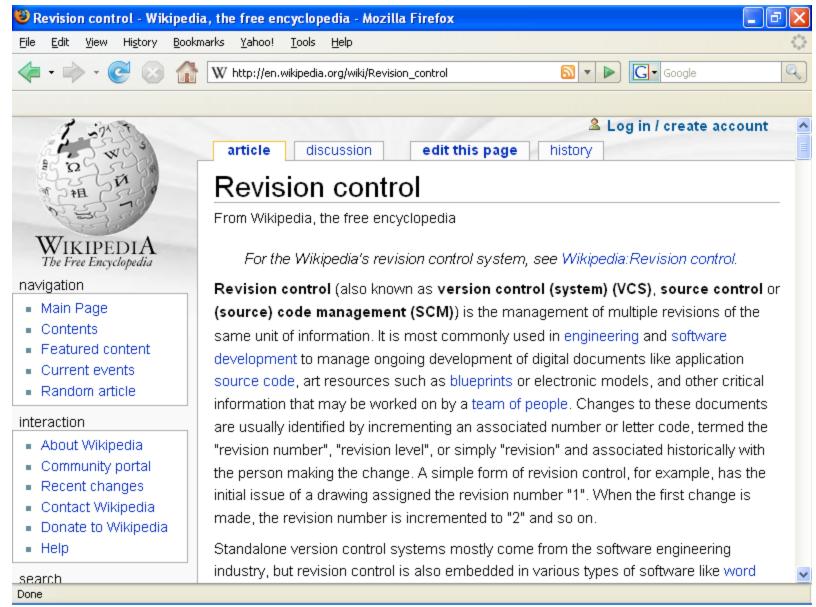
ECE 462 Object-Oriented Programming using C++ and Java

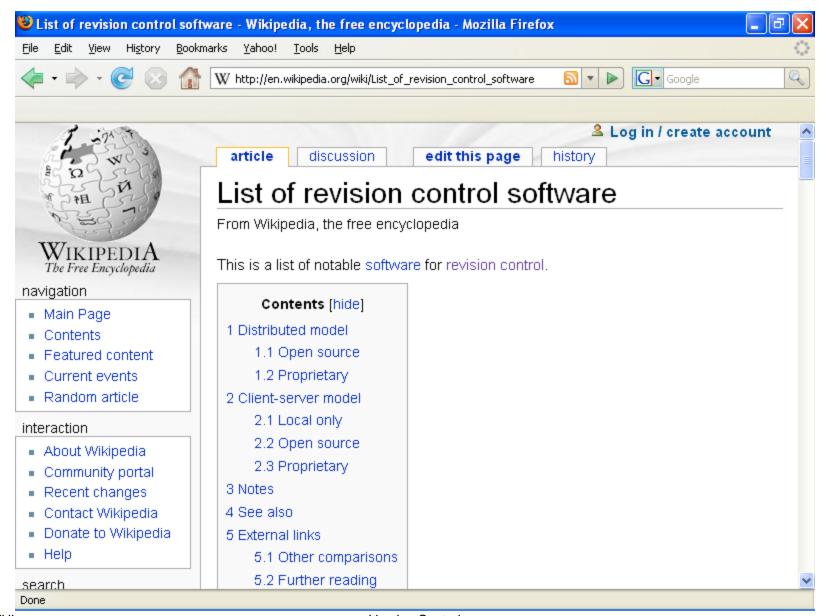
Version Control

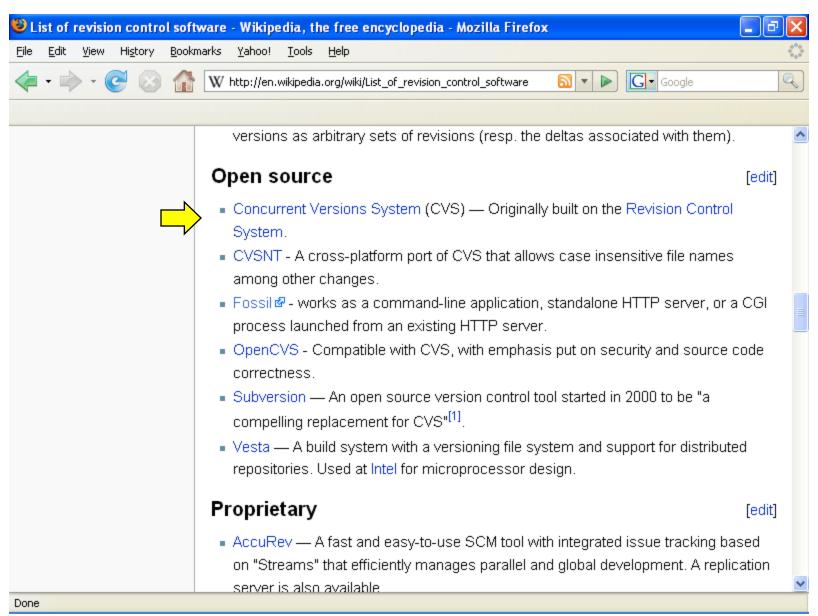
Yung-Hsiang Lu yunglu@purdue.edu

Why Version Control?

- Every software developer must understand version control.
- Version control is a natural way to create backup and much better than creating zip files
 - compare the differences between versions
 - allow concurrent development (people work simultaneously)
 - support development through networks
 - record the contributions by each person
- Version control (CVS) is required in this course.

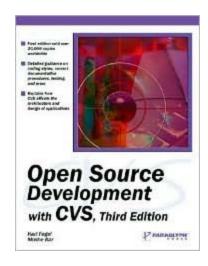


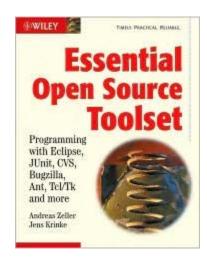


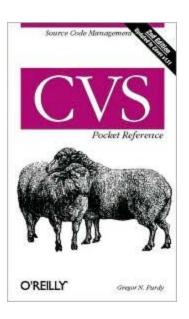


Concurrent Version System (CVS)

- one of the most widely used version control tools
- arguably the foundation of today's open source software
- available in most development tools

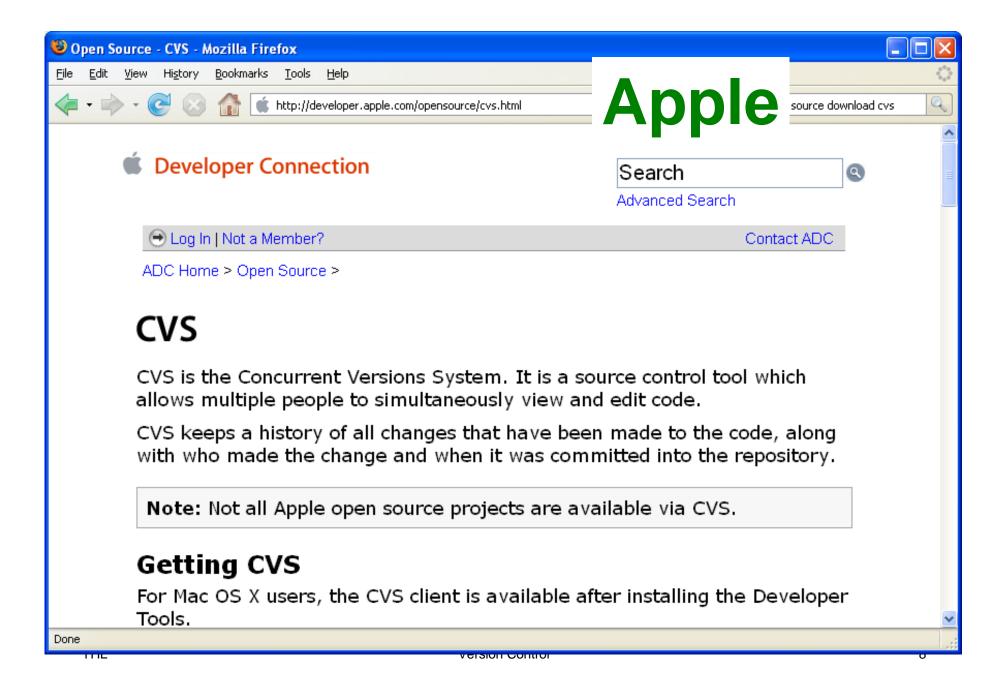






Where is CVS used?

CVS is used in thousands of projects, for examples





Source Download

Review the license (version 1.0).

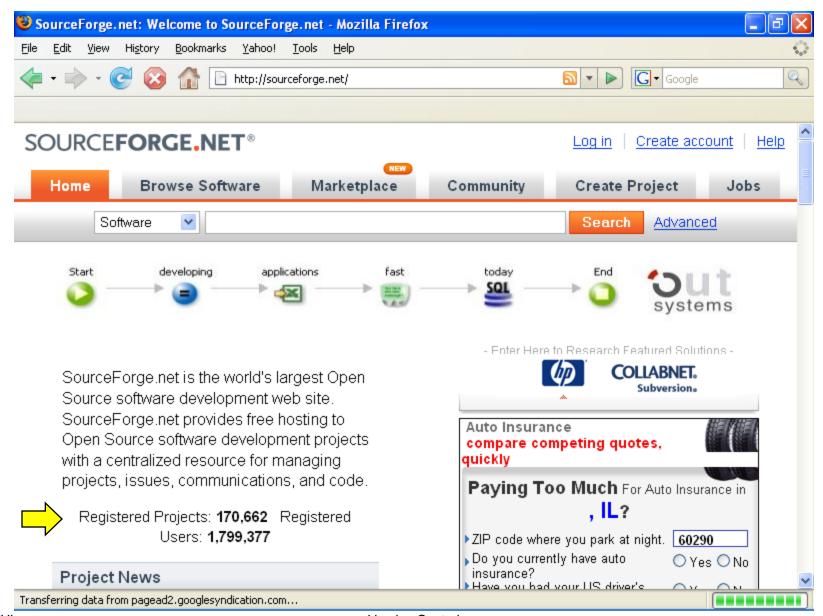
OpenDx 4.2.0 is the latest (20 May 2002) tarball.

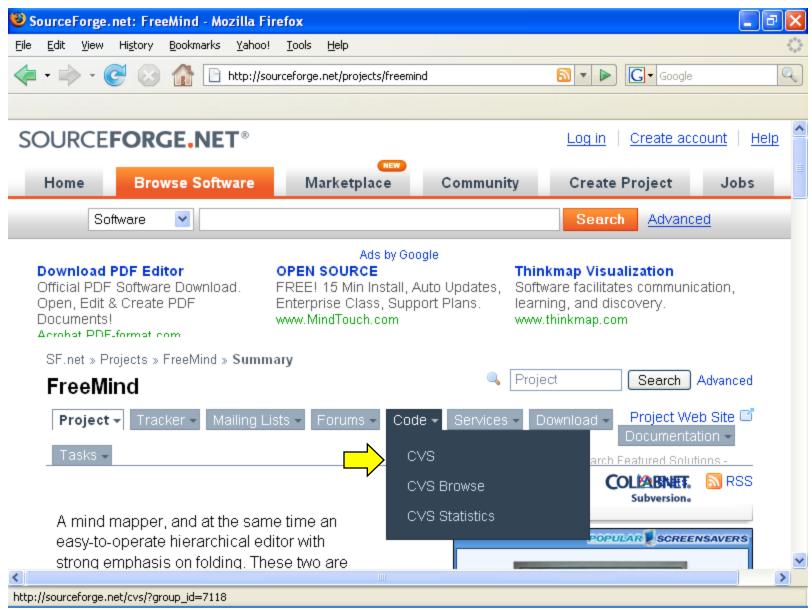
OpenDxSamples 4.2.0 contains the latest samples.

