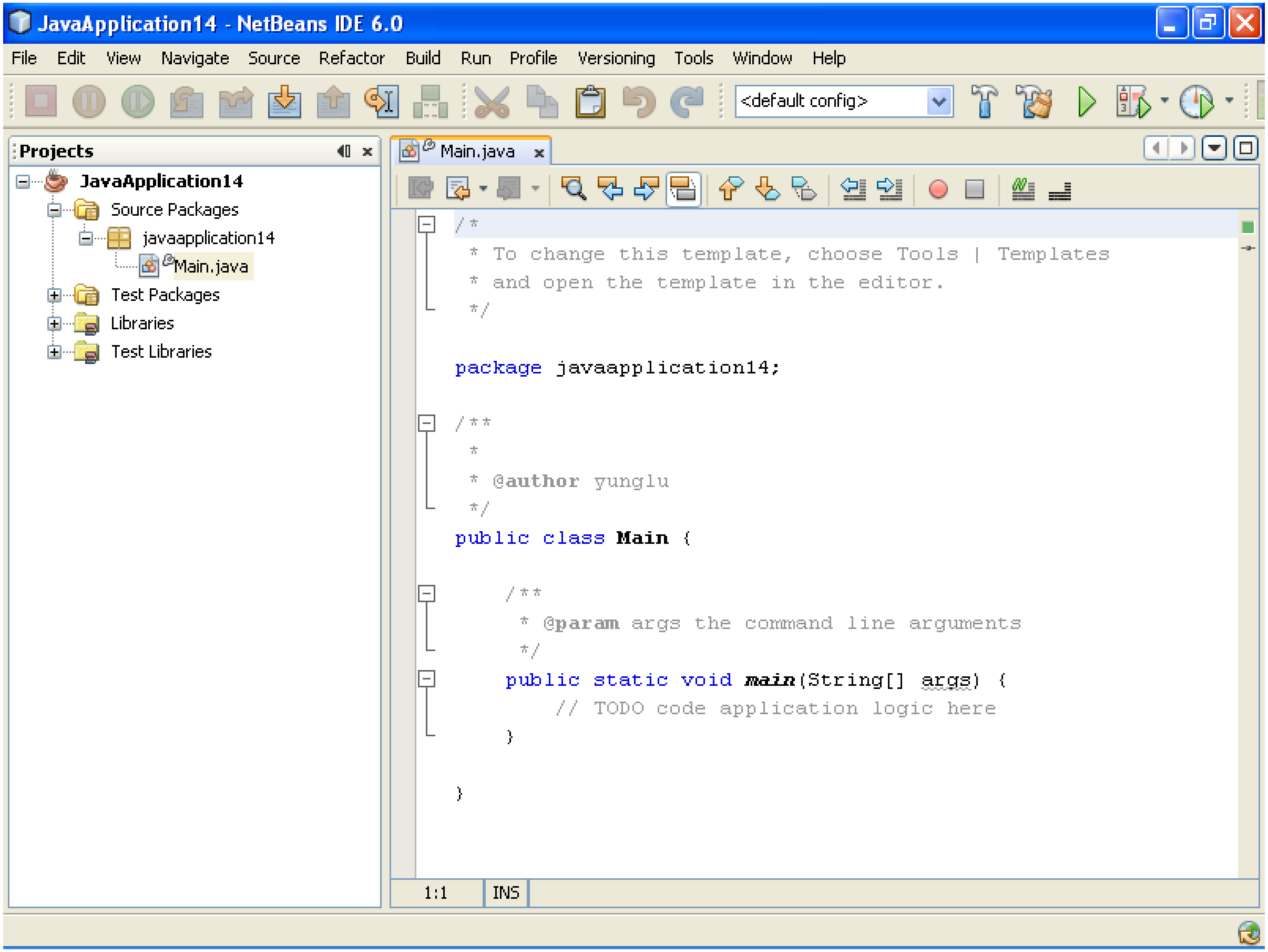
Project Name: JavaApplication14

Project Location: C:\yunglu\java

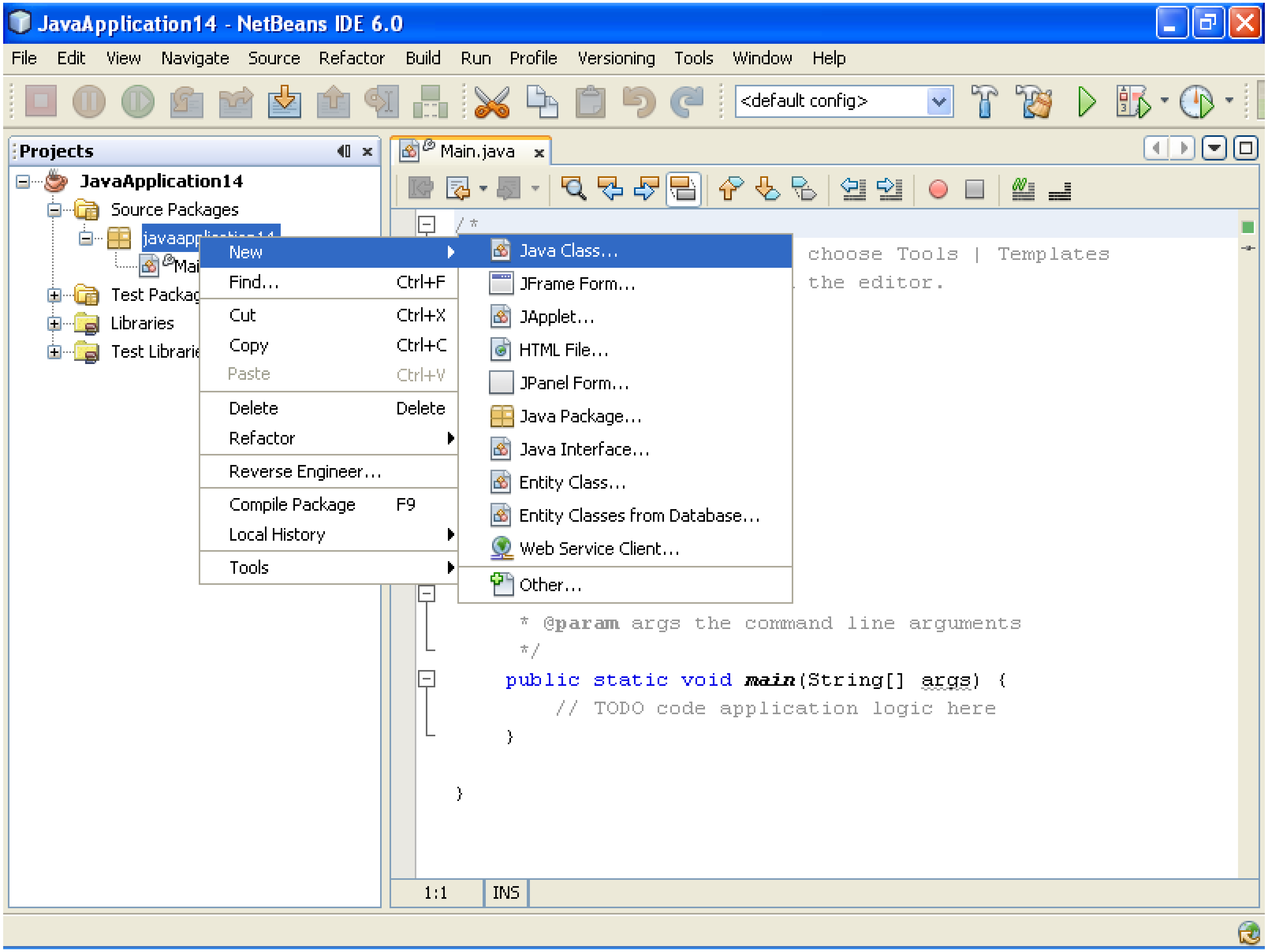
Project Folder: C:\yunglu\java\JavaApplication14

☑ Set as Main Project

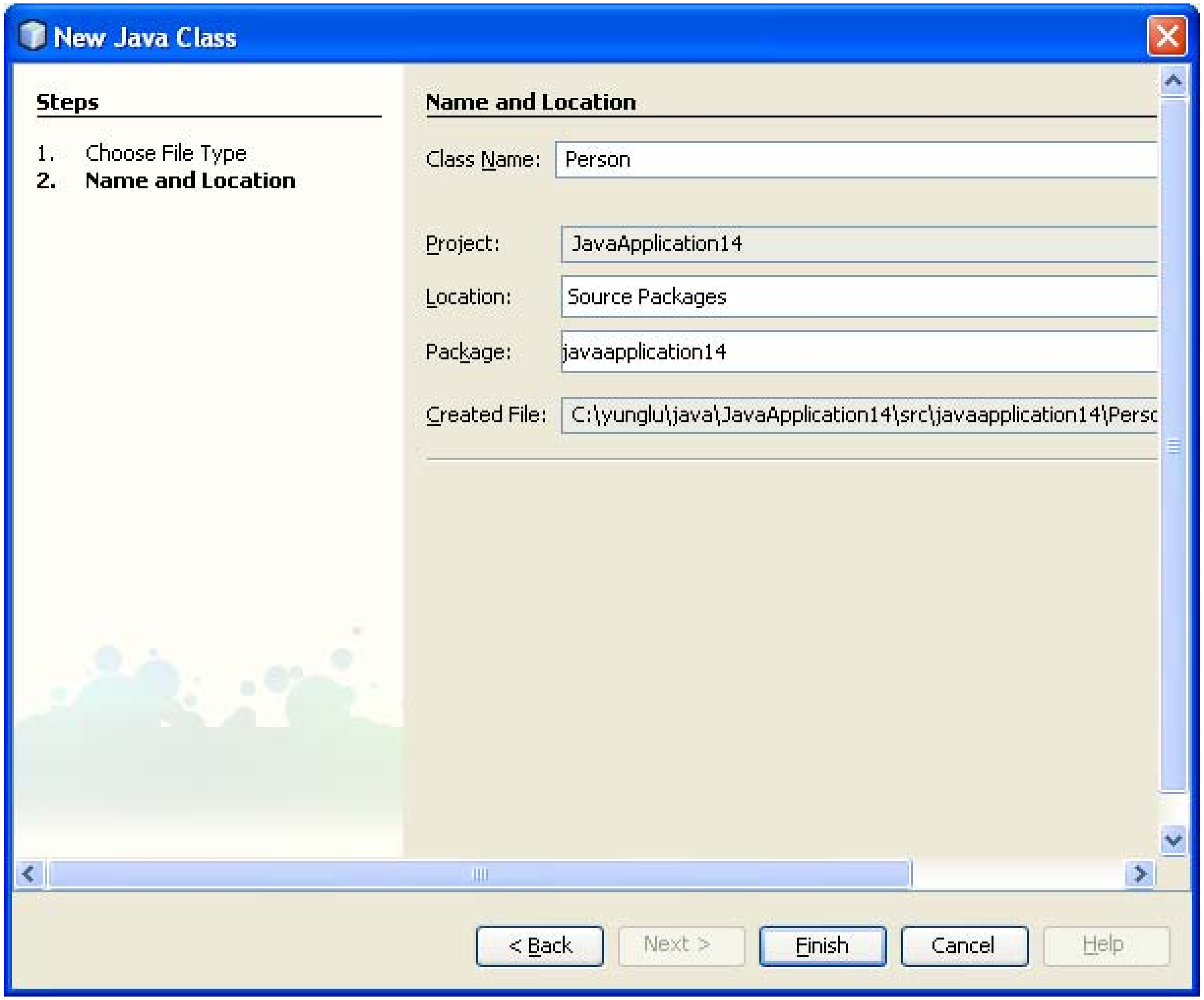
☑ Create Main Class javaapplication14.Main



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package javaapplication14;