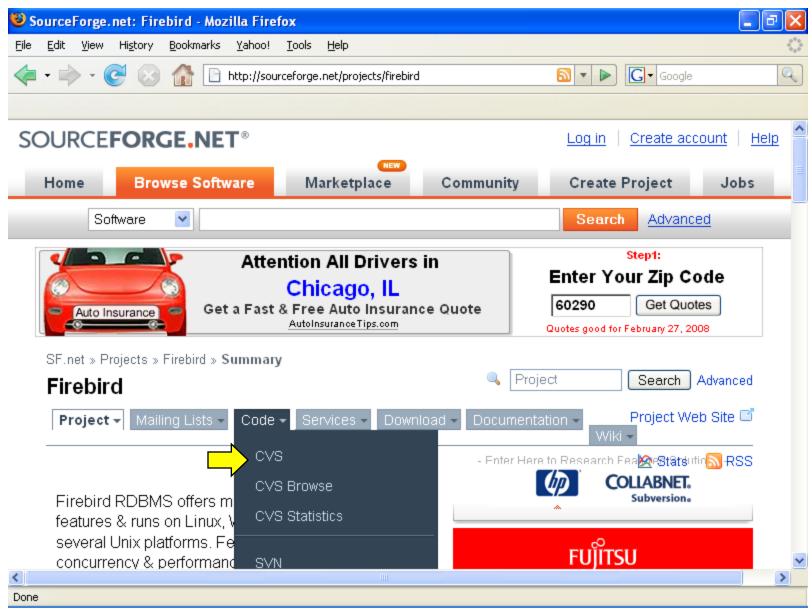
Browse the tarball download directory (this is not the cvs download directory mentioned below).

OpenDx CVS contains the current project source.

Prebuilt versions of OpenDx.





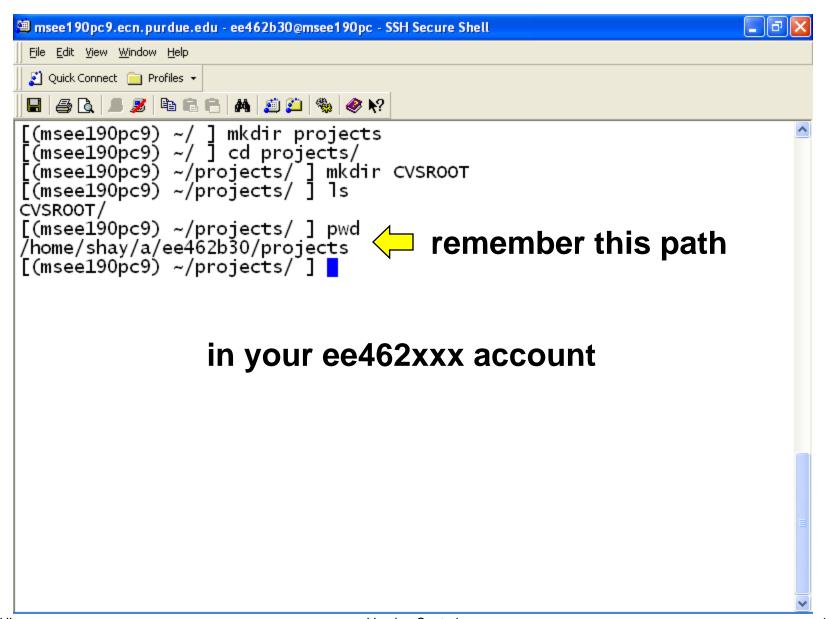




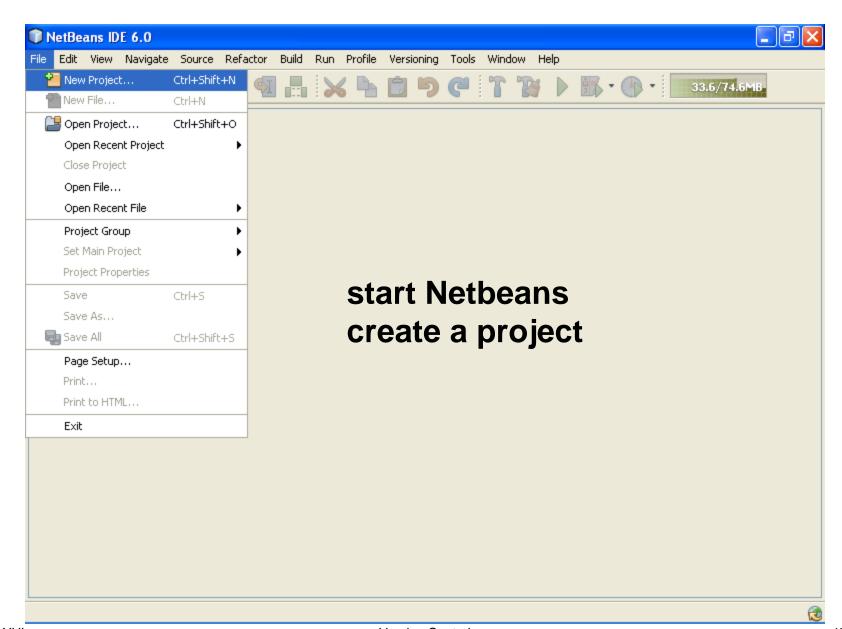
All programming assignments and lab exercises must be submitted using CVS

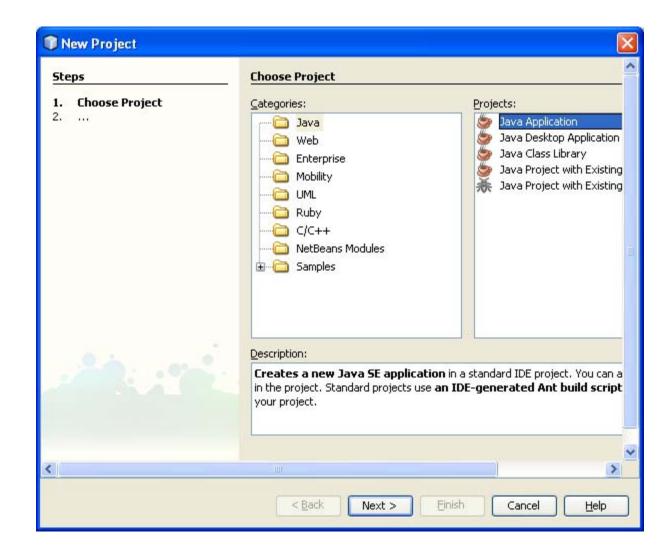
Create a CVS Repository

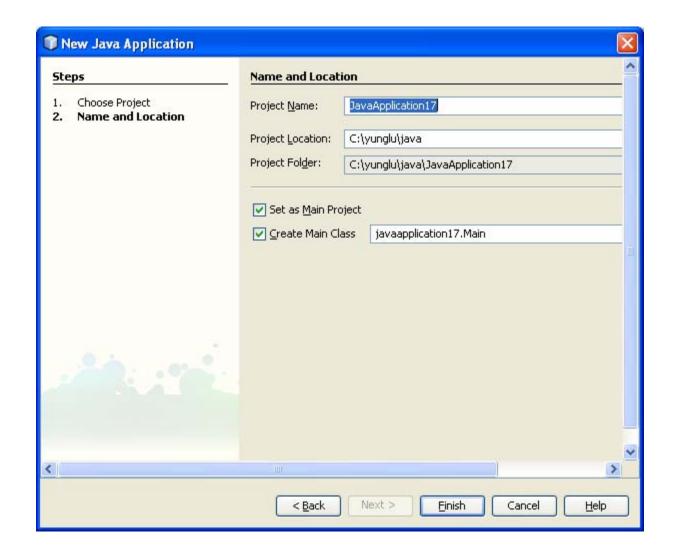
Need to do this only once.

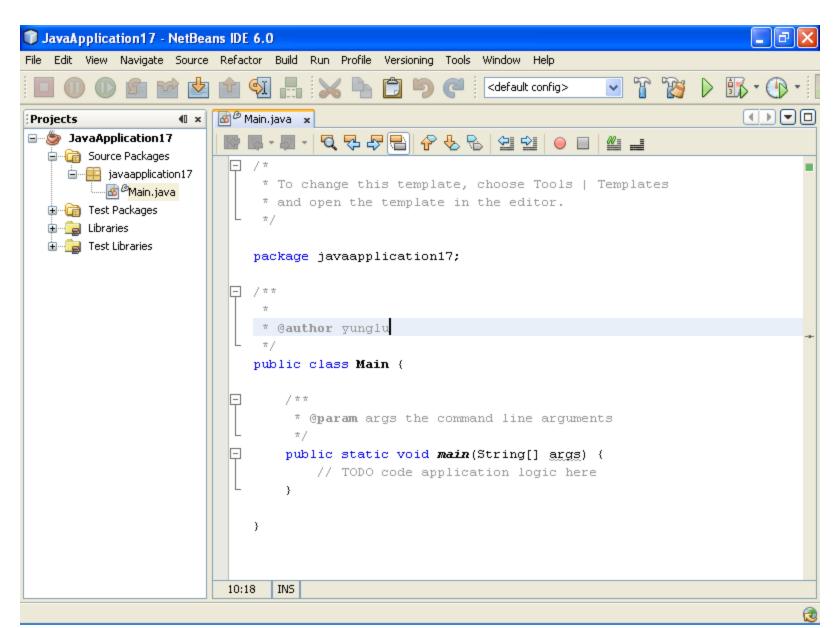


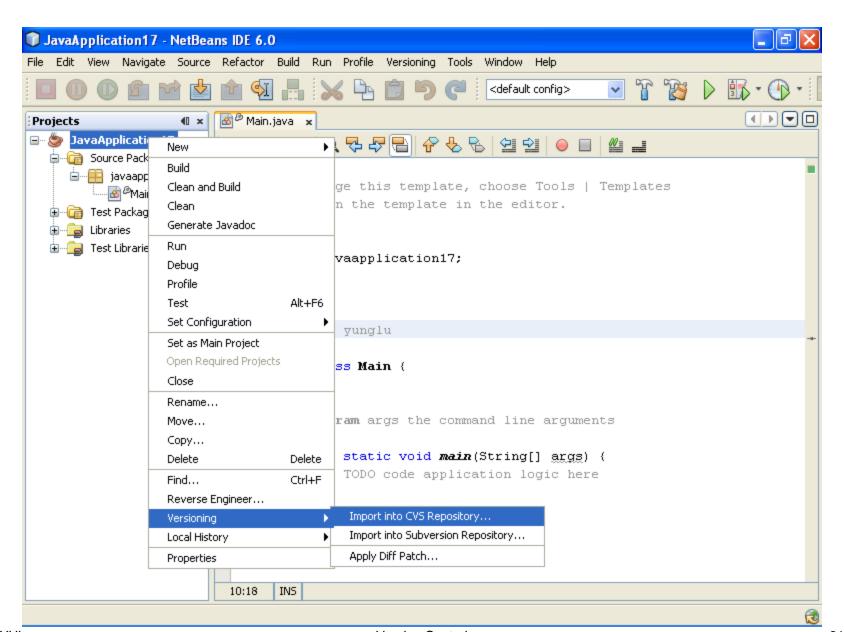
YHL

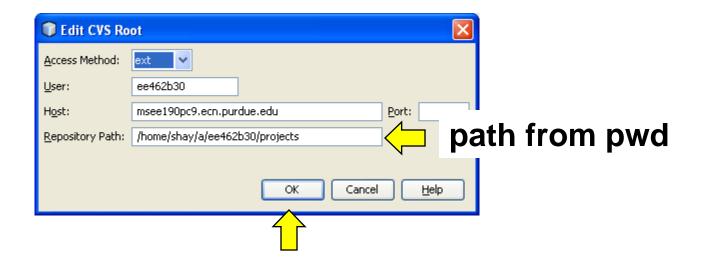


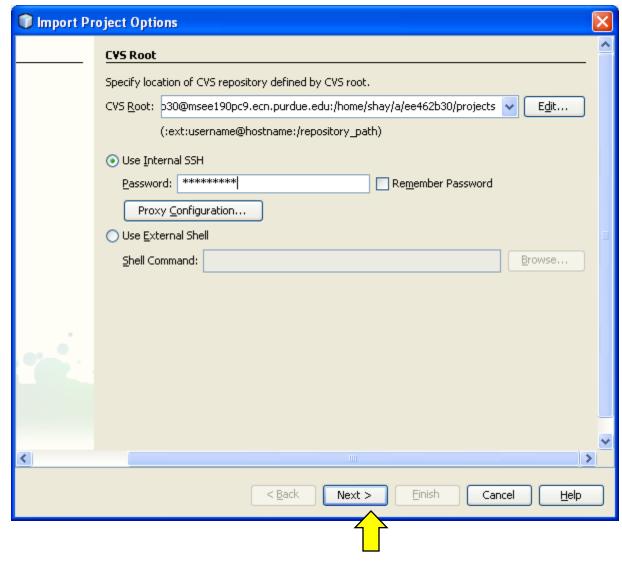


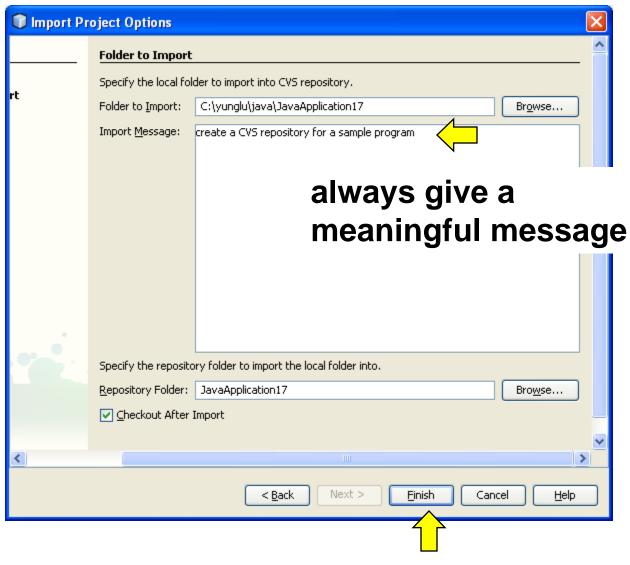


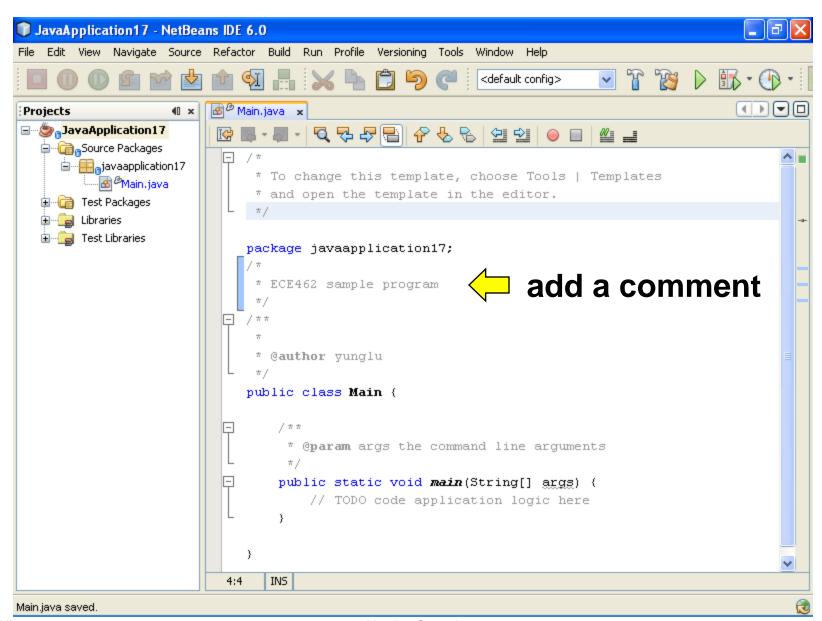




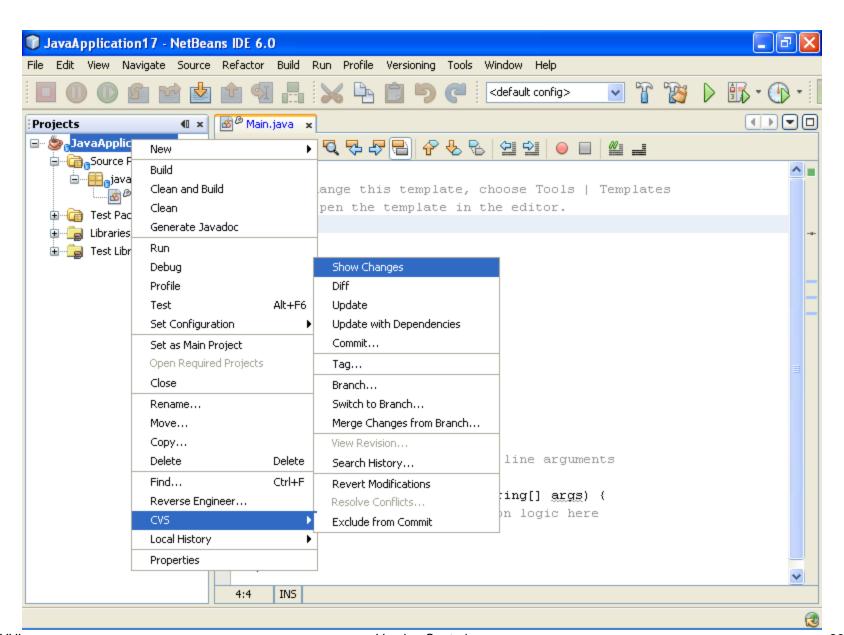


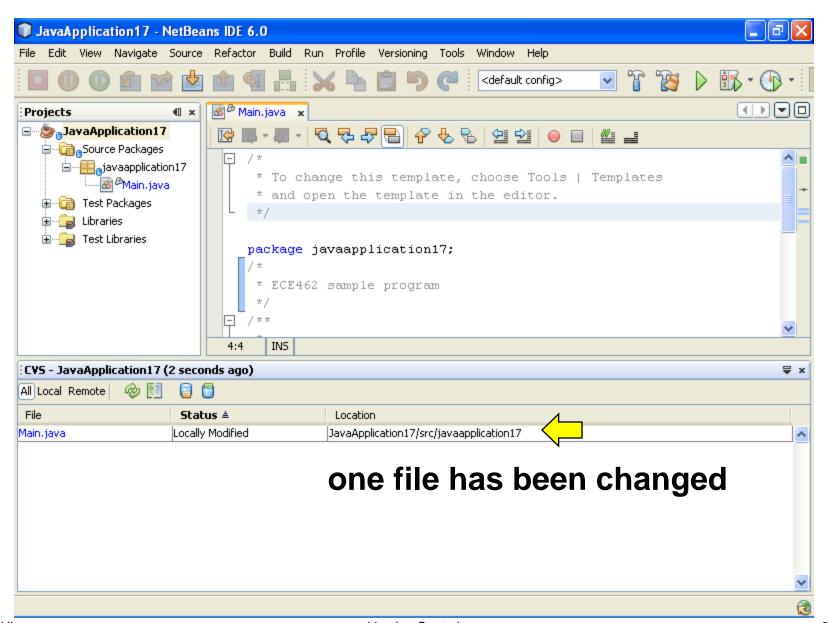




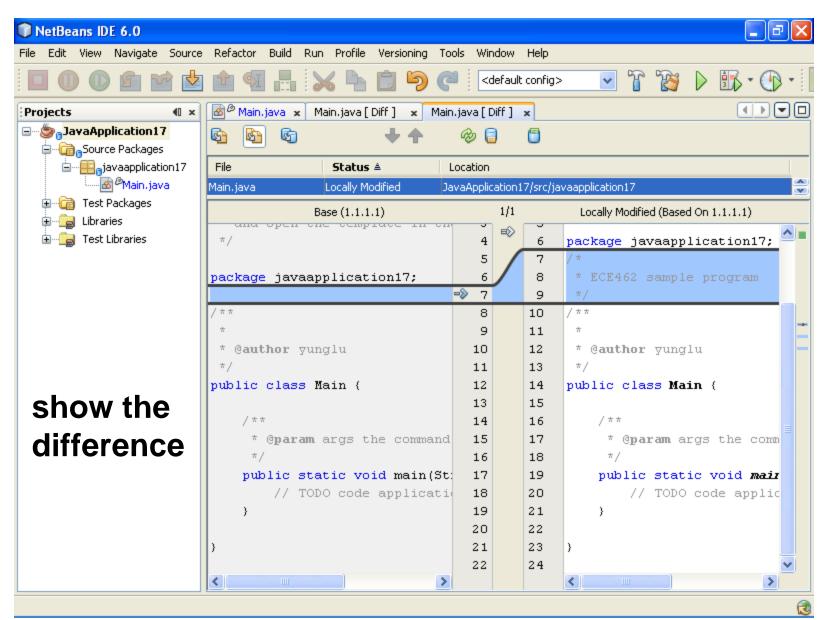


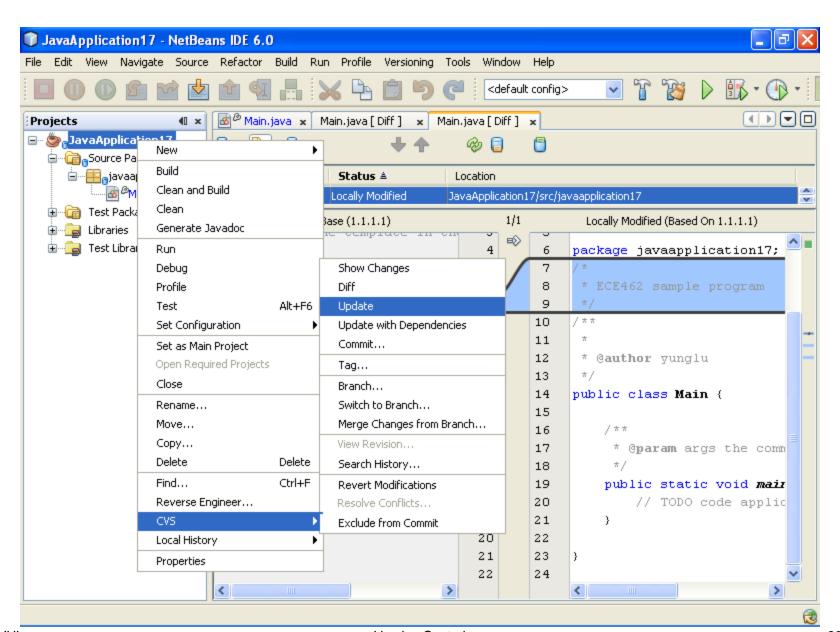