/**
 *
 * @author yunglu
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```



```
/*
        choose Tools | Templates
        the editor.

     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```



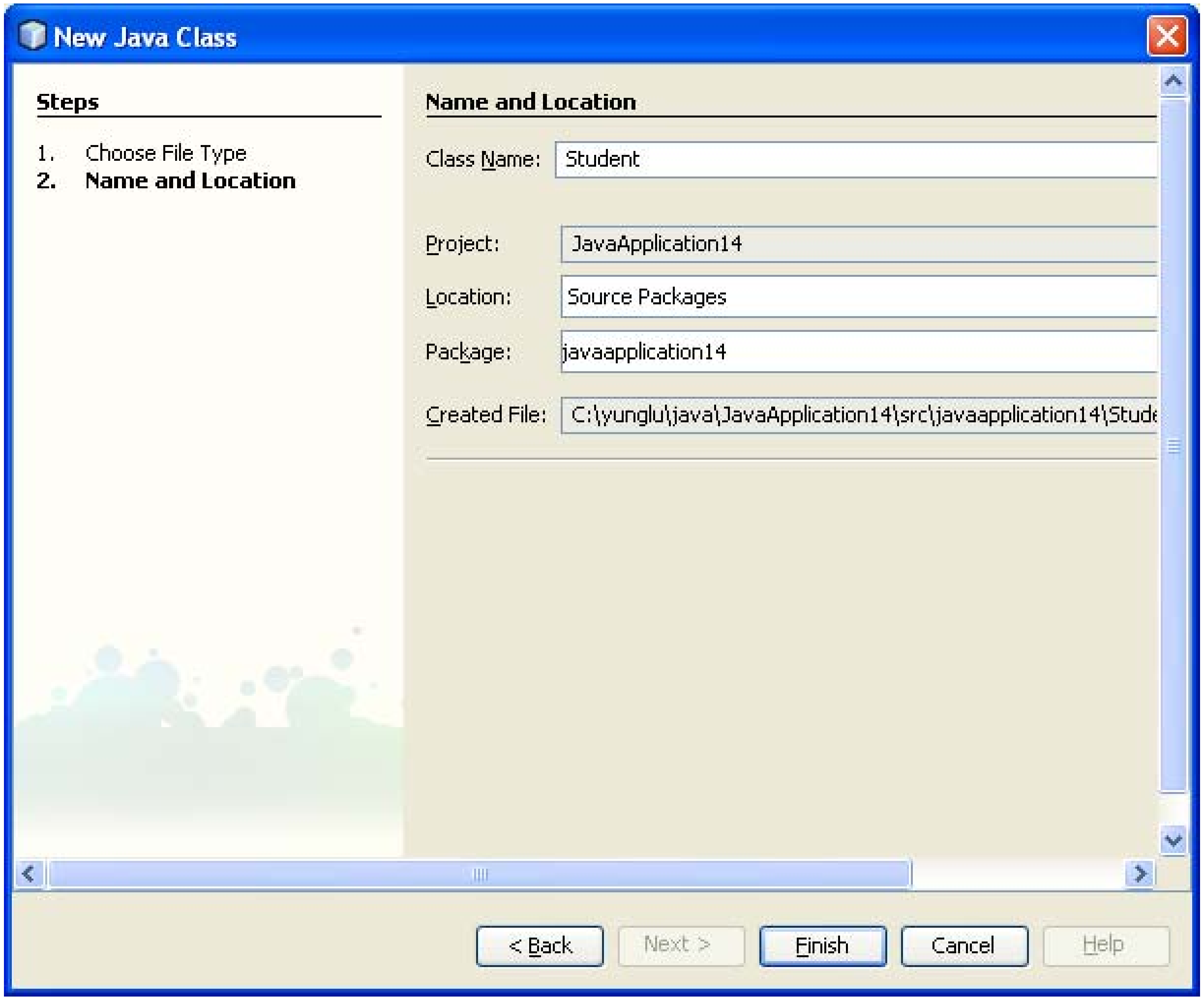
Class Name: Person

Project: JavaApplication14

Location: Source Packages

Package: javaapplication14

Created File: C:\yunglu\java\JavaApplication14\src\javaapplication14\Persc



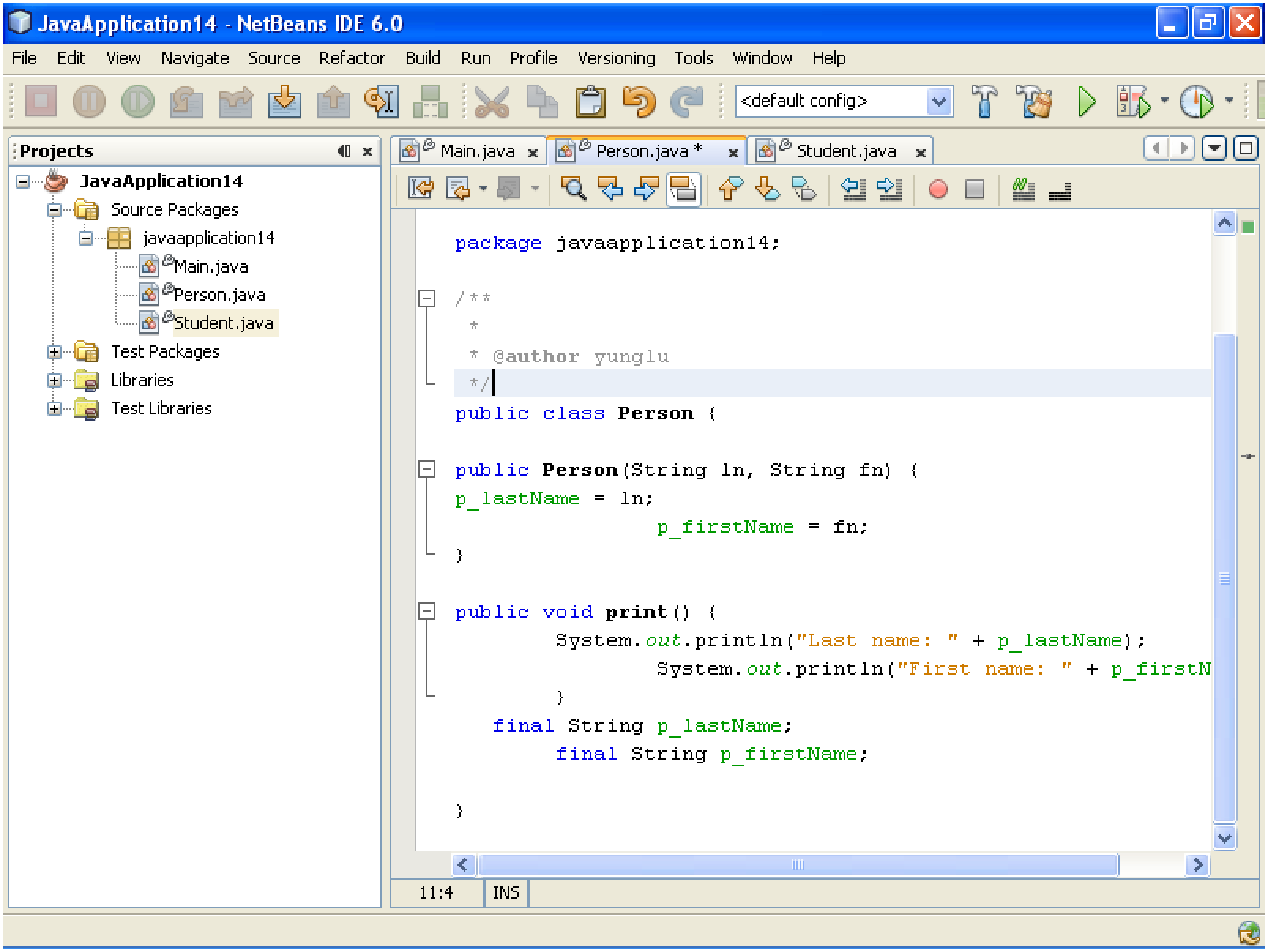
Class Name: Student

Project: JavaApplication14

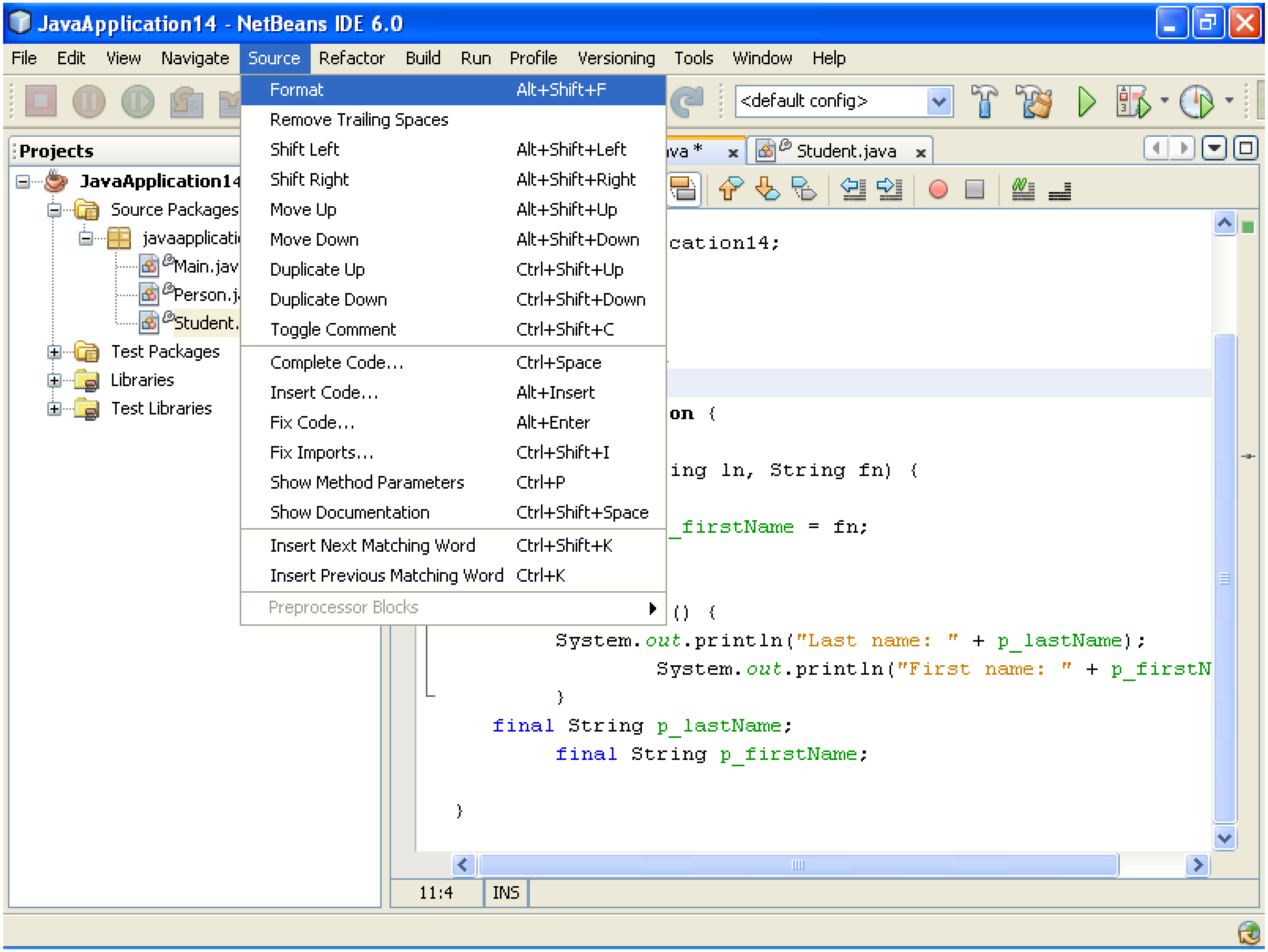
Location: Source Packages

Package: javaapplication14

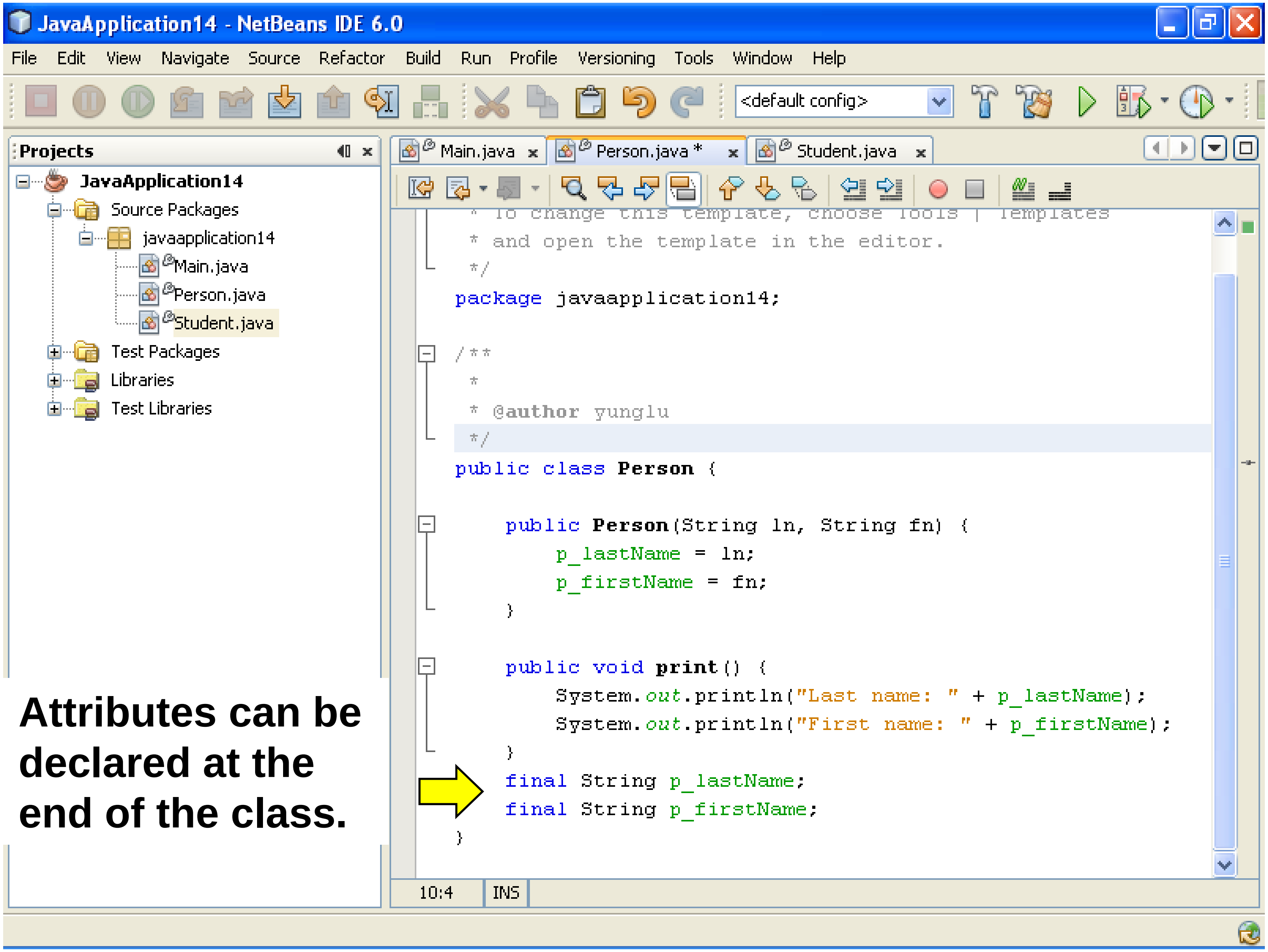
Created File: C:\yunglu\java\JavaApplication14\src\javaapplication14\Stude



```
package javaapplication14;

/**
 *
 * @author yunglu
 */
public class Person {

public Person(String ln, String fn) {
p_lastName = ln;
                p_firstName = fn;
}

public void print() {
        System.out.println("Last name: " + p_lastName);
                System.out.println("First name: " + p_firstN
        }
   final String p_lastName;
        final String p_firstName;

}
```

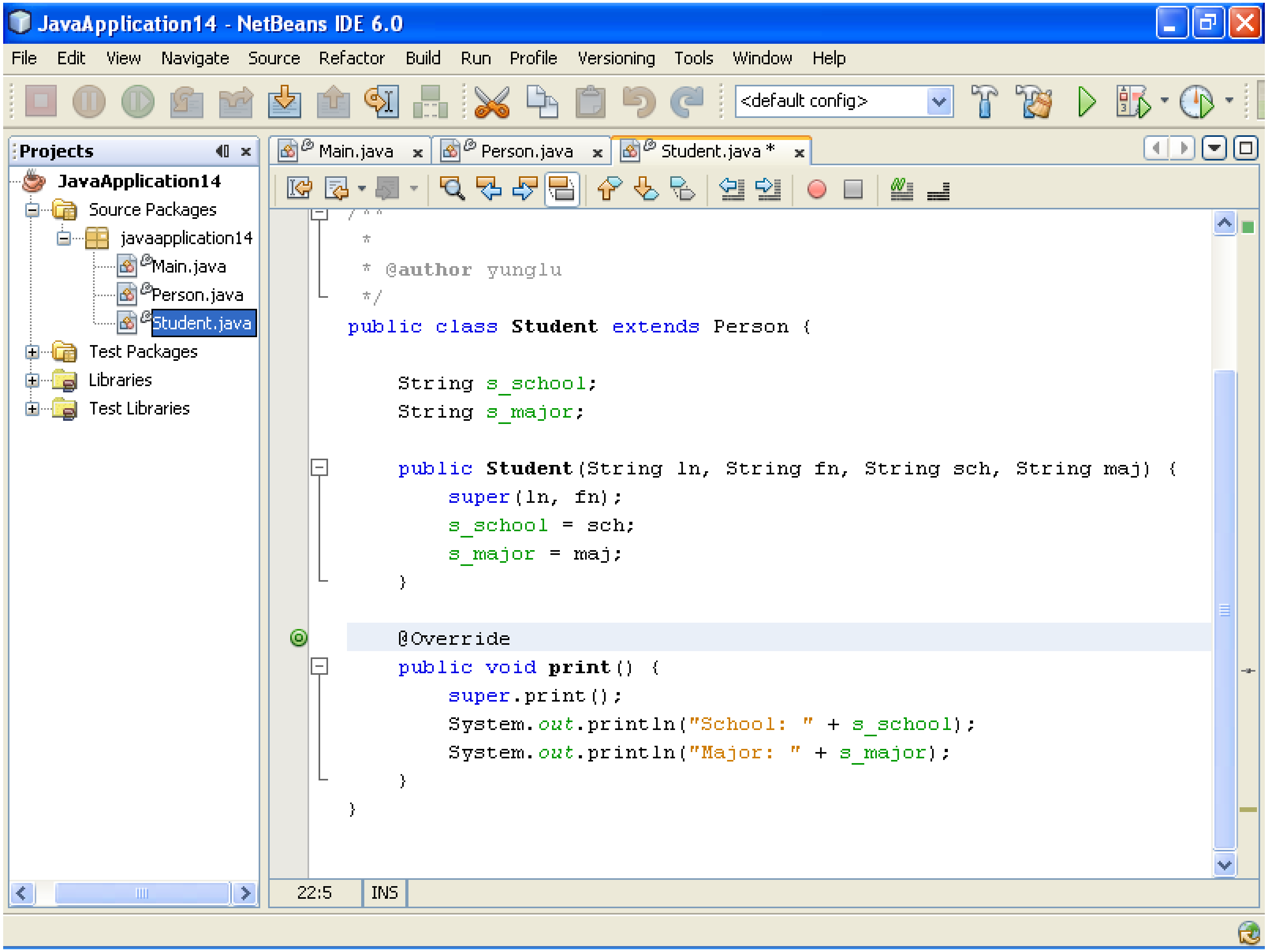

JavaApplication14 - NetBeans IDE 6.0
File
Edit
View
Navigate
Source
Refactor
Build
Run
Profile
Versioning
Tools
Window
Help
Format Alt+Shift+F
Remove Trailing Spaces
Shift Left Alt+Shift+Left
Shift Right Alt+Shift+Right
Move Up Alt+Shift+Up
Move Down Alt+Shift+Down
Duplicate Up Ctrl+Shift+Up
Duplicate Down Ctrl+Shift+Down
Toggle Comment Ctrl+Shift+C
Complete Code... Ctrl+Space
Insert Code... Alt+Insert
Fix Code... Alt+Enter
Fix Imports... Ctrl+Shift+I
Show Method Parameters Ctrl+P
Show Documentation Ctrl+Shift+Space
Insert Next Matching Word Ctrl+Shift+K
Insert Previous Matching Word Ctrl+K
Preprocessor Blocks
<default config>
Projects
JavaApplication14
Source Packages
javaapplicati
Main.jav
Person.j
Student.
Test Packages
Libraries
Test Libraries
va *
Student.java
11:4
INS

```
cation14;

on {

ing ln, String fn) {

_firstName = fn;

() {
        System.out.println("Last name: " + p_lastName);
            System.out.println("First name: " + p_firstN
        }
    final String p_lastName;
        final String p_firstName;

}
```

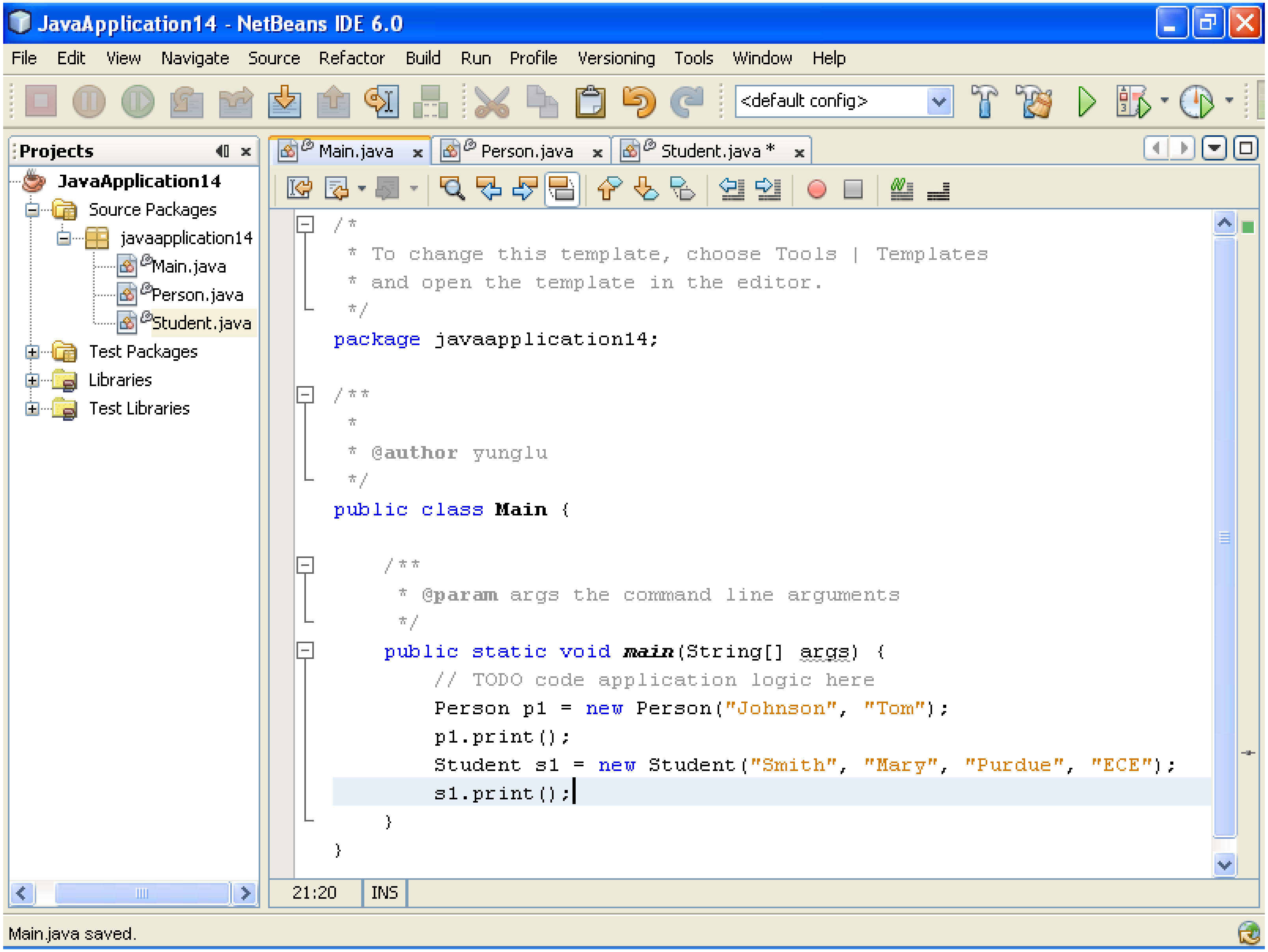


```
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication14;

/**
 *
 * @author yunglu
 */
public class Person {

    public Person(String ln, String fn) {
        p_lastName = ln;
        p_firstName = fn;
    }

    public void print() {
        System.out.println("Last name: " + p_lastName);
        System.out.println("First name: " + p_firstName);
    }
    final String p_lastName;
    final String p_firstName;
}
```

Attributes can be declared at the end of the class.

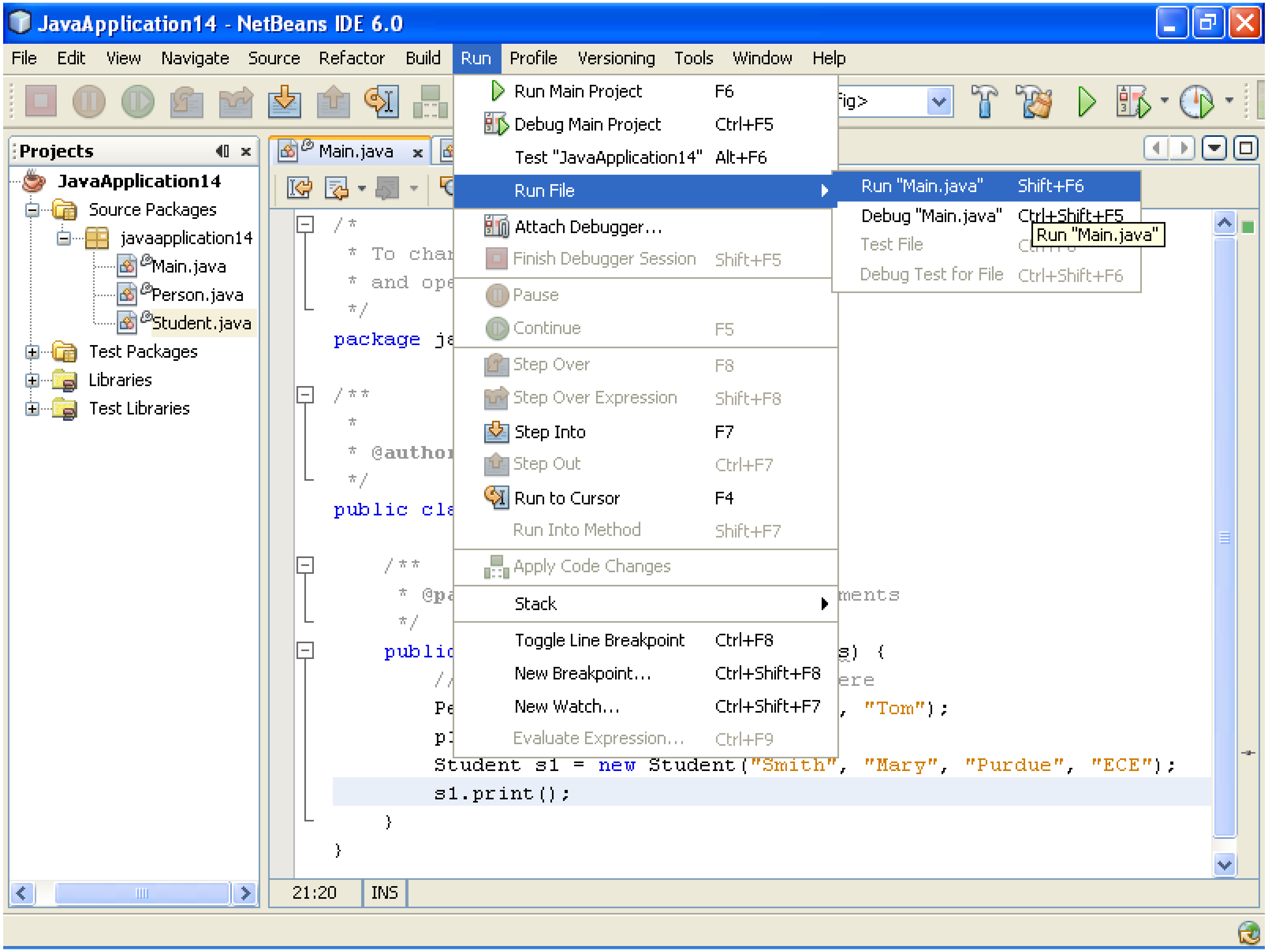


```
/**
 *
 * @author yunglu
 */
public class Student extends Person {

    String s_school;
    String s_major;

    public Student(String ln, String fn, String sch, String maj) {
        super(ln, fn);
        s_school = sch;
        s_major = maj;
    }

    @Override
    public void print() {
        super.print();
        System.out.println("School: " + s_school);
        System.out.println("Major: " + s_major);
    }
}
```

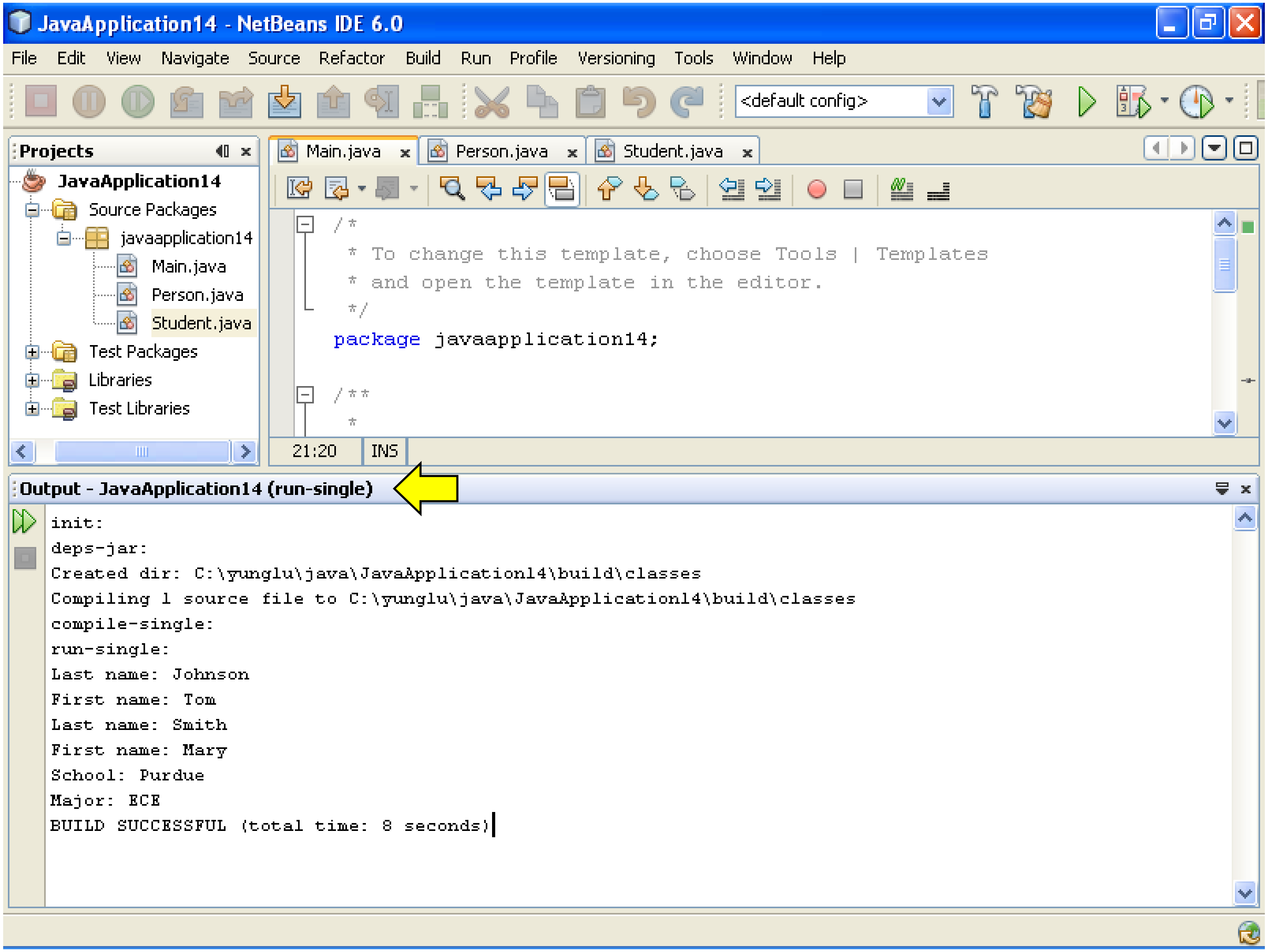


```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication14;

/**
 *
 * @author yunglu
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Person p1 = new Person("Johnson", "Tom");
        p1.print();
        Student s1 = new Student("Smith", "Mary", "Purdue", "ECE");
        s1.print();
    }
}
```

Main.java saved.

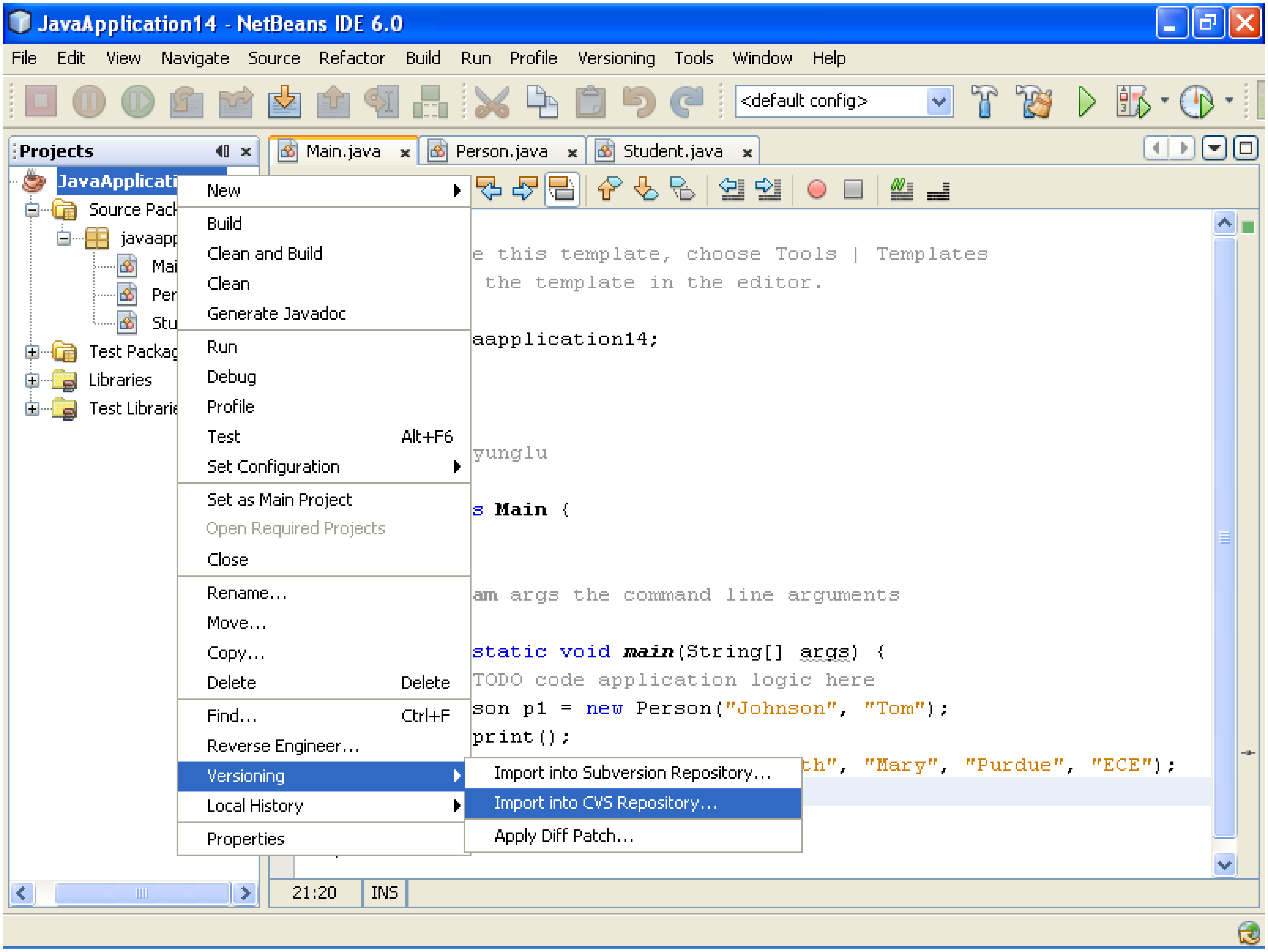


```
/*
 * To char
 * and ope
 */
package ja

/**
 *
 * @author
 */
public cla

    /**
     * @pa                                        ments
     */
    publi                                         s) {
        //                                        ere
        Pe                                        , "Tom");
        p
        Student s1 = new Student("Smith", "Mary", "Purdue", "ECE");
        s1.print();
    }
}
```

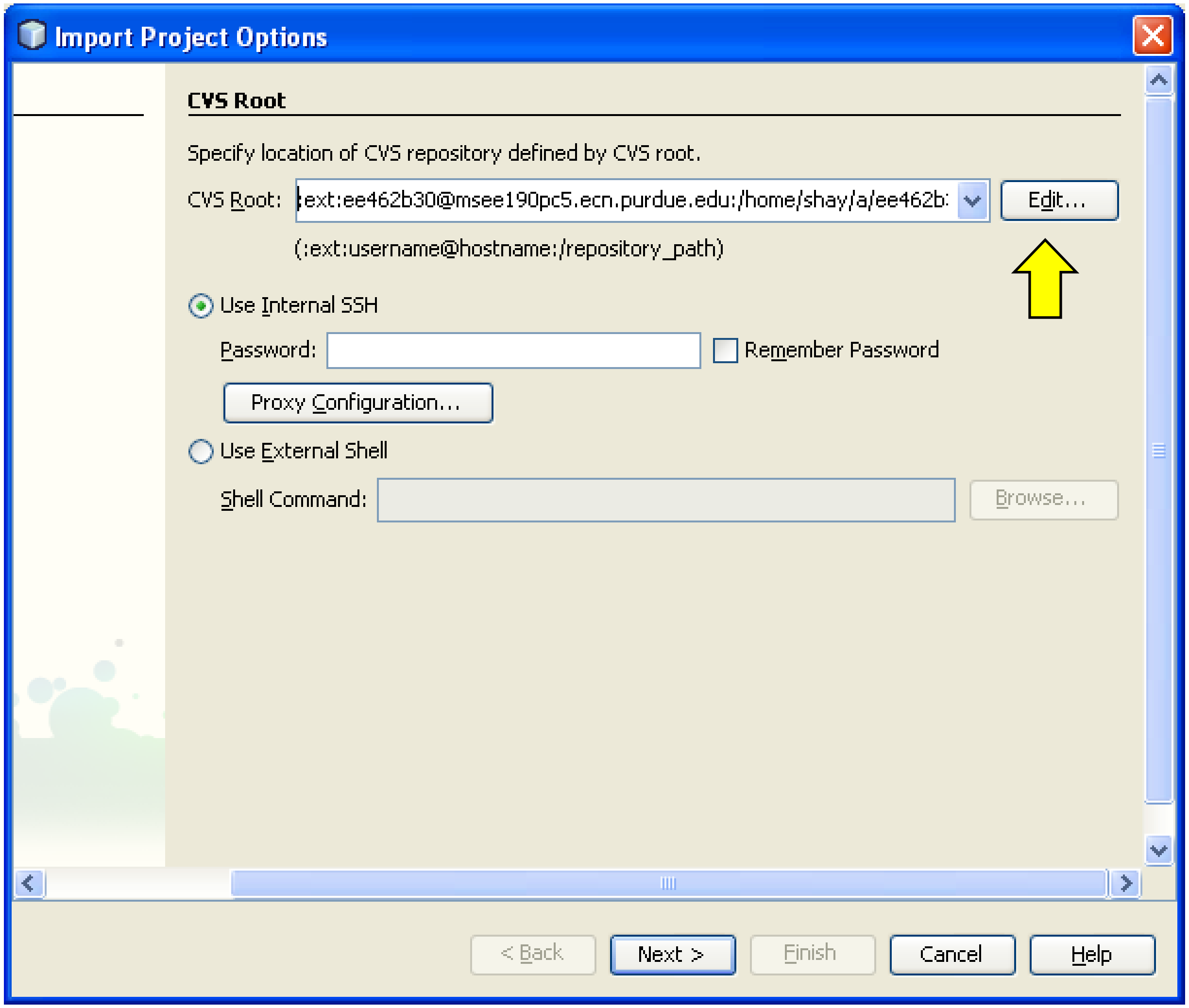