YHL Version Control 25





YHL

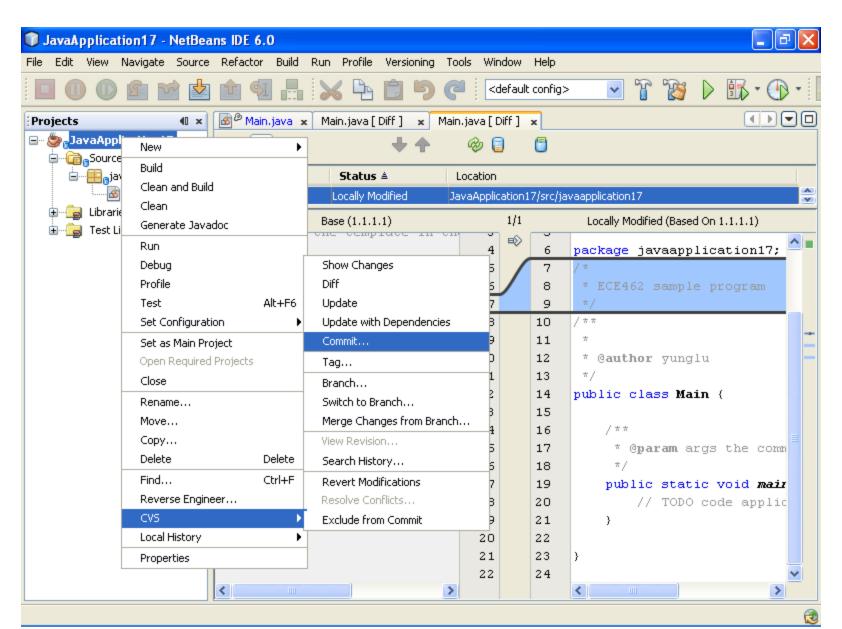


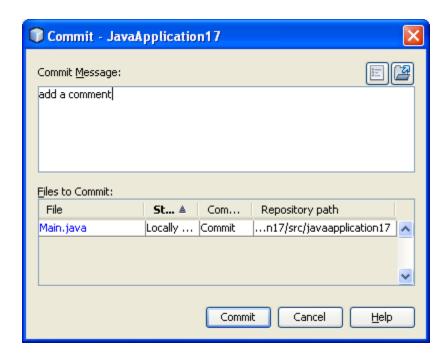


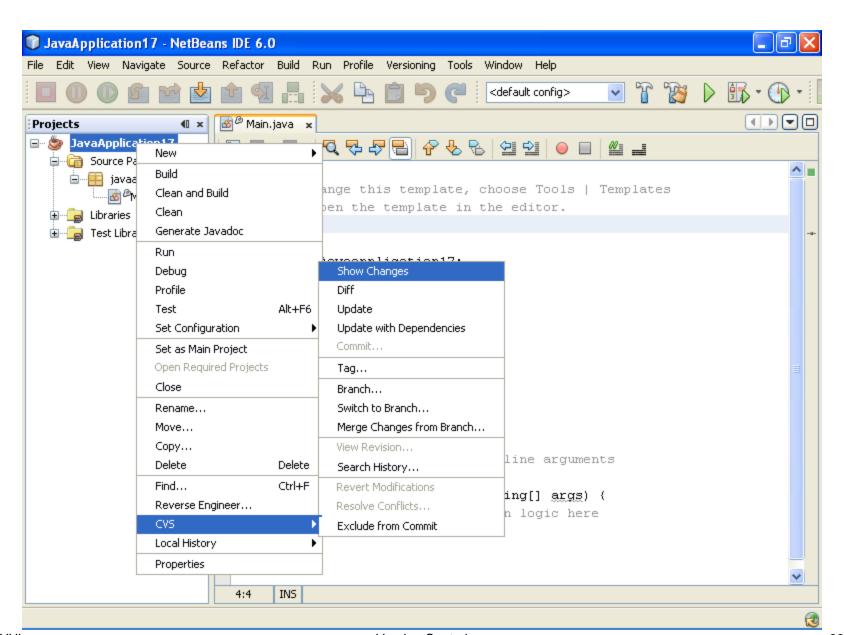
Update frequently Always update before commit.

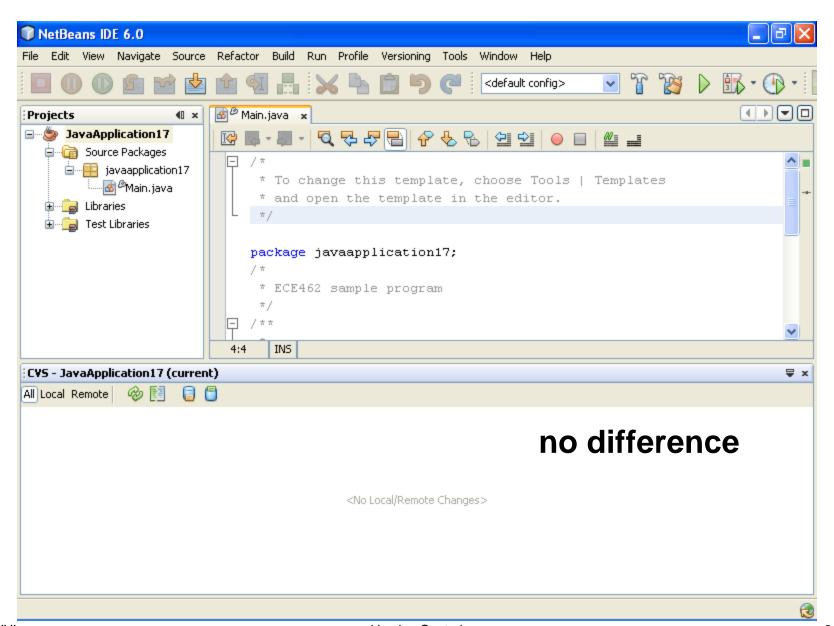
Update pulls the changes your teammates have committed.

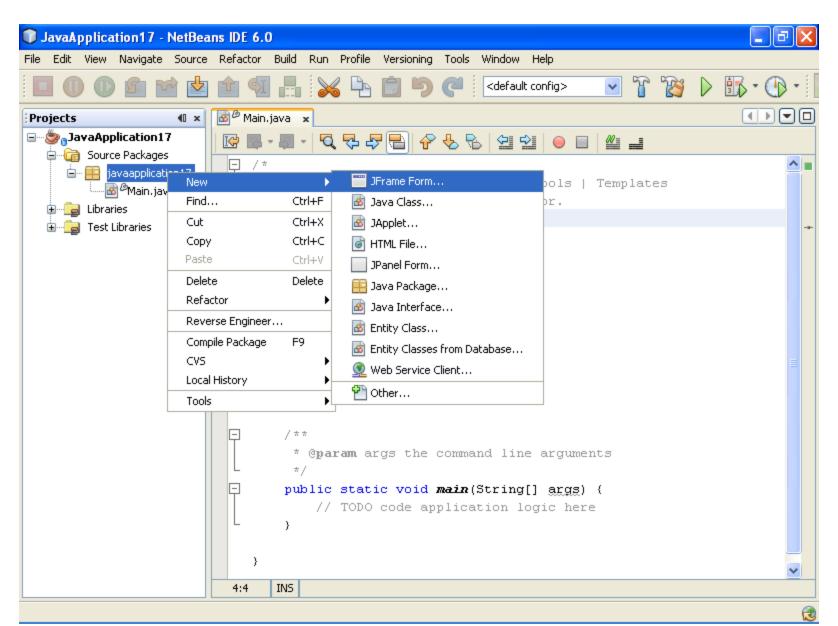
Commit pushes the changes you have made so that your teammates can see.