```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication14;

/**
 *
```

```
init:
deps-jar:
Created dir: C:\yunglu\java\JavaApplication14\build\classes
Compiling 1 source file to C:\yunglu\java\JavaApplication14\build\classes
compile-single:
run-single:
Last name: Johnson
First name: Tom
Last name: Smith
First name: Mary
School: Purdue
Major: ECE
BUILD SUCCESSFUL (total time: 8 seconds)
```



```
e this template, choose Tools | Templates
 the template in the editor.

aapplication14;

yunglu

s Main {

am args the command line arguments

static void main(String[] args) {
TODO code application logic here
son p1 = new Person("Johnson", "Tom");
print();
th", "Mary", "Purdue", "ECE");
```



Import Project Options

CVS Root

Specify location of CVS repository defined by CVS root.

CVS Root: :ext:ee462b30@msee190pc5.ecn.purdue.edu:/home/shay/a/ee462b:

(:ext:username@hostname:/repository_path)

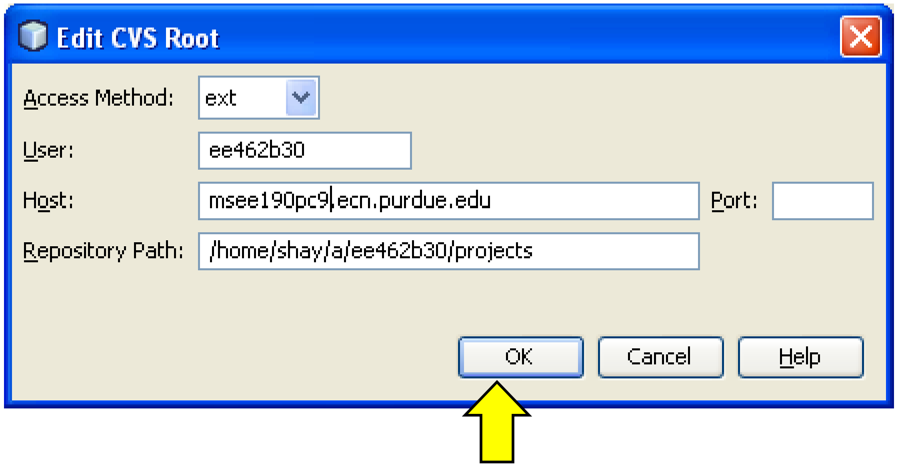
● Use Internal SSH

Password:

□ Remember Password

○ Use External Shell

Shell Command:



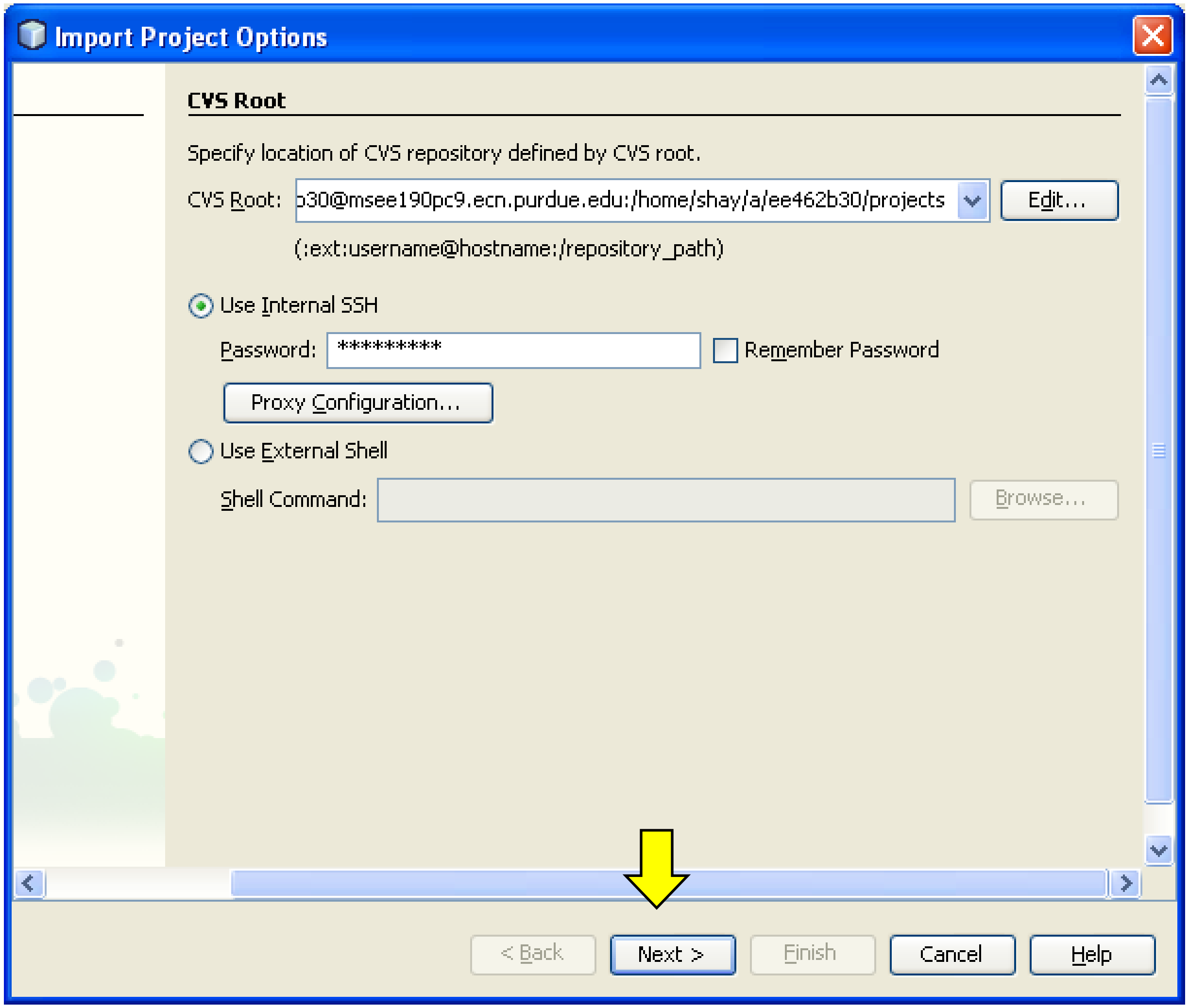
Access Method: ext

User: ee462b30

Host: msee190pc9.ecn.purdue.edu

Port:

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



Import Project Options

CVS Root

Specify location of CVS repository defined by CVS root.

CVS Root: b30@msee190pc9.ecn.purdue.edu:/home/shay/a/ee462b30/projects

(:ext:username@hostname:/repository_path)

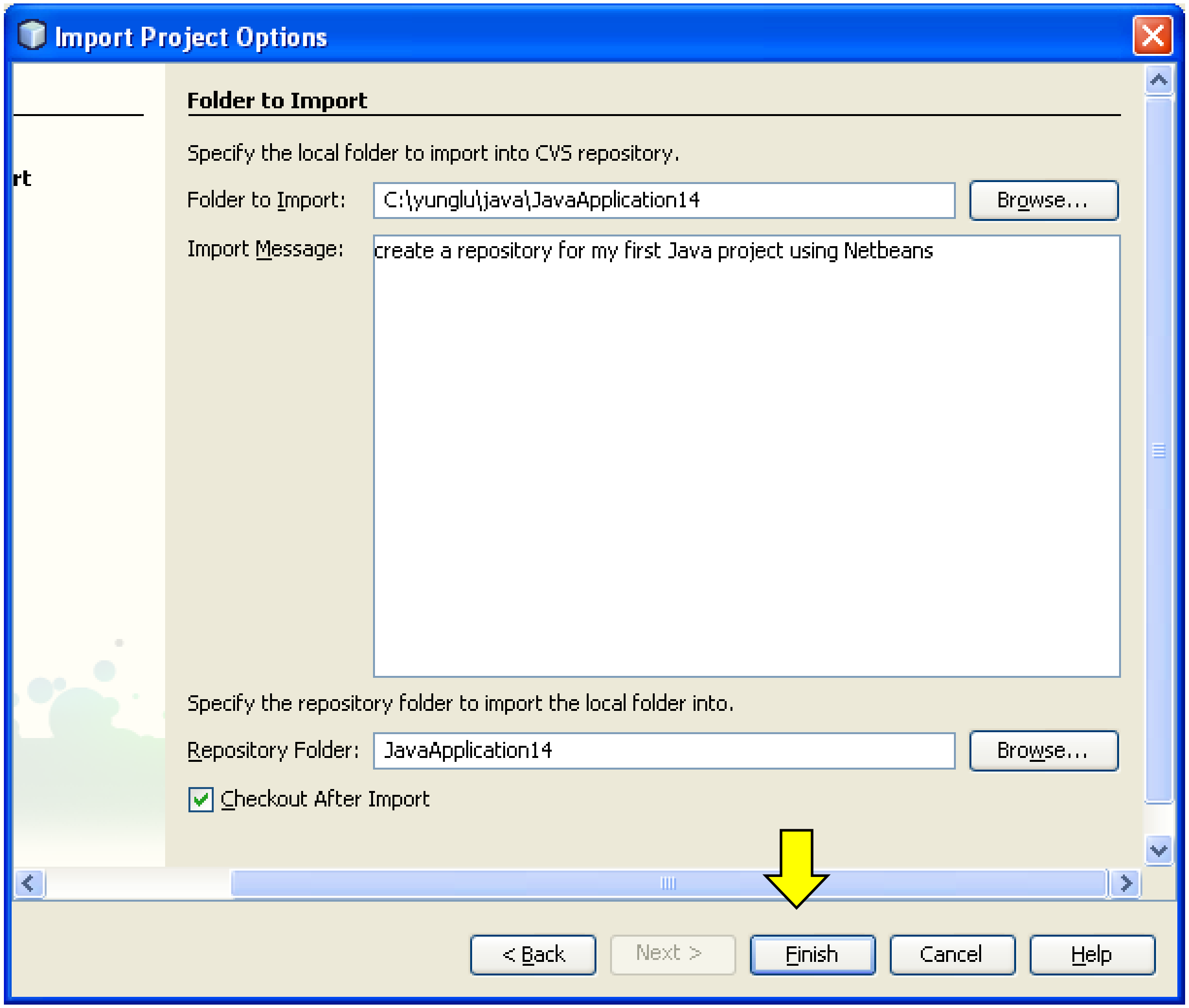
● Use Internal SSH

Password: **********

□ Remember Password

○ Use External Shell

Shell Command:



Specify the local folder to import into CVS repository.

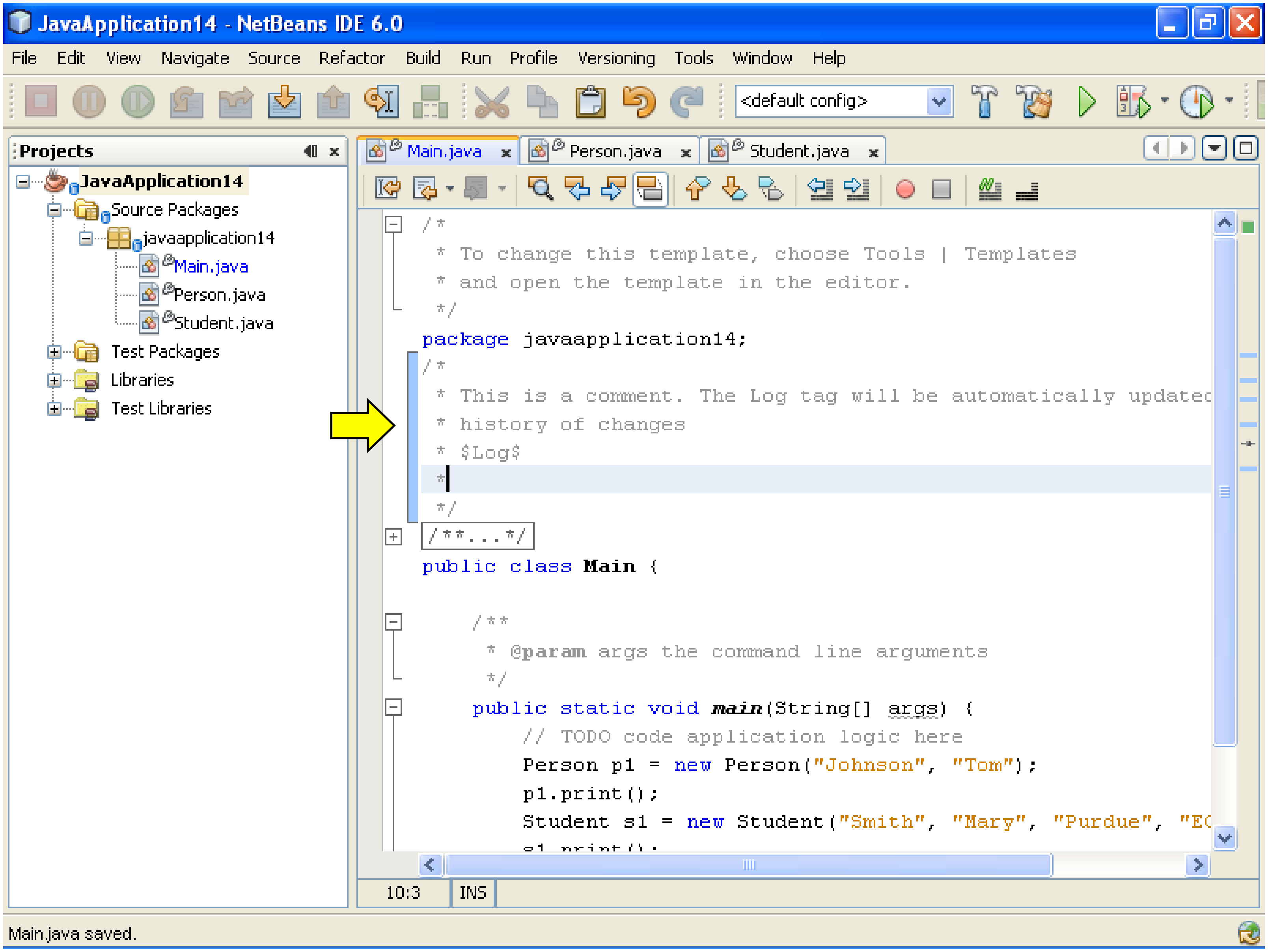
Folder to Import: C:\yunglu\java\JavaApplication14

Import Message: create a repository for my first Java project using Netbeans

Specify the repository folder to import the local folder into.

Repository Folder: JavaApplication14

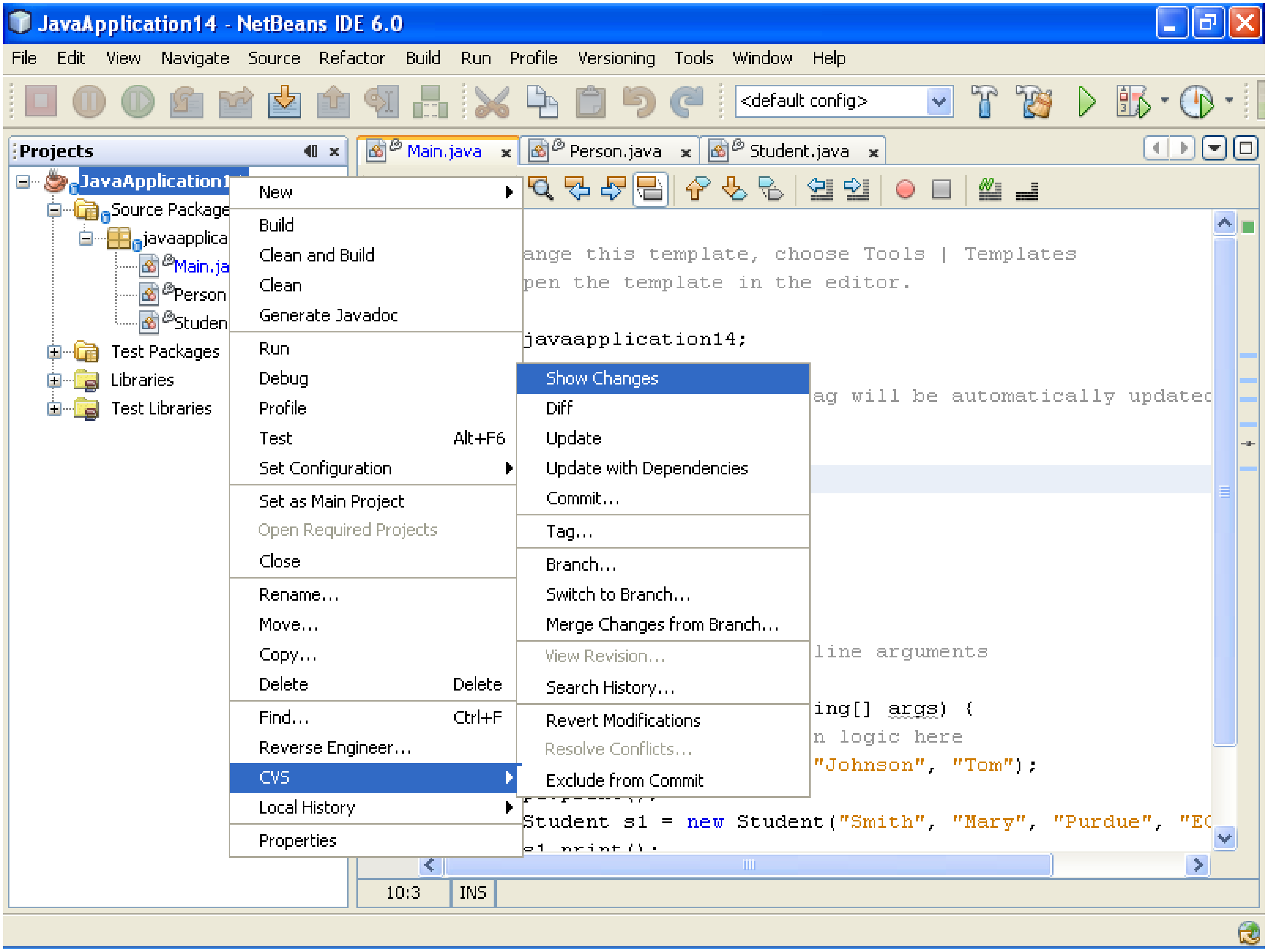
☑ Checkout After Import



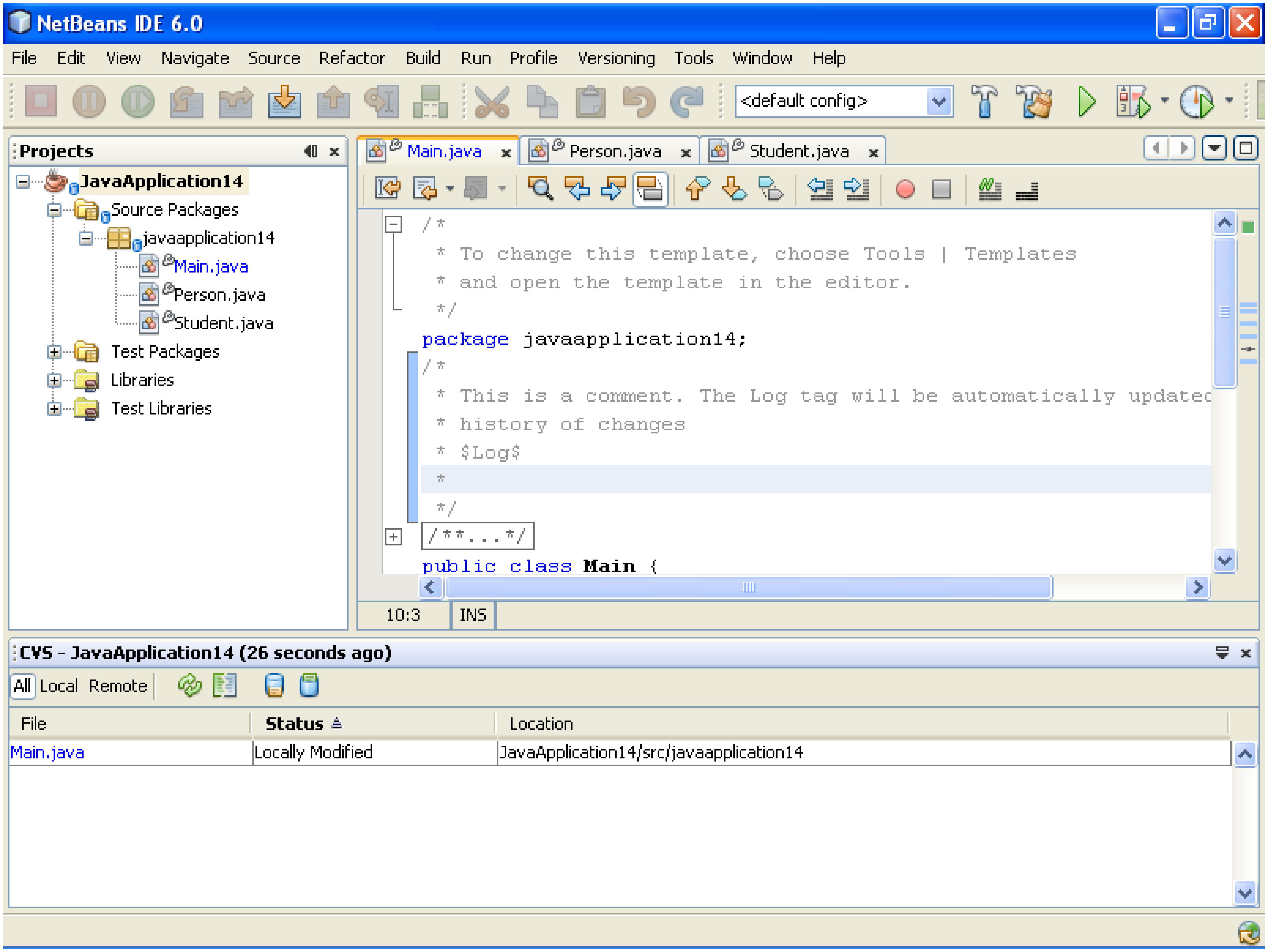
```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication14;
/*
 * This is a comment. The Log tag will be automatically updated
 * history of changes
 * $Log$
 *
 */
/**...*/
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Person p1 = new Person("Johnson", "Tom");
        p1.print();
        Student s1 = new Student("Smith", "Mary", "Purdue", "EC
        s1.print();
```

10:3 INS

Main.java saved.

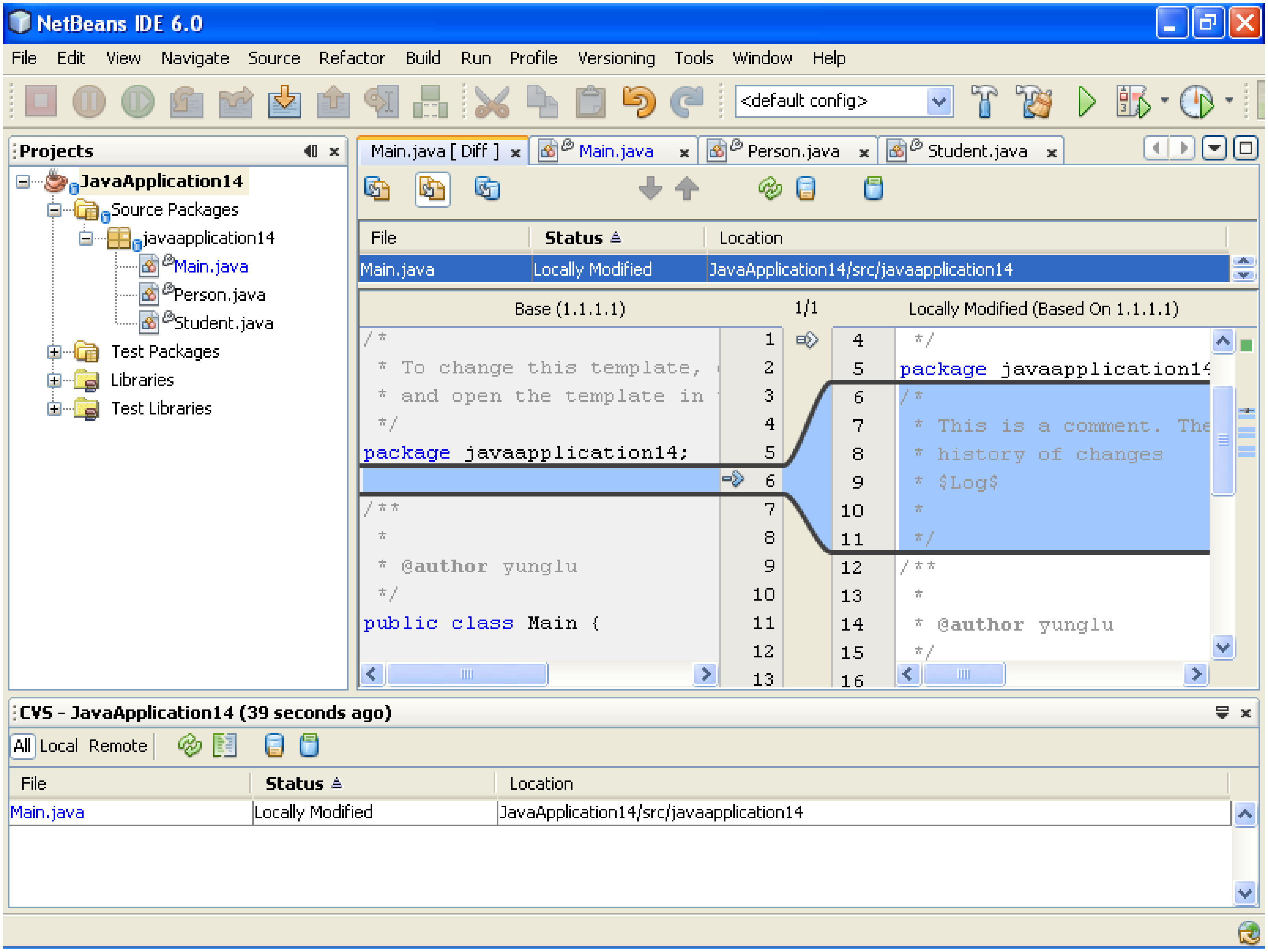


```
ange this template, choose Tools | Templates
pen the template in the editor.

javaapplication14;

ag will be automatically updated

line arguments

ing[] args) {
n logic here
"Johnson", "Tom");

Student s1 = new Student("Smith", "Mary", "Purdue", "EC
s1.print();
```



```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication14;
/*
 * This is a comment. The Log tag will be automatically updatec
 * history of changes
 * $Log$
 *
 */
/**...*/
public class Main {
```

File	Status	Location
Main.java	Locally Modified	JavaApplication14/src/javaapplication14



File	Status ≙	Location
Main.java	Locally Modified	JavaApplication14/src/javaapplication14

Base (1.1.1.1)

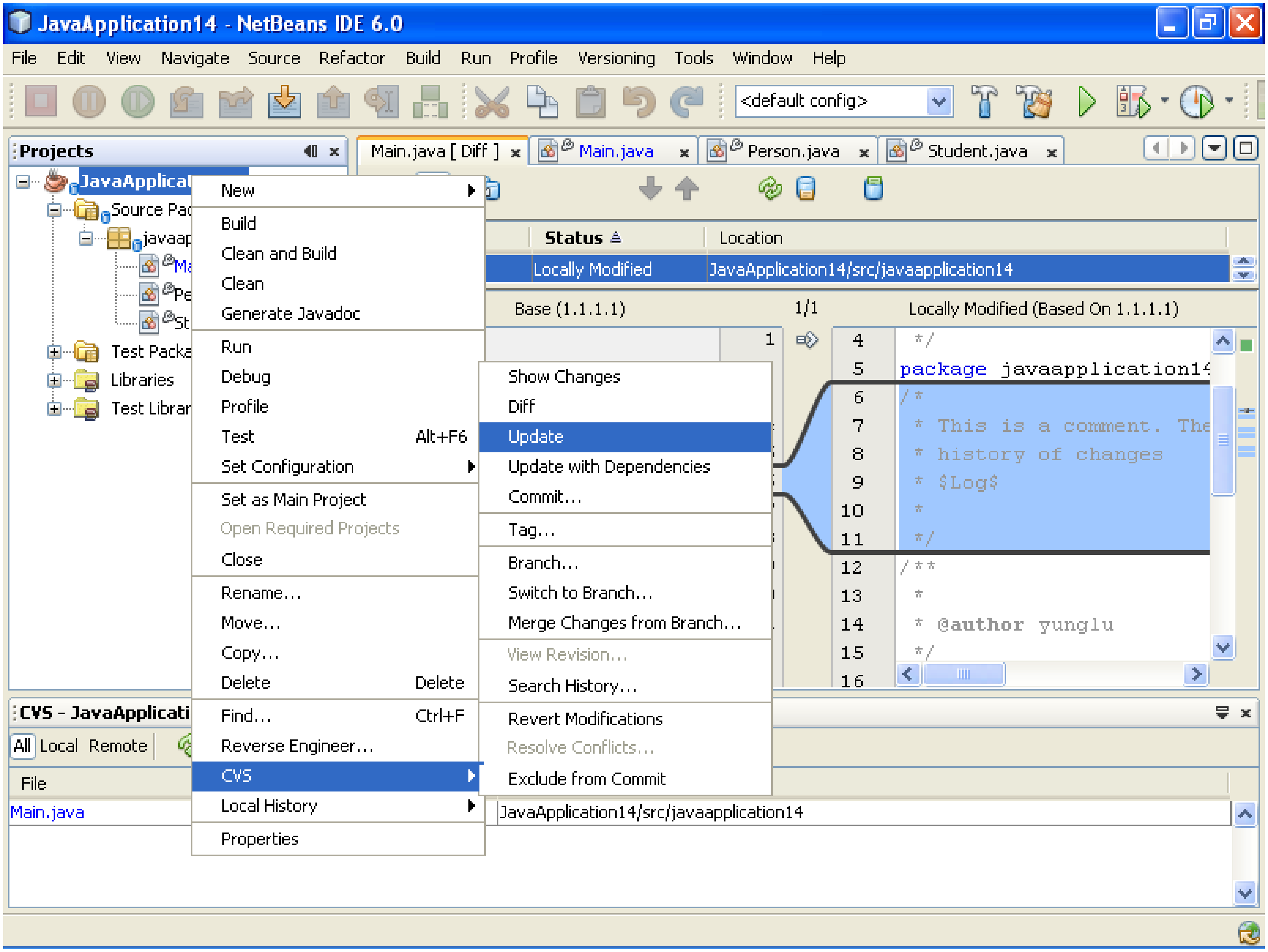
```
/*
 * To change this template,
 * and open the template in
 */
package javaapplication14;

/**
 *
 * @author yunglu
 */
public class Main {
```

Locally Modified (Based On 1.1.1.1)