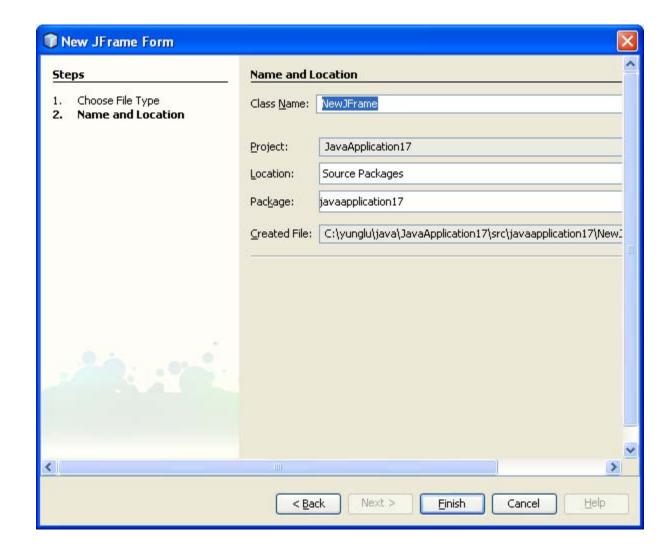




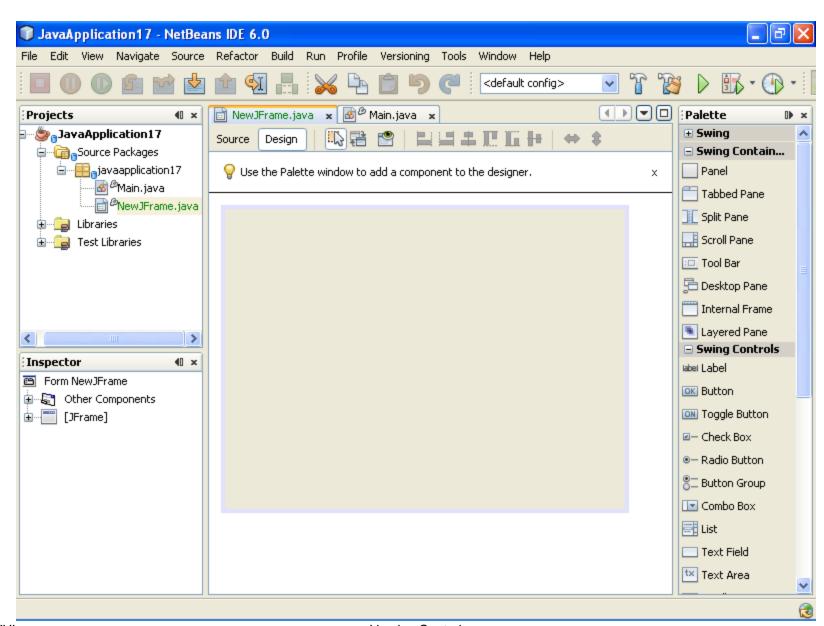


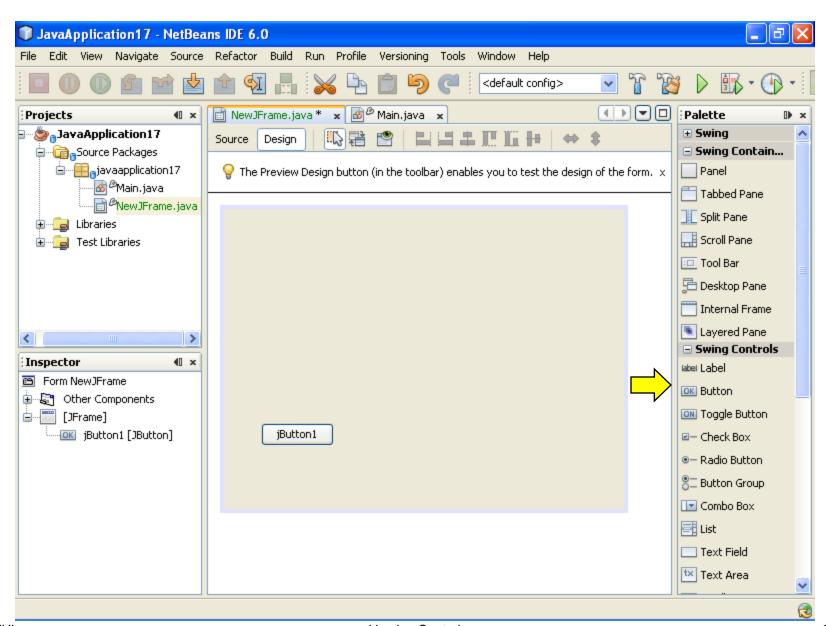


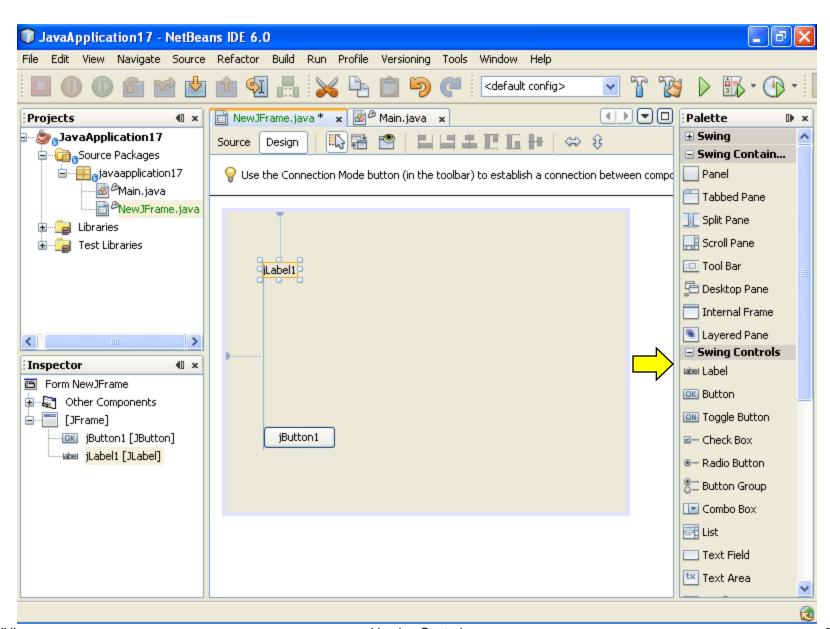


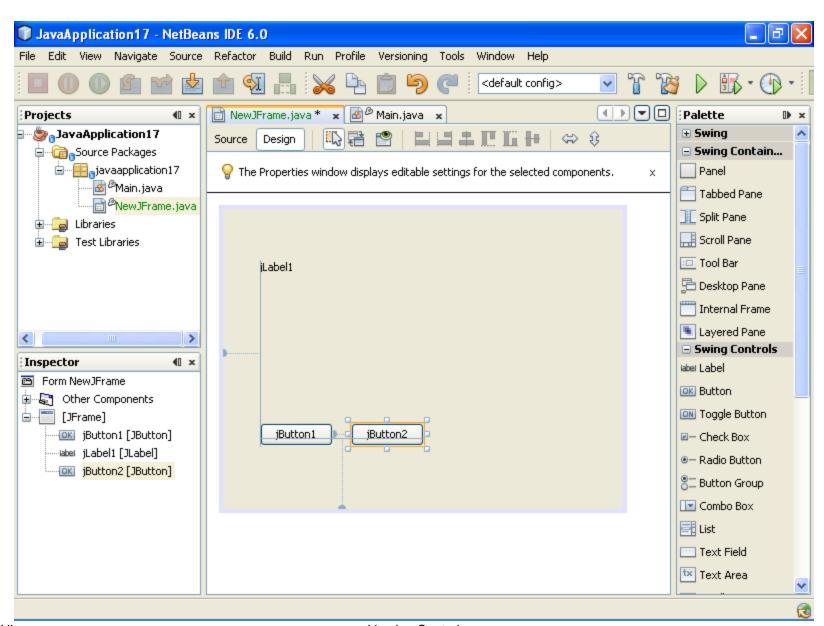


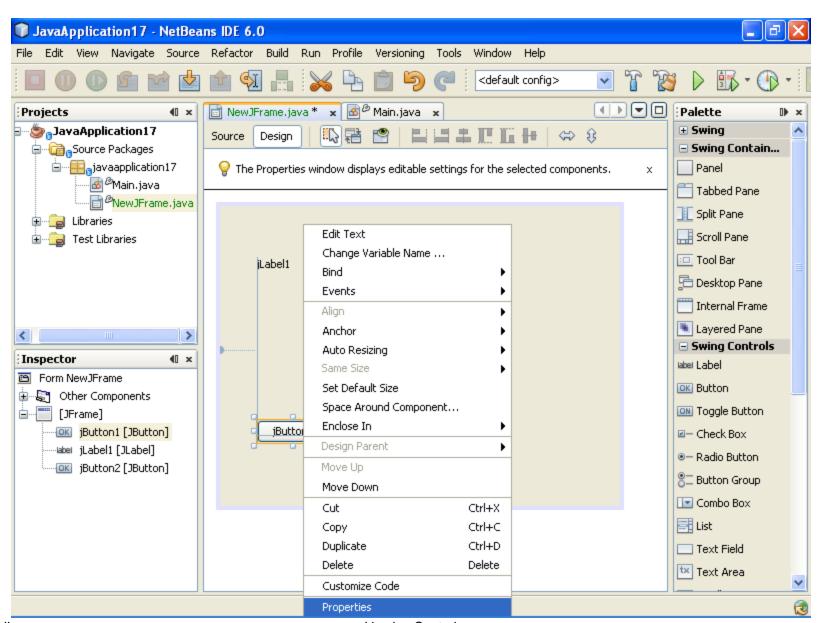
36

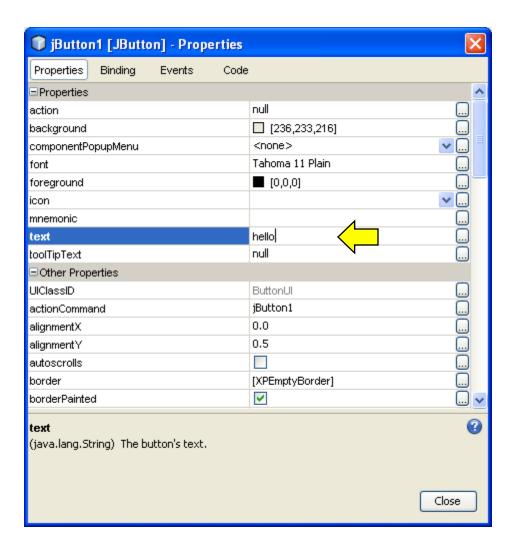


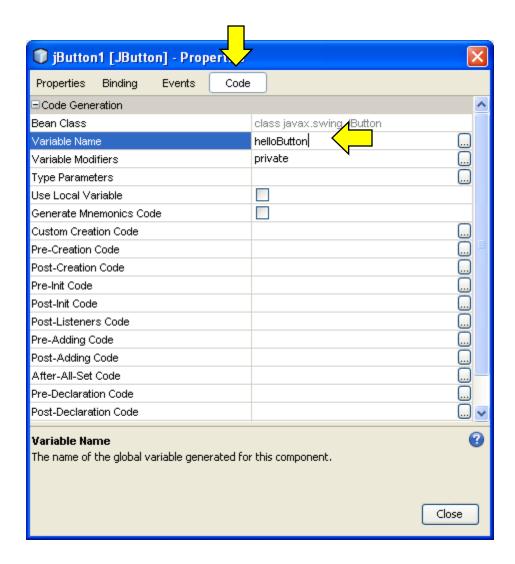


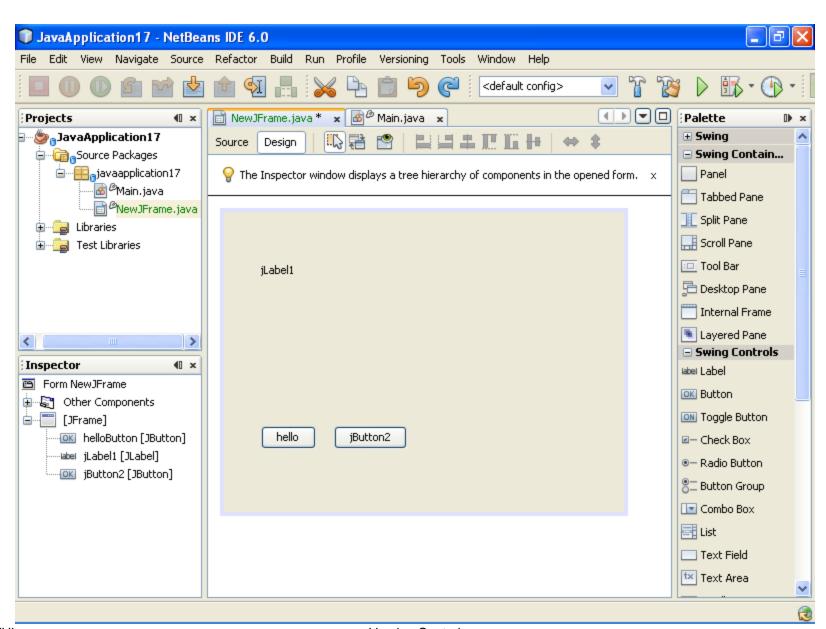


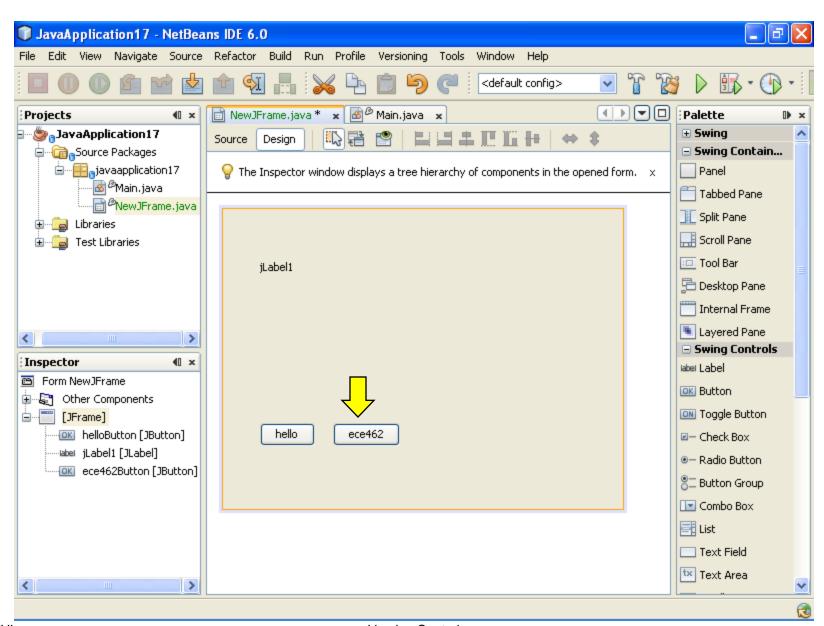


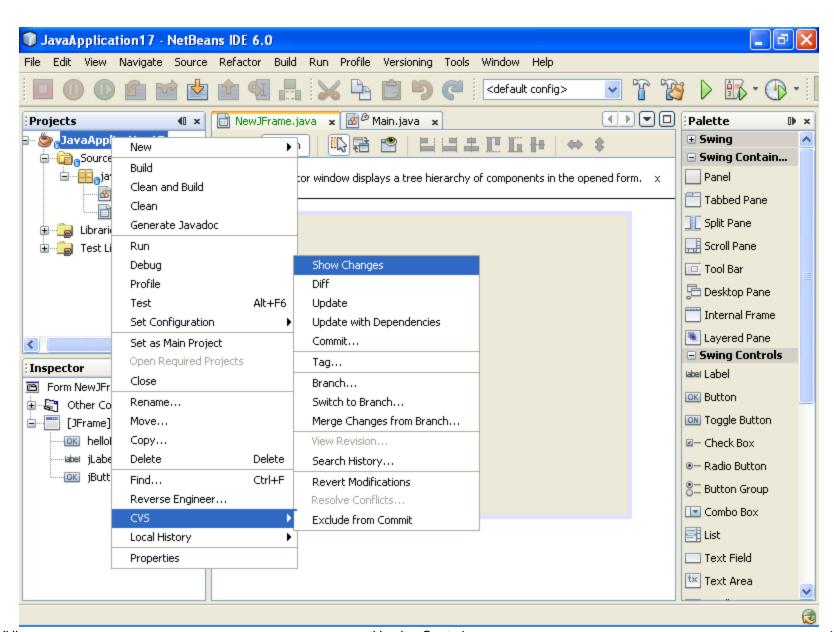


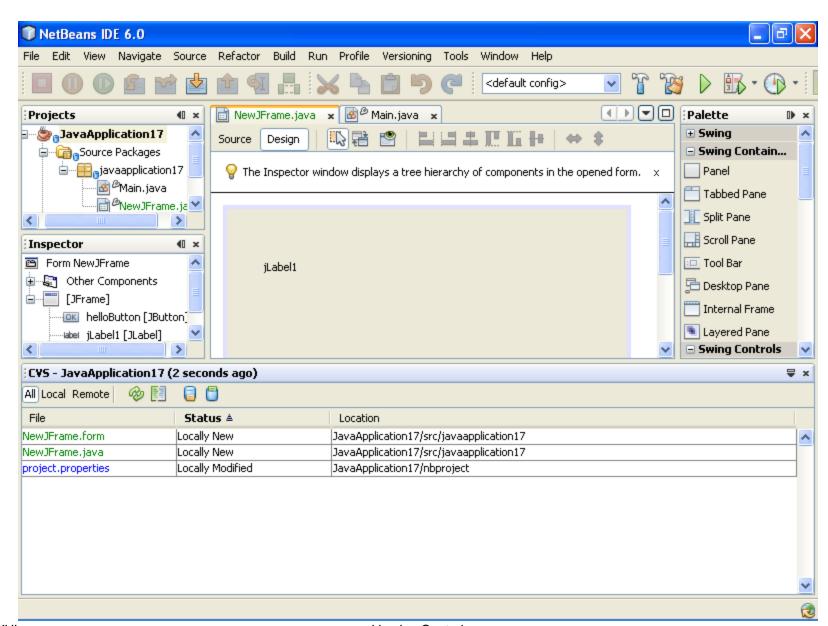


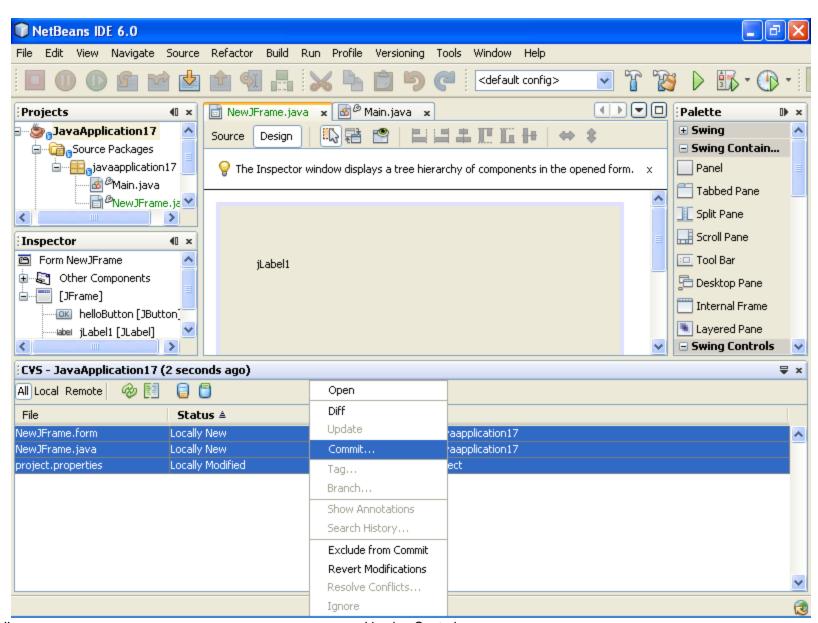


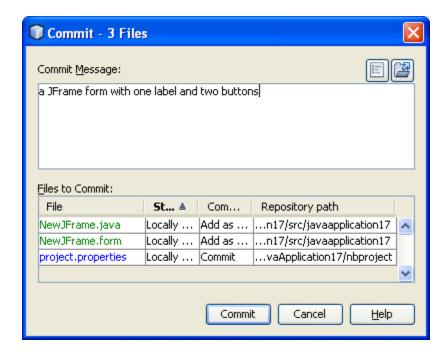


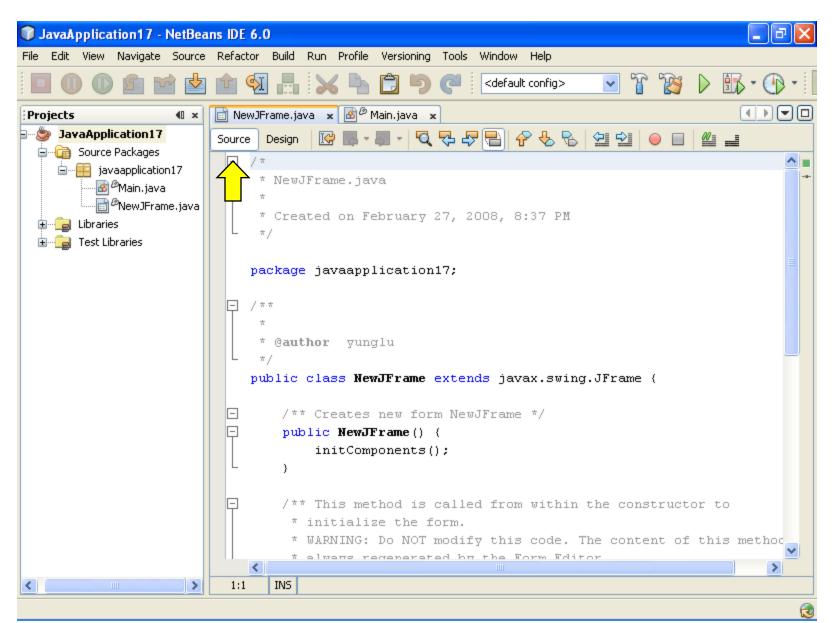


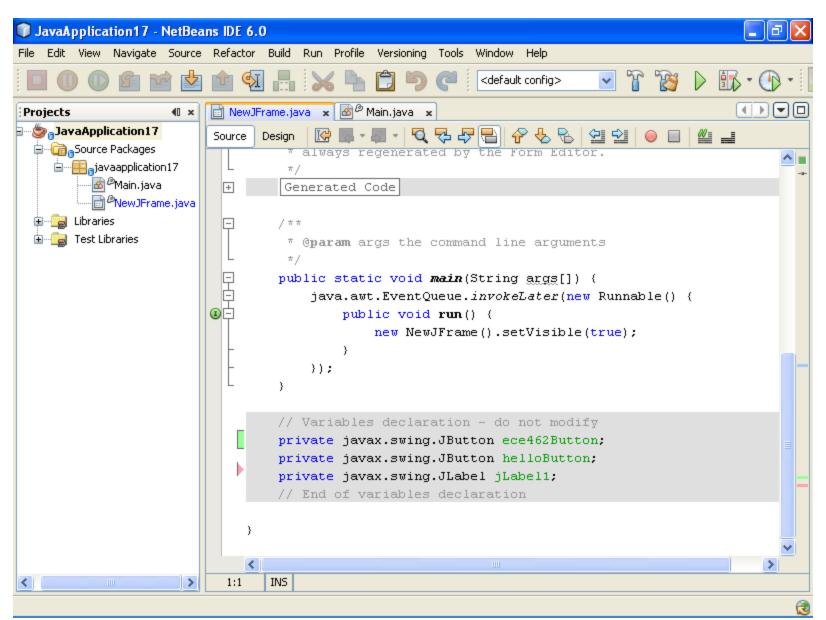


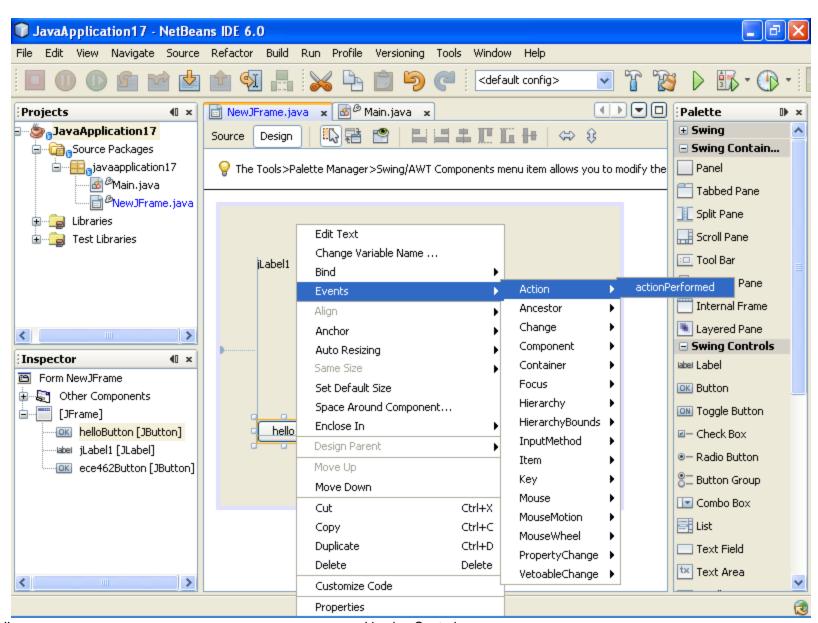


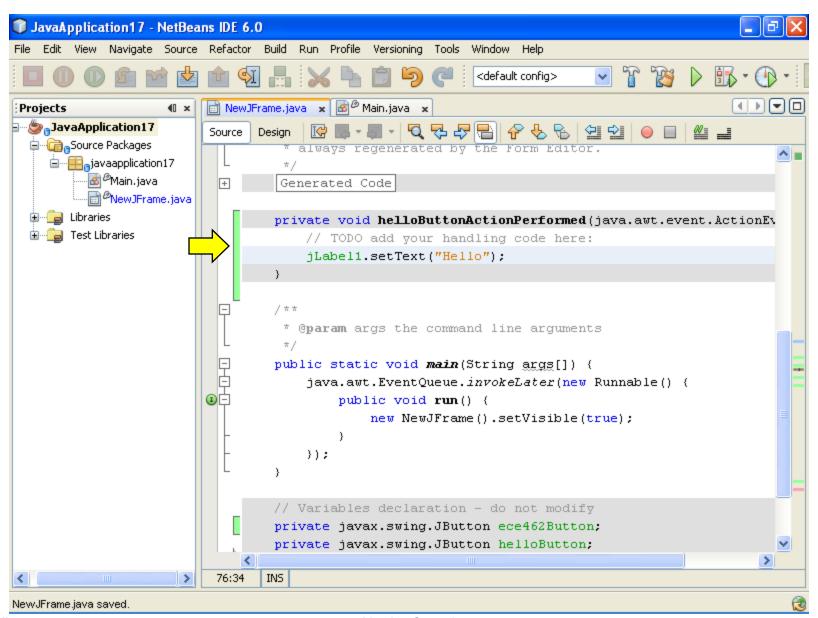