```
 */
package javaapplication14
/*
 * This is a comment. The
 * history of changes
 * $Log$
 *
 */
/**
 *
 * @author yunglu
 */
```

CVS - JavaApplication14 (39 seconds ago)

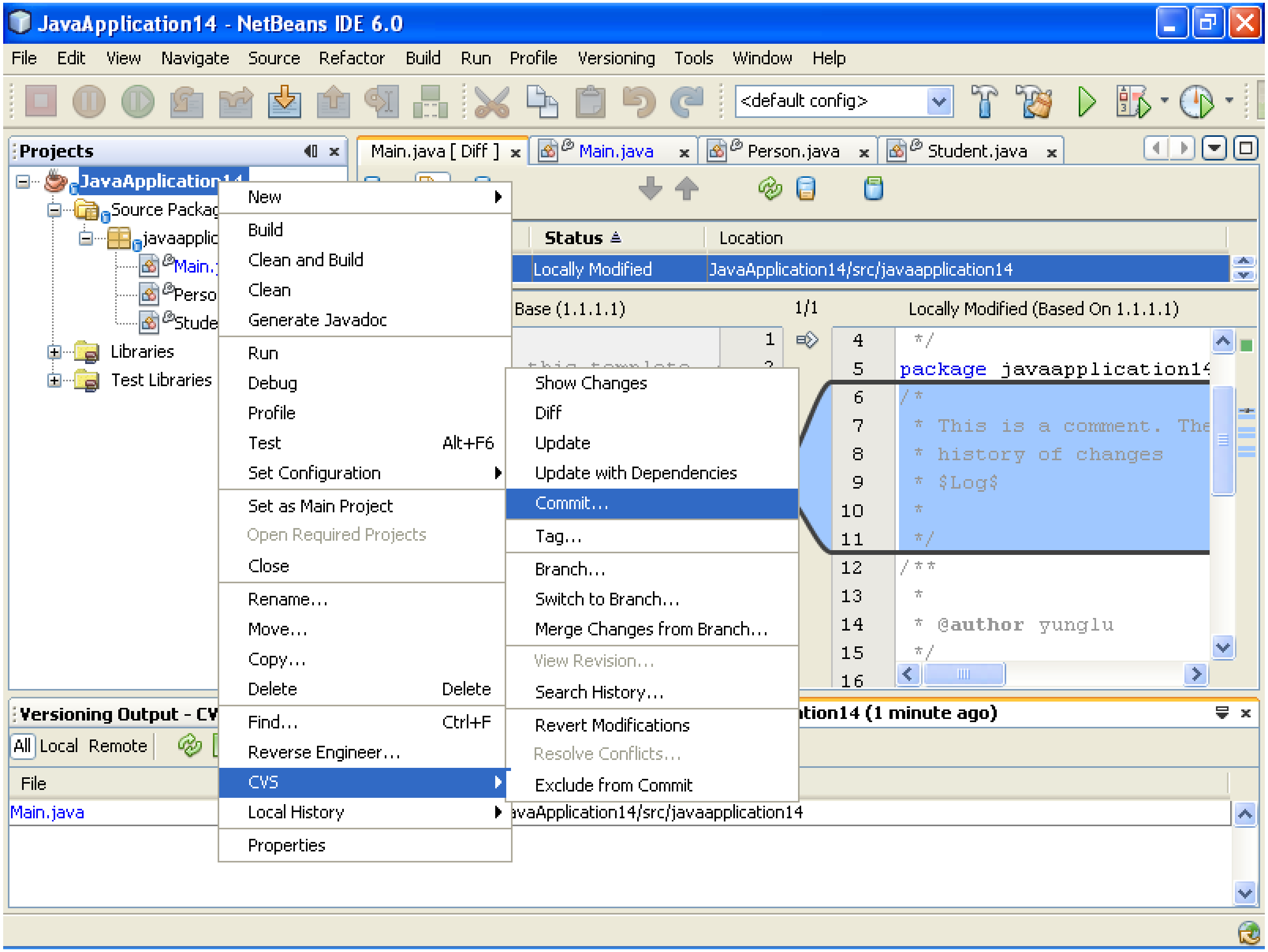
File	Status ≙	Location
Main.java	Locally Modified	JavaApplication14/src/javaapplication14



Status	Location
Locally Modified	JavaApplication14/src/javaapplication14

```
 */
package javaapplication1
/*
 * This is a comment. The
 * history of changes
 * $Log$
 *
 */
/**
 *
 * @author yunglu
 */
```

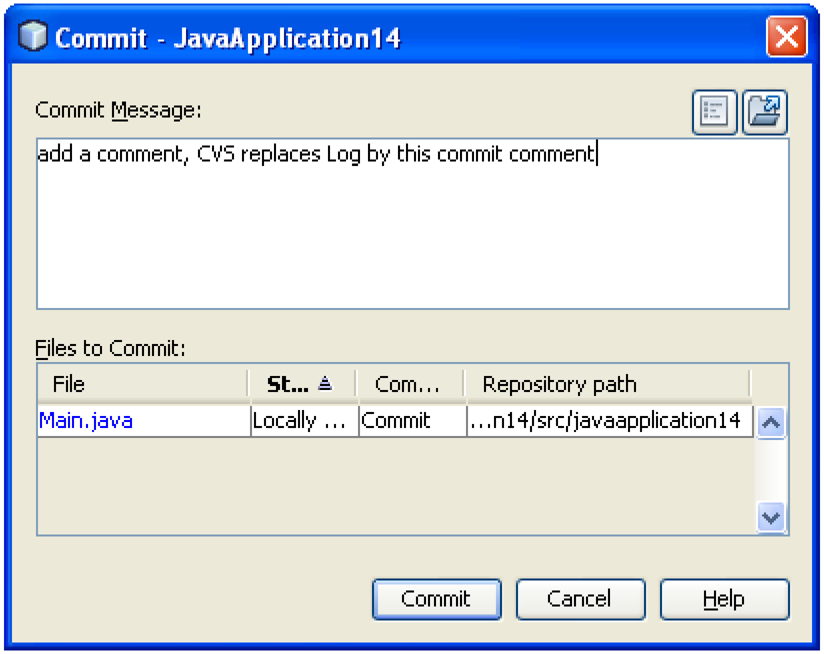
File	
Main.java	JavaApplication14/src/javaapplication14



Status	Location
Locally Modified	JavaApplication14/src/javaapplication14

```
 */
package javaapplication14
/*
 * This is a comment. The
 * history of changes
 * $Log$
 *
 */
/**
 *
 * @author yunglu
 */
```

avaApplication14/src/javaapplication14

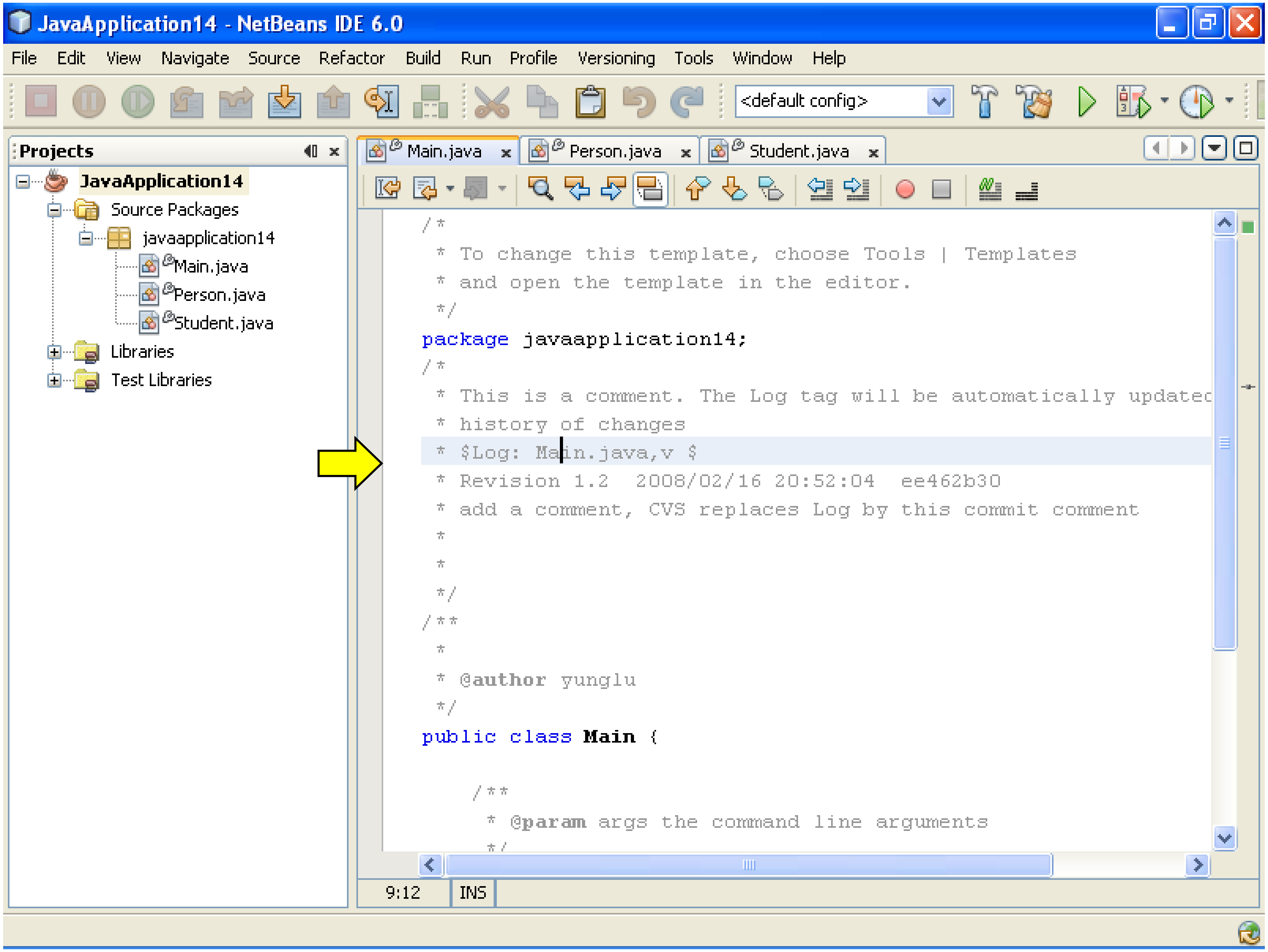


Commit Message:

add a comment, CVS replaces Log by this commit comment

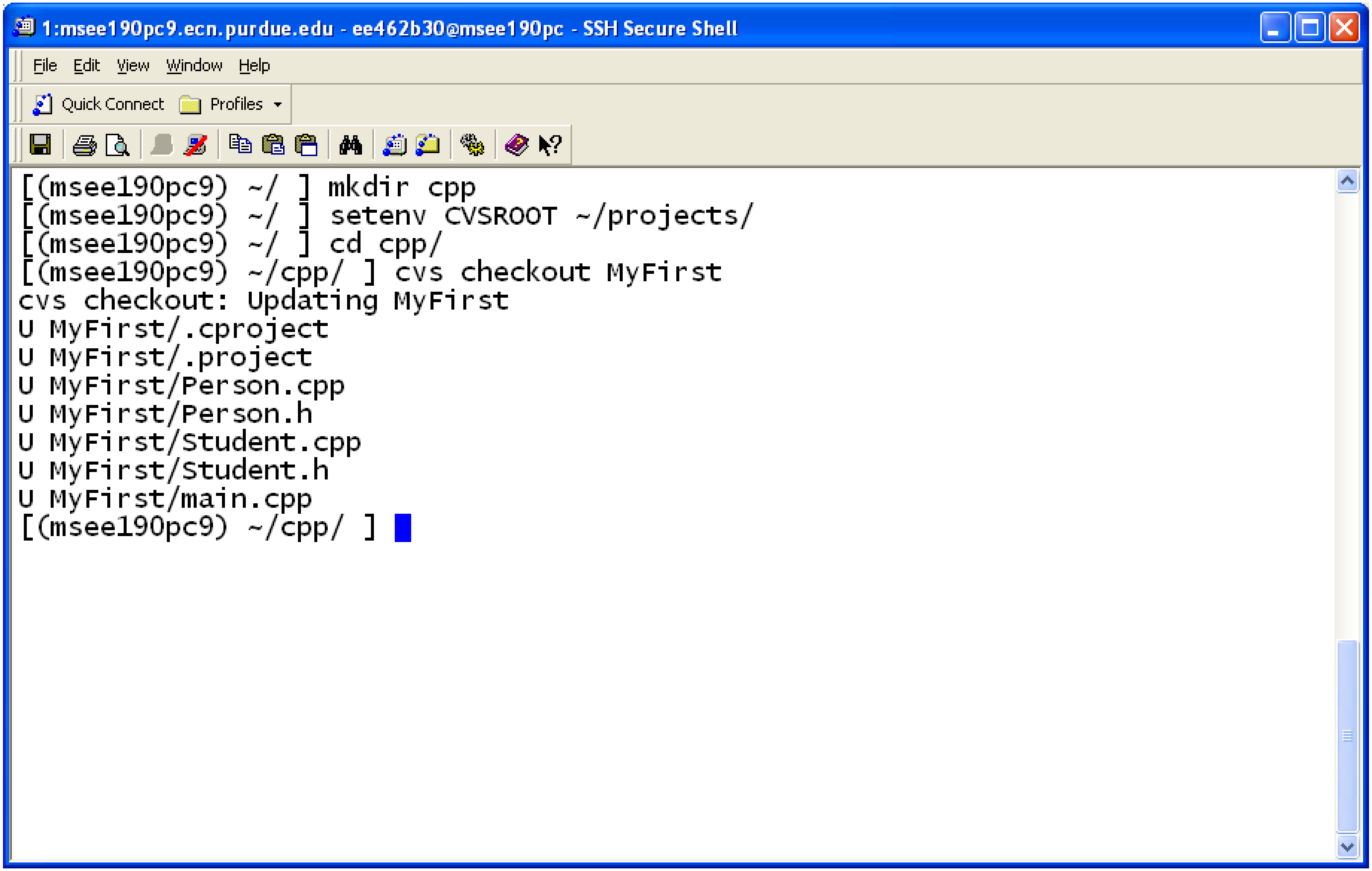
Files to Commit:

File	St...	Com...	Repository path
Main.java	Locally ...	Commit	...n14/src/javaapplication14

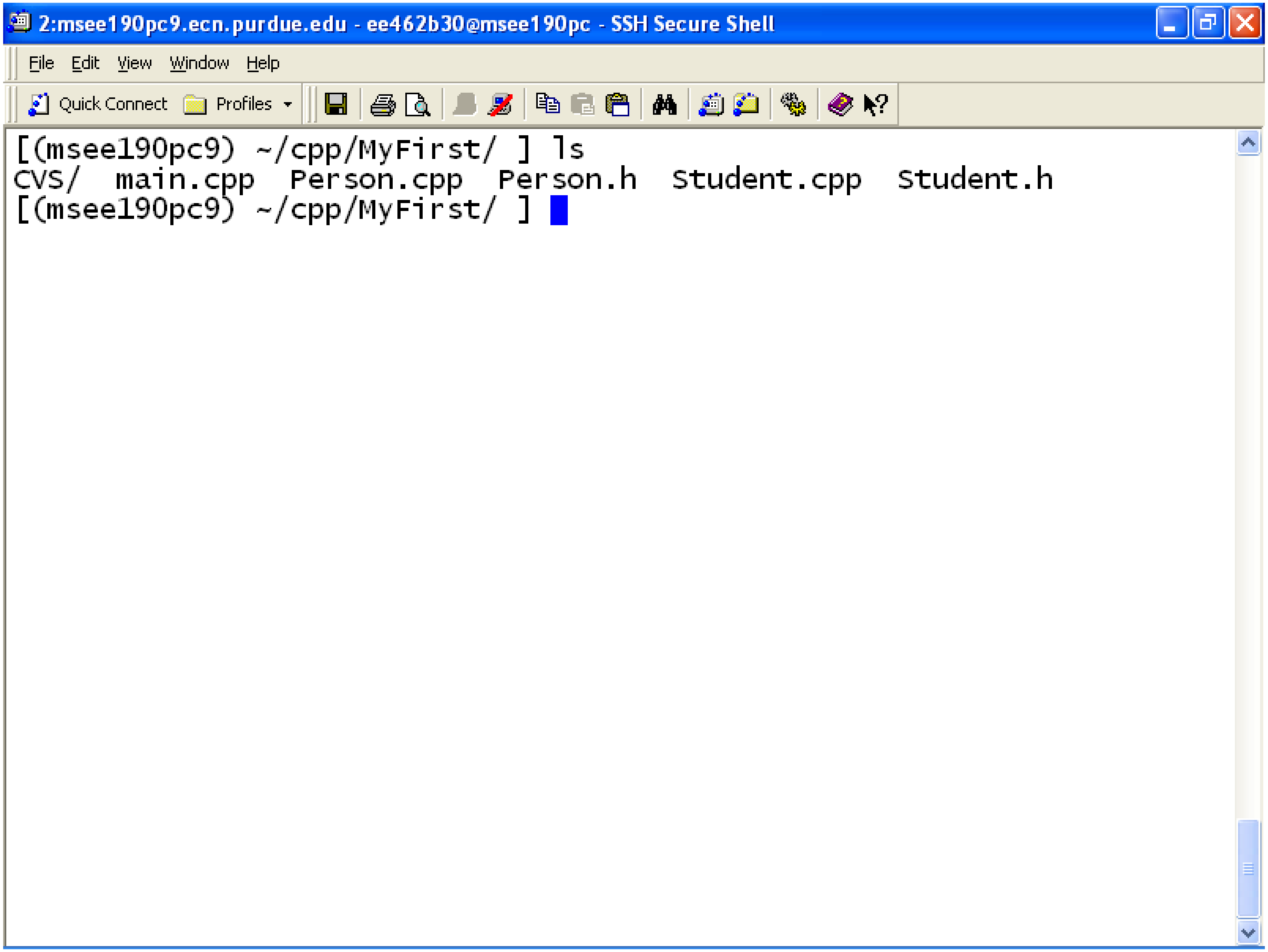


```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication14;
/*
 * This is a comment. The Log tag will be automatically updated
 * history of changes
 * $Log: Main.java,v $
 * Revision 1.2  2008/02/16 20:52:04  ee462b30
 * add a comment, CVS replaces Log by this commit comment
 *
 *
 */
/**
 *
 * @author yunglu
 */
public class Main {