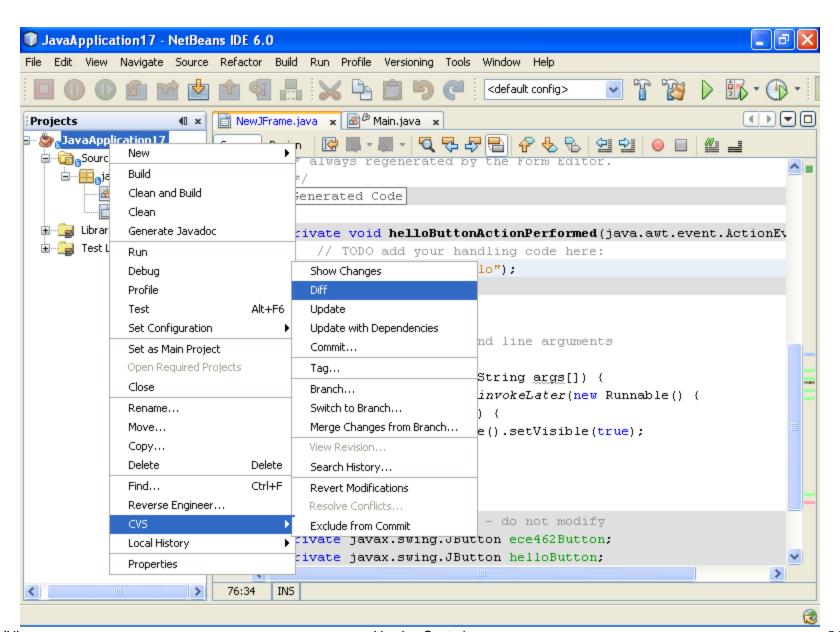


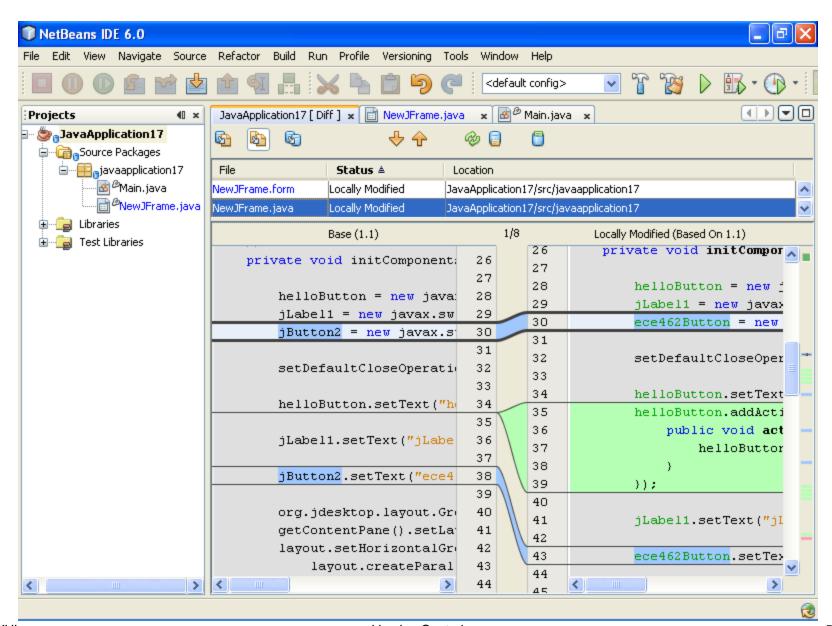


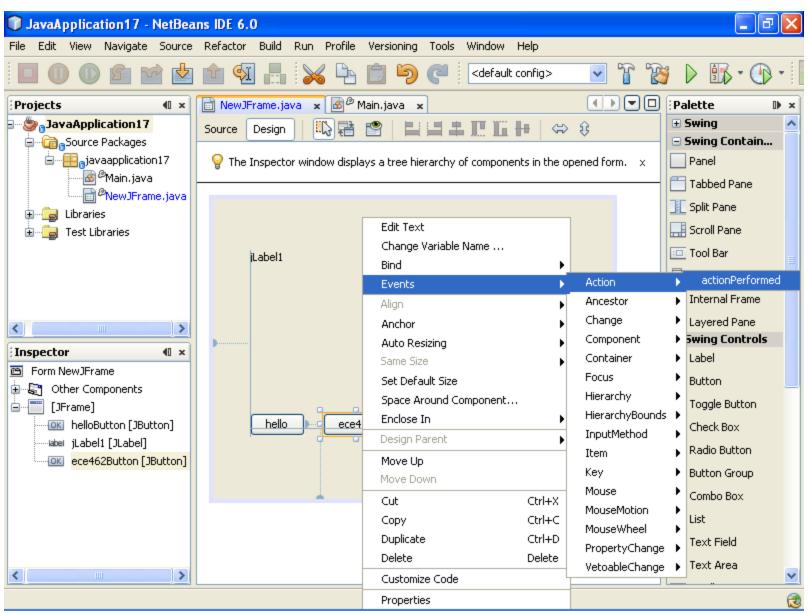




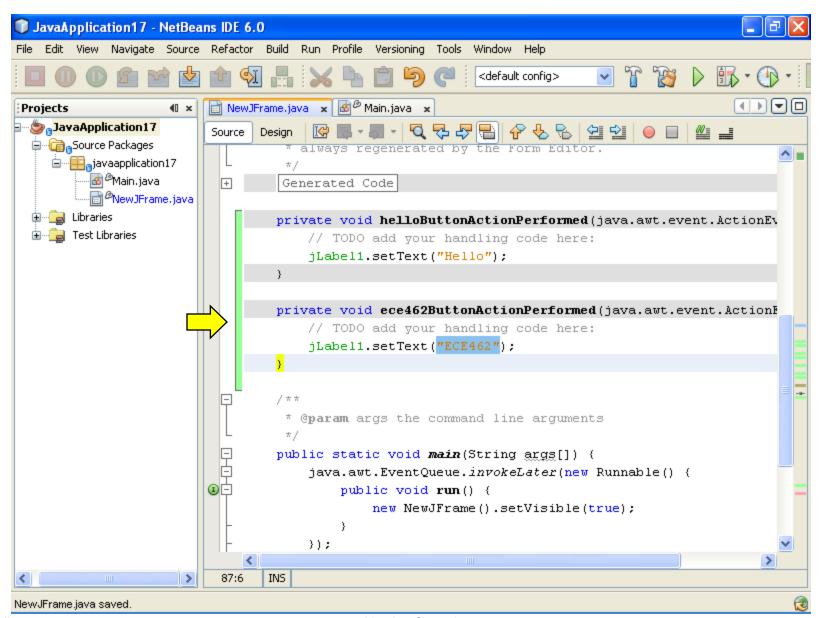


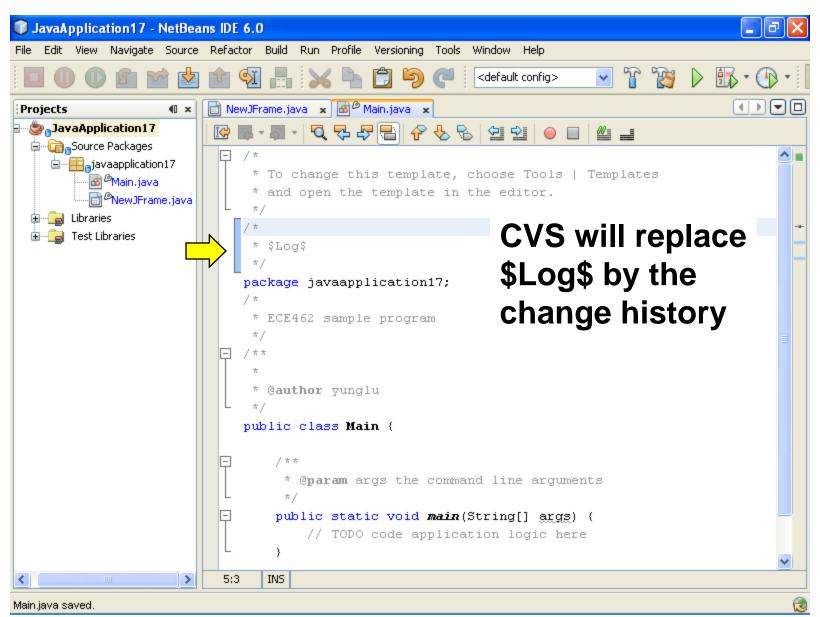




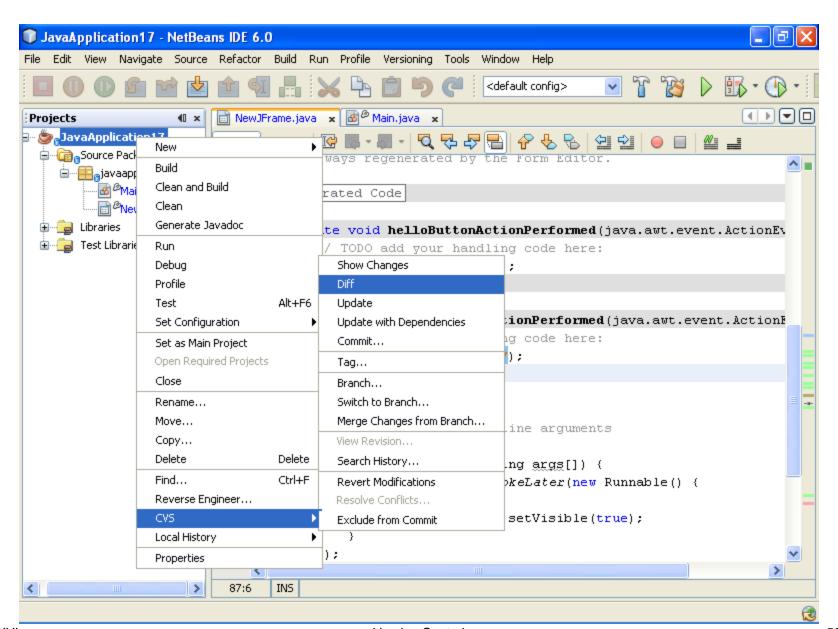


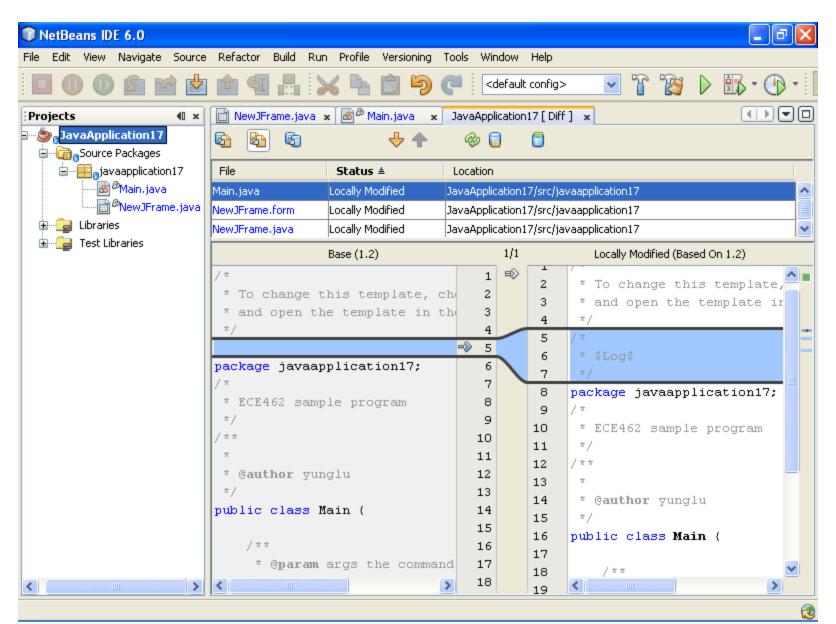
YHL

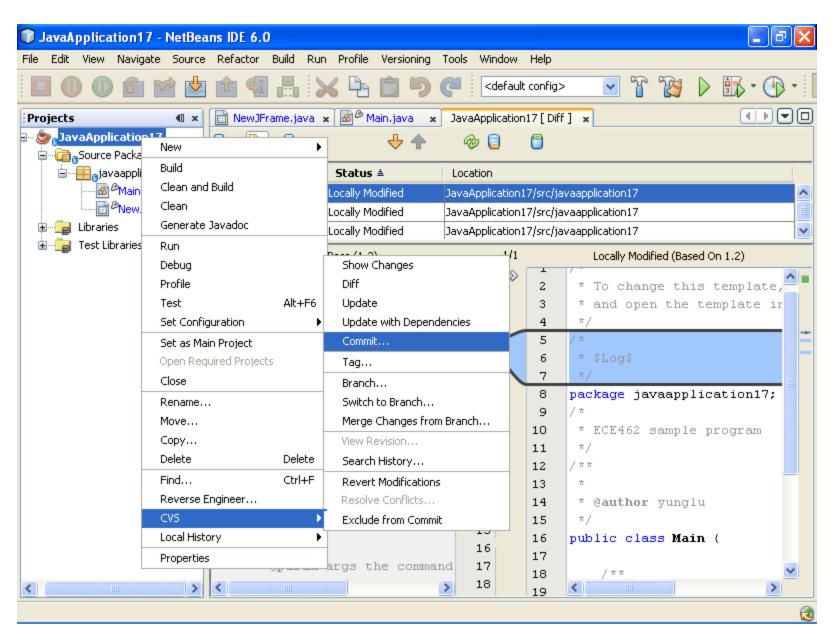


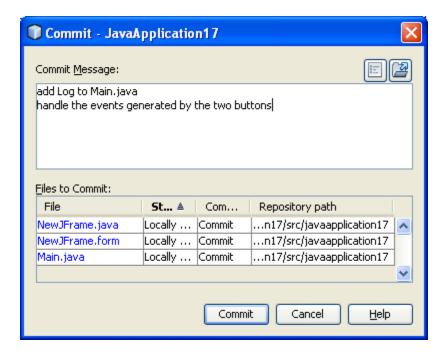