    /**
     * @param args the command line arguments
     */
```

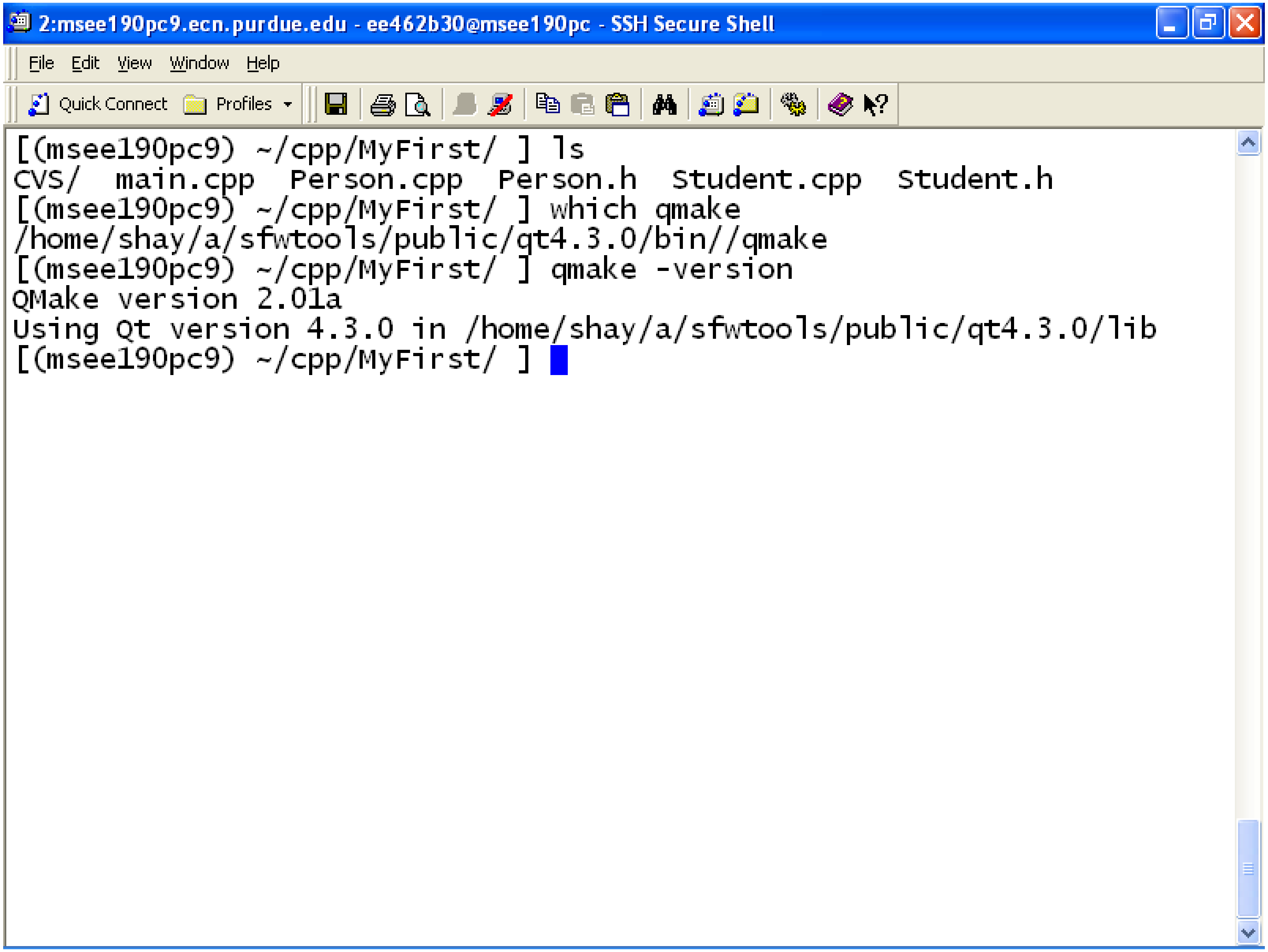
Compile / Execute C++ Programs in Linux Shell


1:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

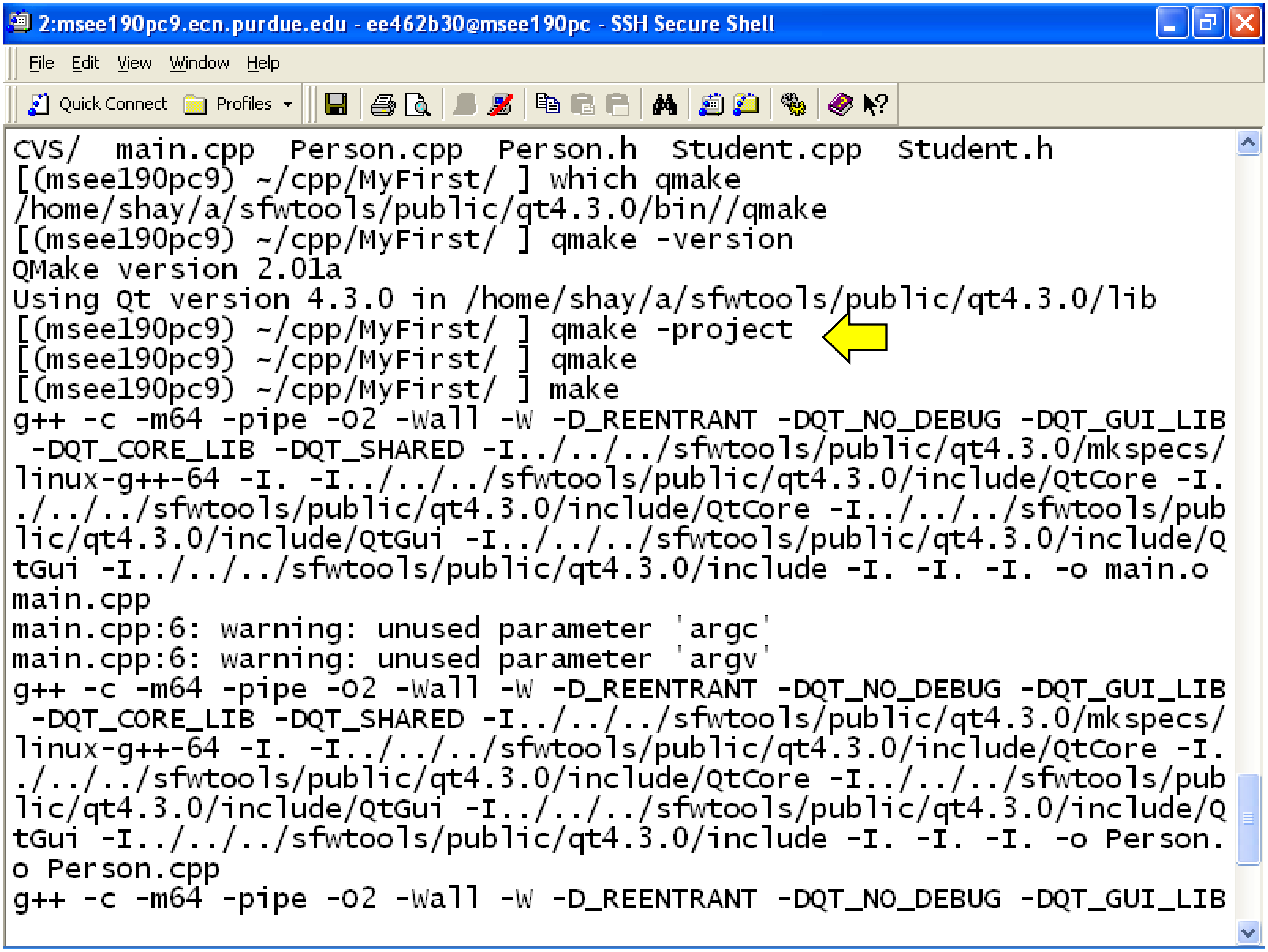
```
[(msee190pc9) ~/ ] mkdir cpp
[(msee190pc9) ~/ ] setenv CVSROOT ~/projects/
[(msee190pc9) ~/ ] cd cpp/
[(msee190pc9) ~/cpp/ ] cvs checkout MyFirst
cvs checkout: Updating MyFirst
U MyFirst/.cproject
U MyFirst/.project
U MyFirst/Person.cpp
U MyFirst/Person.h
U MyFirst/Student.cpp
U MyFirst/Student.h
U MyFirst/main.cpp
[(msee190pc9) ~/cpp/ ]
```


2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

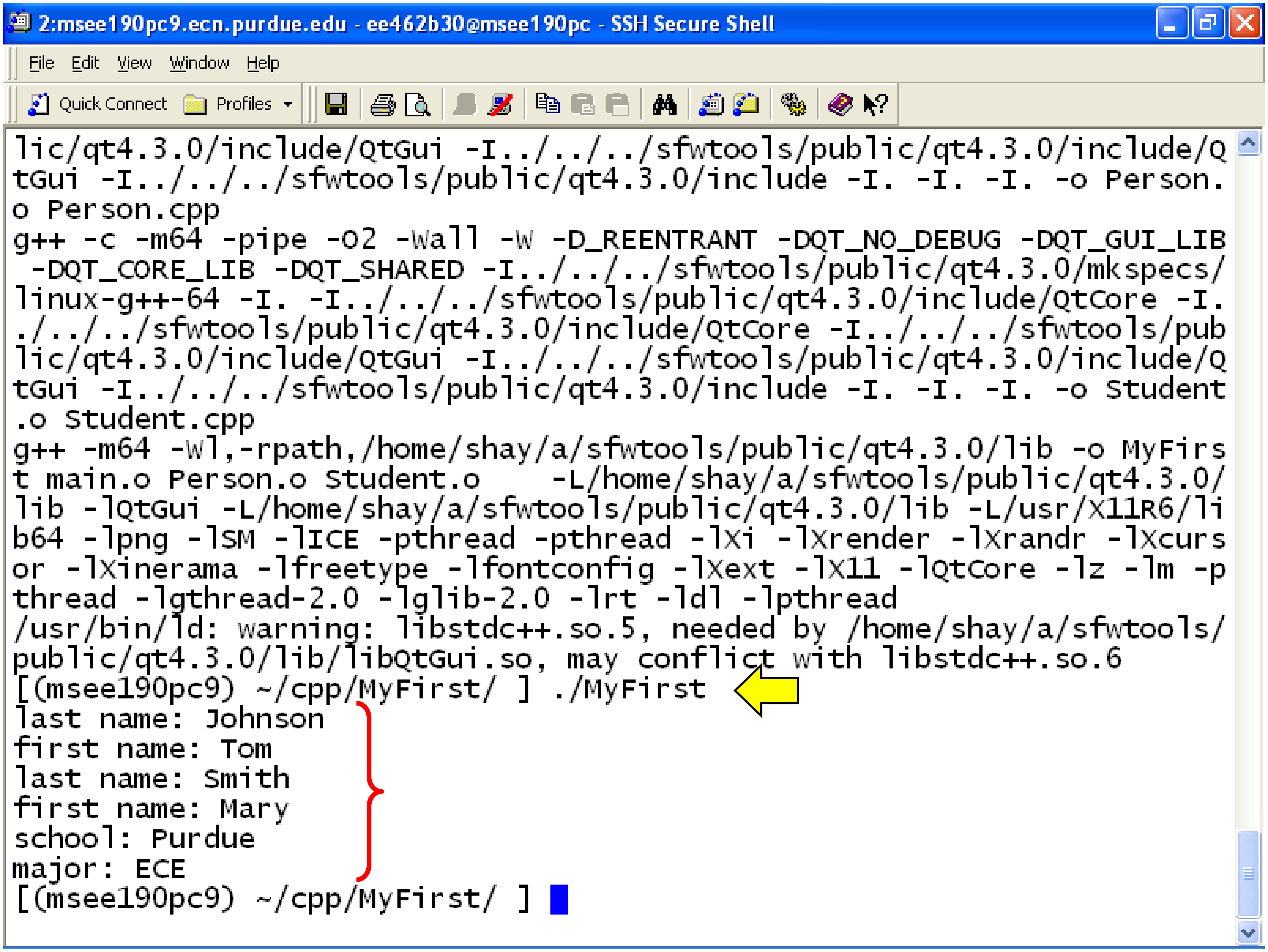
```
[(msee190pc9) ~/cpp/MyFirst/ ] ls
CVS/  main.cpp  Person.cpp  Person.h  Student.cpp  Student.h
[(msee190pc9) ~/cpp/MyFirst/ ]
```


2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
[(msee190pc9) ~/cpp/MyFirst/ ] ls
CVS/  main.cpp  Person.cpp  Person.h  Student.cpp  Student.h
[(msee190pc9) ~/cpp/MyFirst/ ] which qmake
/home/shay/a/sfwtools/public/qt4.3.0/bin//qmake
[(msee190pc9) ~/cpp/MyFirst/ ] qmake -version
QMake version 2.01a
Using Qt version 4.3.0 in /home/shay/a/sfwtools/public/qt4.3.0/lib
[(msee190pc9) ~/cpp/MyFirst/ ] 
```

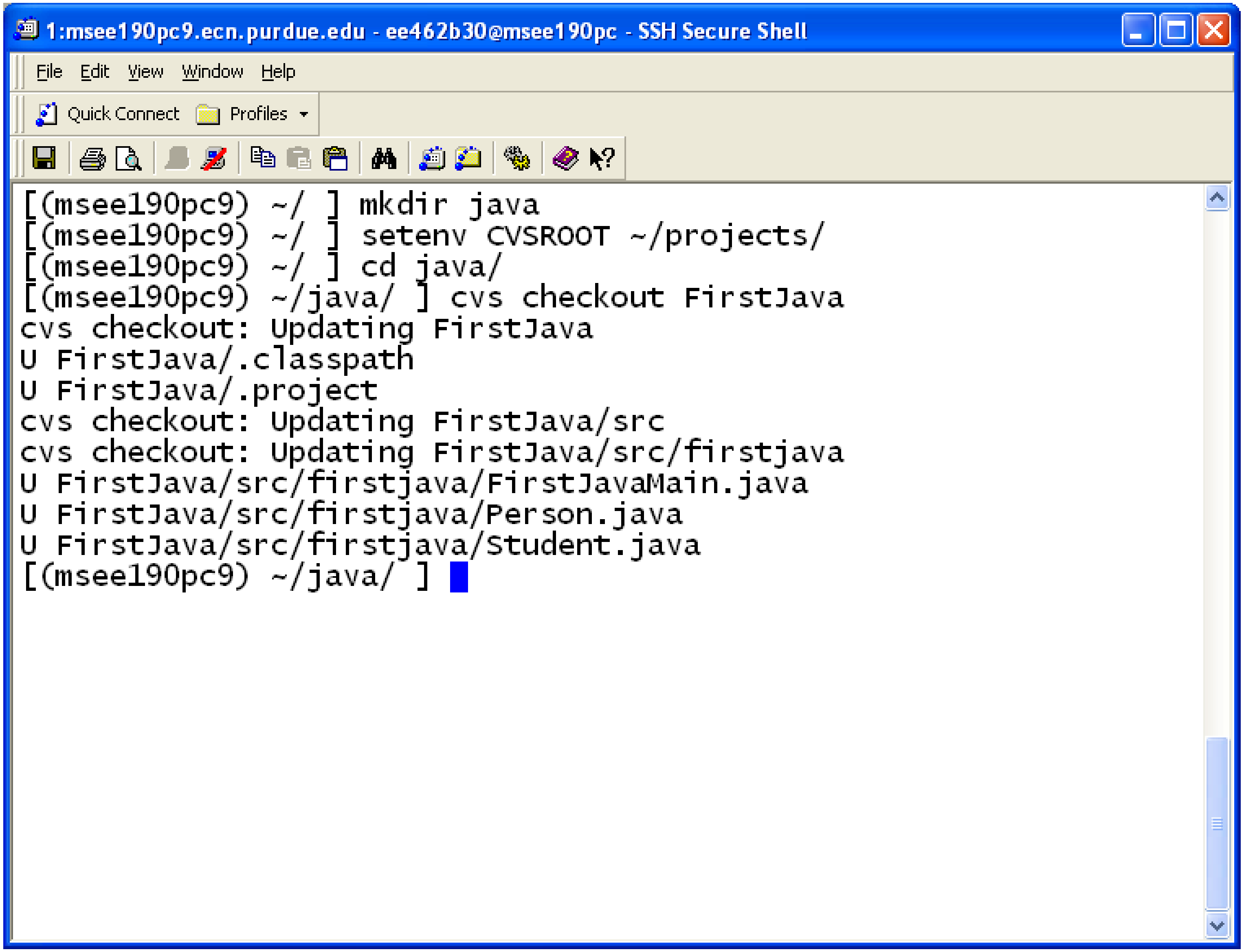

2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File
Edit
View
Window
Help
Quick Connect
Profiles

```
CVS/  main.cpp  Person.cpp  Person.h  Student.cpp  Student.h
[(msee190pc9) ~/cpp/MyFirst/ ] which qmake
/home/shay/a/sfwtools/public/qt4.3.0/bin//qmake
[(msee190pc9) ~/cpp/MyFirst/ ] qmake -version
QMake version 2.01a
Using Qt version 4.3.0 in /home/shay/a/sfwtools/public/qt4.3.0/lib
[(msee190pc9) ~/cpp/MyFirst/ ] qmake -project
[(msee190pc9) ~/cpp/MyFirst/ ] qmake
[(msee190pc9) ~/cpp/MyFirst/ ] make
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB
 -DQT_CORE_LIB -DQT_SHARED -I../../../sfwtools/public/qt4.3.0/mkspecs/
linux-g++-64 -I. -I../../../sfwtools/public/qt4.3.0/include/QtCore -I.
./../../sfwtools/public/qt4.3.0/include/QtCore -I../../../sfwtools/pub
lic/qt4.3.0/include/QtGui -I../../../sfwtools/public/qt4.3.0/include/Q
tGui -I../../../sfwtools/public/qt4.3.0/include -I. -I. -I. -o main.o
main.cpp
main.cpp:6: warning: unused parameter 'argc'
main.cpp:6: warning: unused parameter 'argv'
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB
 -DQT_CORE_LIB -DQT_SHARED -I../../../sfwtools/public/qt4.3.0/mkspecs/
linux-g++-64 -I. -I../../../sfwtools/public/qt4.3.0/include/QtCore -I.
./../../sfwtools/public/qt4.3.0/include/QtCore -I../../../sfwtools/pub
lic/qt4.3.0/include/QtGui -I../../../sfwtools/public/qt4.3.0/include/Q
tGui -I../../../sfwtools/public/qt4.3.0/include -I. -I. -I. -o Person.
o Person.cpp
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB
```


2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File
Edit
View
Window
Help
Quick Connect
Profiles

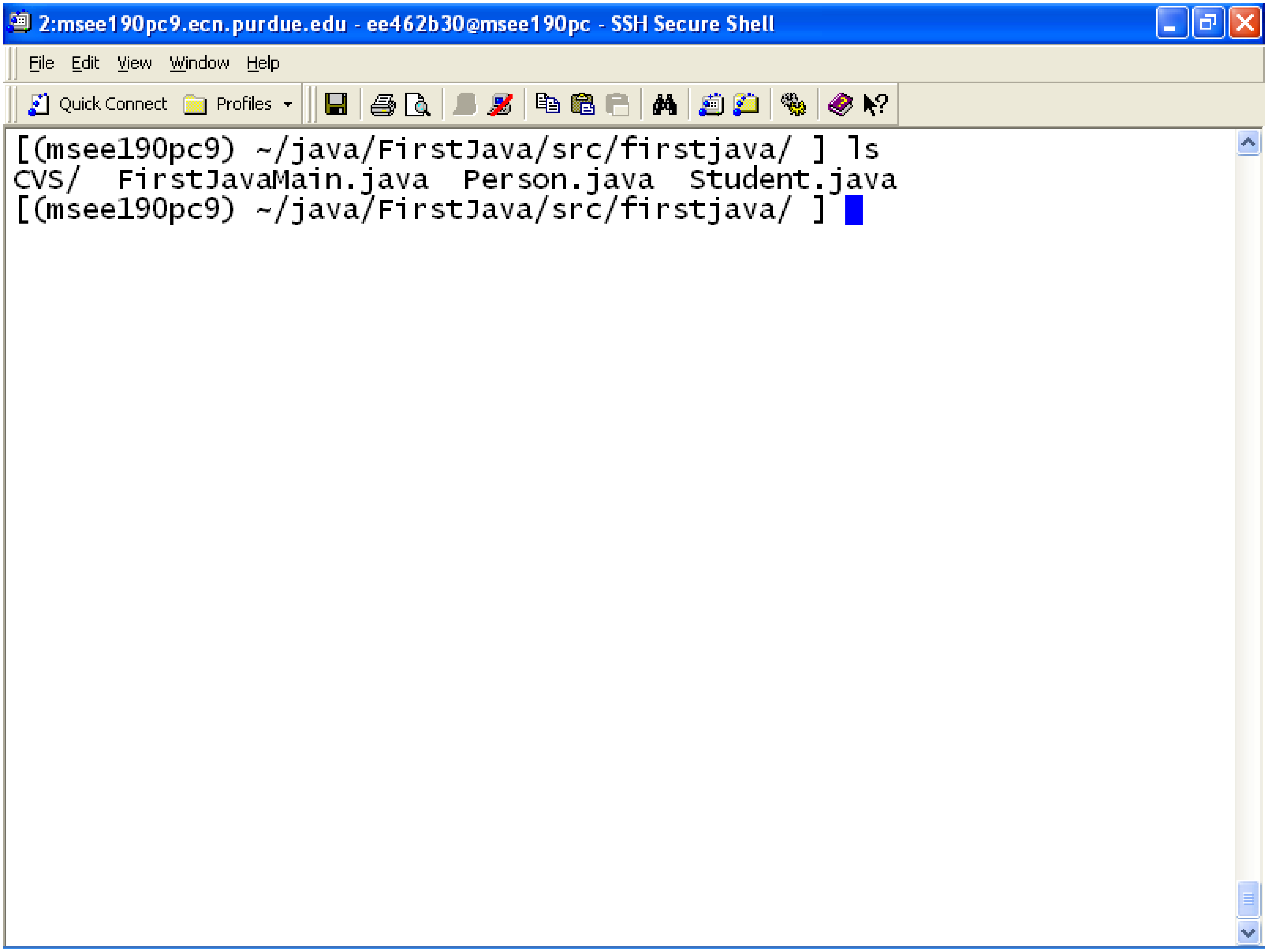
```
lic/qt4.3.0/include/QtGui -I../../../sfwtools/public/qt4.3.0/include/Q
tGui -I../../../sfwtools/public/qt4.3.0/include -I. -I. -I. -o Person.
o Person.cpp
g++ -c -m64 -pipe -O2 -Wall -W -D_REENTRANT -DQT_NO_DEBUG -DQT_GUI_LIB
 -DQT_CORE_LIB -DQT_SHARED -I../../../sfwtools/public/qt4.3.0/mkspecs/
linux-g++-64 -I. -I../../../sfwtools/public/qt4.3.0/include/QtCore -I.
./../../sfwtools/public/qt4.3.0/include/QtCore -I../../../sfwtools/pub
lic/qt4.3.0/include/QtGui -I../../../sfwtools/public/qt4.3.0/include/Q
tGui -I../../../sfwtools/public/qt4.3.0/include -I. -I. -I. -o Student
.o Student.cpp
g++ -m64 -Wl,-rpath,/home/shay/a/sfwtools/public/qt4.3.0/lib -o MyFirs
t main.o Person.o Student.o    -L/home/shay/a/sfwtools/public/qt4.3.0/
lib -lQtGui -L/home/shay/a/sfwtools/public/qt4.3.0/lib -L/usr/X11R6/li
b64 -lpng -lSM -lICE -pthread -pthread -lXi -lXrender -lXrandr -lXcurs
or -lXinerama -lfreetype -lfontconfig -lXext -lX11 -lQtCore -lz -lm -p
thread -lgthread-2.0 -lglib-2.0 -lrt -ldl -lpthread
/usr/bin/ld: warning: libstdc++.so.5, needed by /home/shay/a/sfwtools/
public/qt4.3.0/lib/libQtGui.so, may conflict with libstdc++.so.6
[(msee190pc9) ~/cpp/MyFirst/ ] ./MyFirst
last name: Johnson
first name: Tom
last name: Smith
first name: Mary
school: Purdue
major: ECE
[(msee190pc9) ~/cpp/MyFirst/ ]
```

Compile / Execute Java Programs in Linux Shell



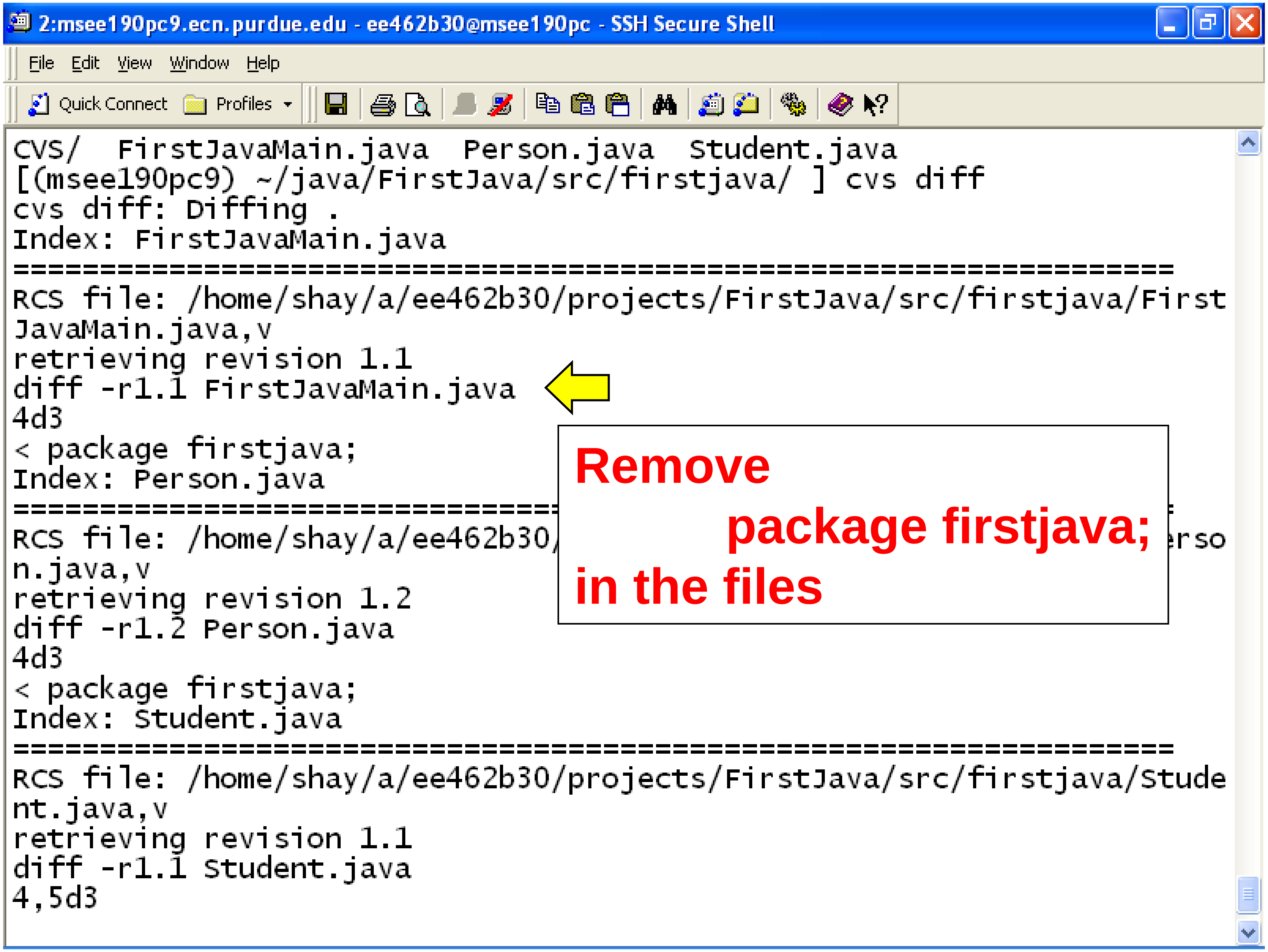
1:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
[(msee190pc9) ~/ ] mkdir java
[(msee190pc9) ~/ ] setenv CVSROOT ~/projects/
[(msee190pc9) ~/ ] cd java/
[(msee190pc9) ~/java/ ] cvs checkout FirstJava
cvs checkout: Updating FirstJava
U FirstJava/.classpath
U FirstJava/.project
cvs checkout: Updating FirstJava/src
cvs checkout: Updating FirstJava/src/firstjava
U FirstJava/src/firstjava/FirstJavaMain.java
U FirstJava/src/firstjava/Person.java
U FirstJava/src/firstjava/Student.java
[(msee190pc9) ~/java/ ] 
```



2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] ls
CVS/  FirstJavaMain.java  Person.java  Student.java
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ]
```

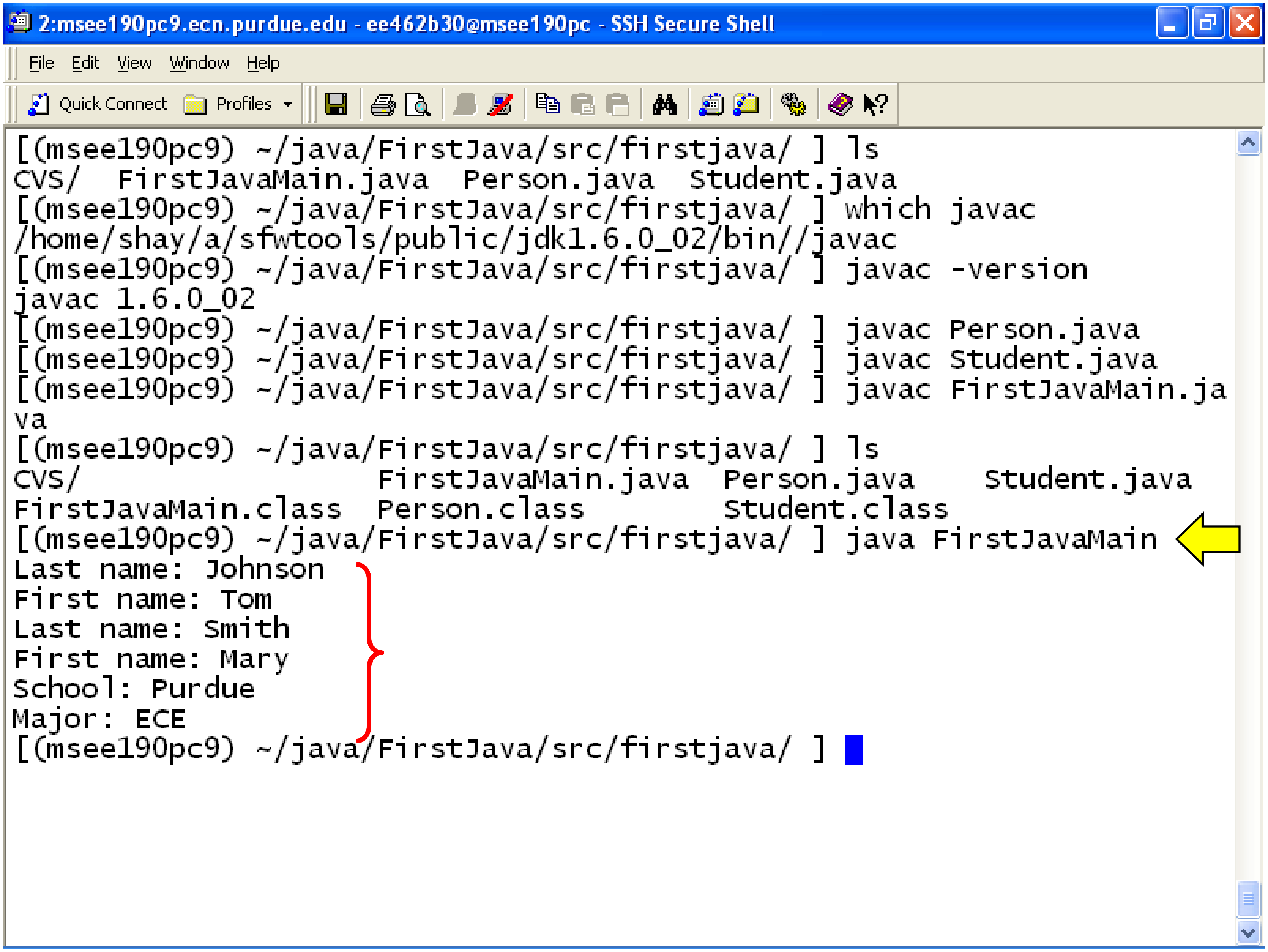

2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
CVS/  FirstJavaMain.java  Person.java  Student.java
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] cvs diff
cvs diff: Diffing .
Index: FirstJavaMain.java
====================================================================
RCS file: /home/shay/a/ee462b30/projects/FirstJava/src/firstjava/First
JavaMain.java,v
retrieving revision 1.1
diff -r1.1 FirstJavaMain.java
4d3
< package firstjava;
Index: Person.java
================================                                   =
RCS file: /home/shay/a/ee462b30/                                  erso
n.java,v
retrieving revision 1.2
diff -r1.2 Person.java
4d3
< package firstjava;
Index: Student.java
====================================================================
RCS file: /home/shay/a/ee462b30/projects/FirstJava/src/firstjava/Stude
nt.java,v
retrieving revision 1.1
diff -r1.1 Student.java
4,5d3
```

Remove

package firstjava;

in the files


2:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles

```
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] ls
CVS/  FirstJavaMain.java  Person.java  Student.java
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] which javac
/home/shay/a/sfwtools/public/jdk1.6.0_02/bin//javac
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] javac -version
javac 1.6.0_02
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] javac Person.java
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] javac Student.java
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] javac FirstJavaMain.ja
va
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] ls
CVS/                    FirstJavaMain.java  Person.java    Student.java
FirstJavaMain.class     Person.class        Student.class
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ] java FirstJavaMain
Last name: Johnson
First name: Tom
Last name: Smith
First name: Mary
School: Purdue
Major: ECE
[(msee190pc9) ~/java/FirstJava/src/firstjava/ ]
```

Self Test