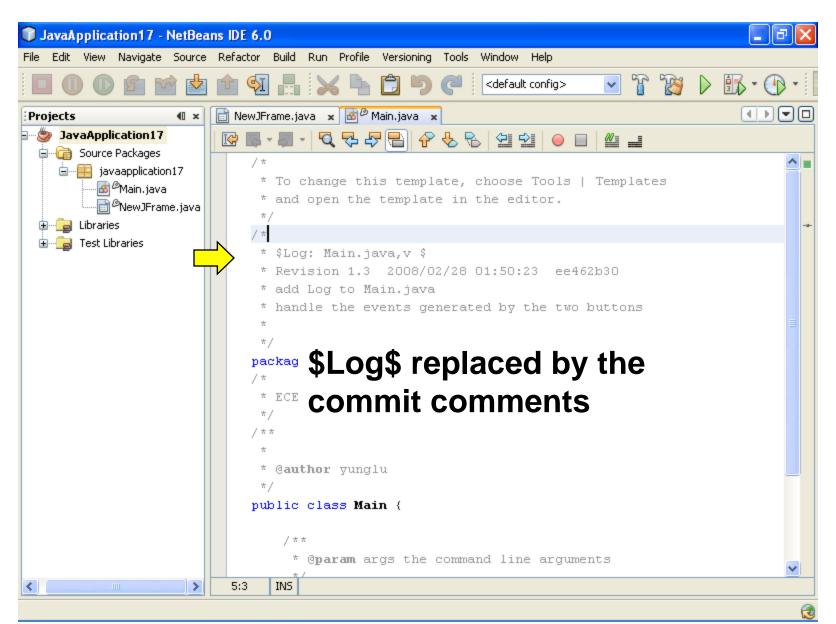
YHL Version Control 58

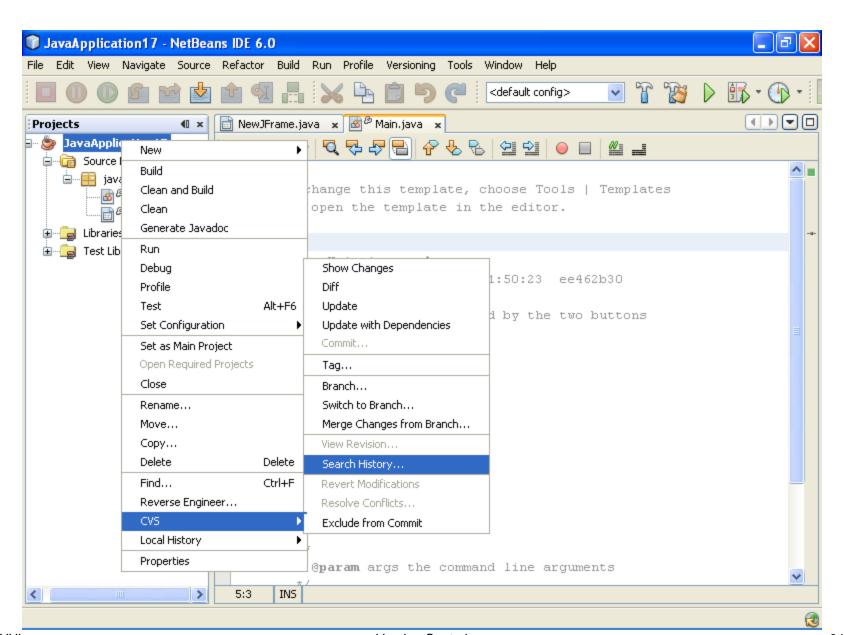


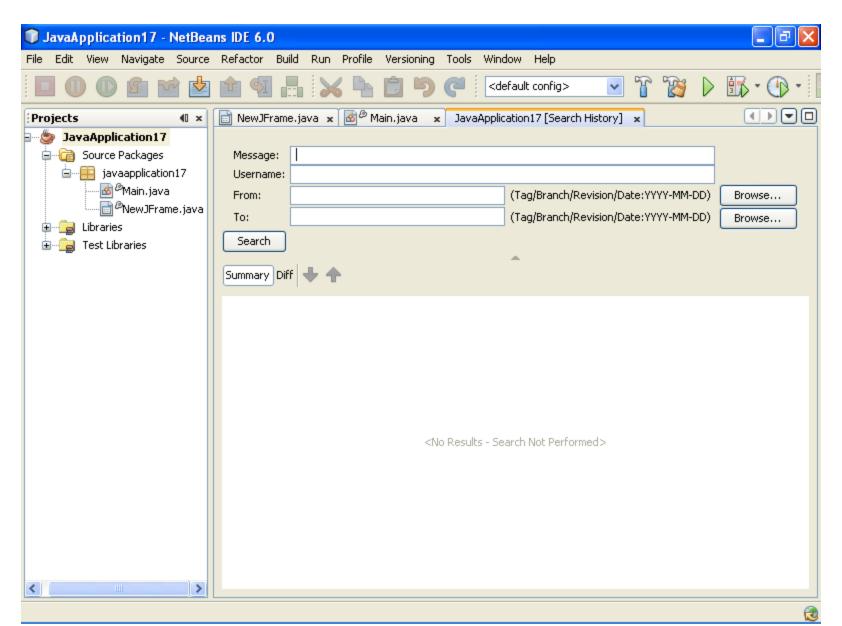


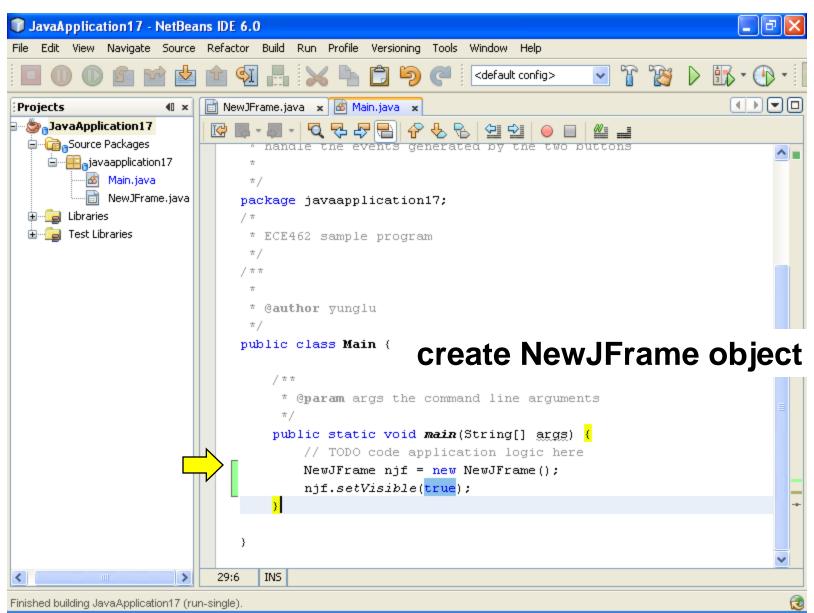


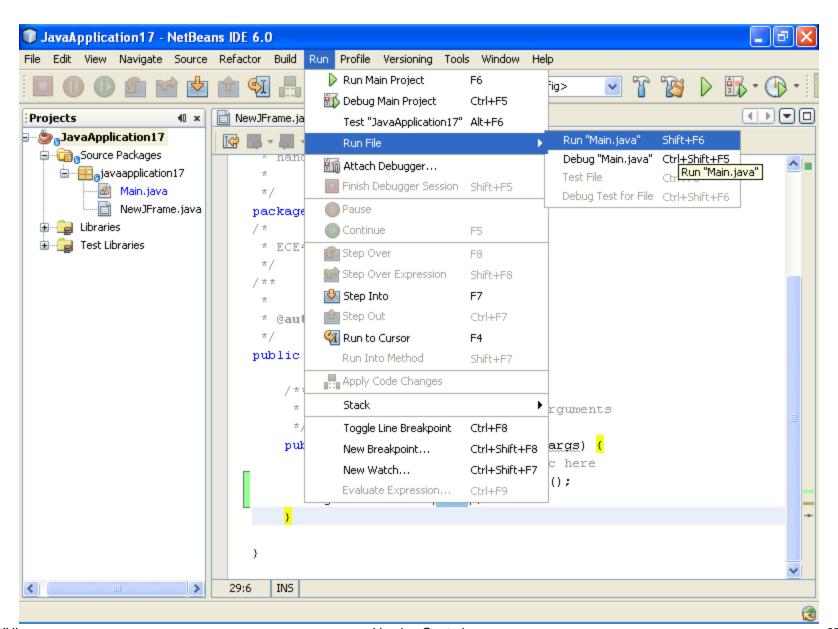




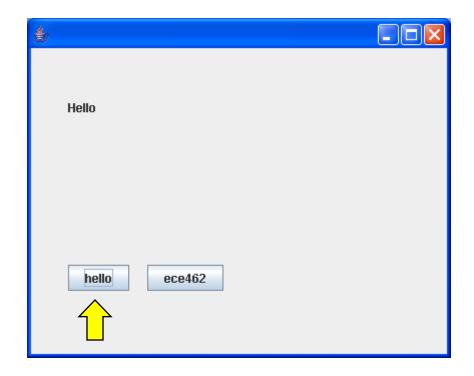


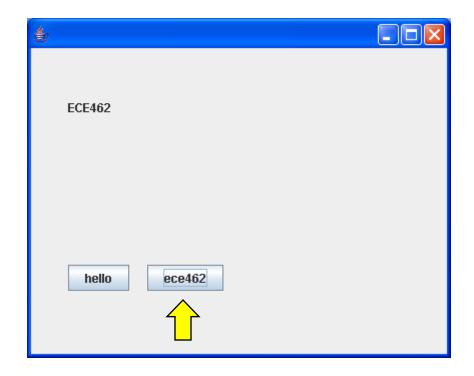


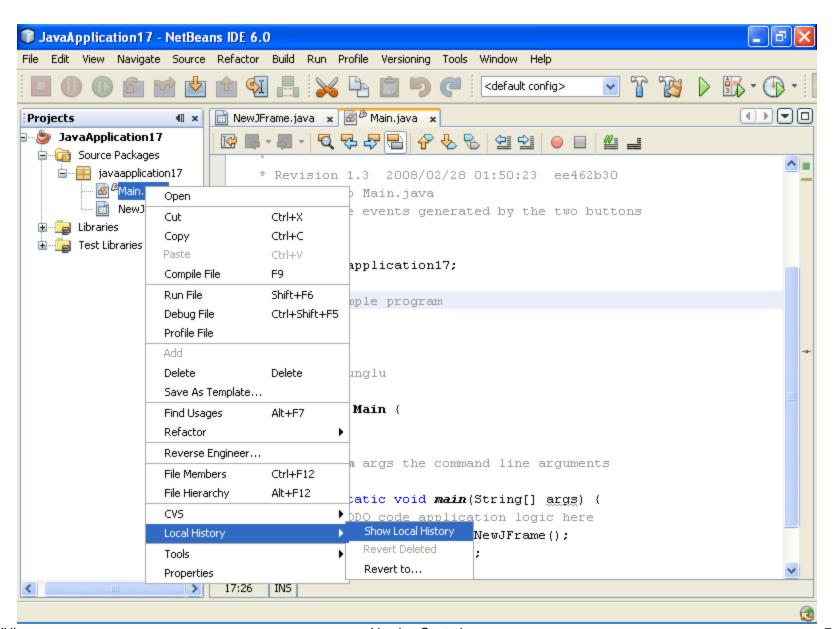


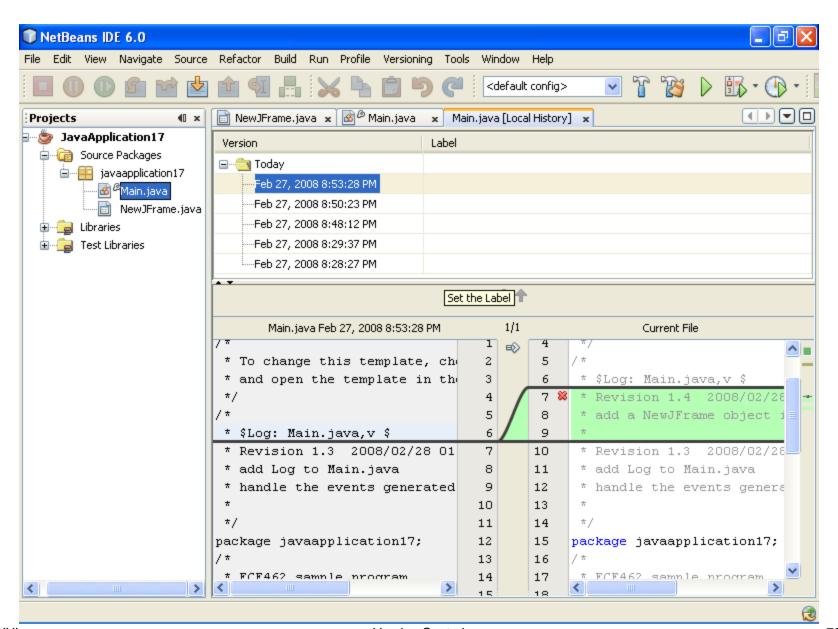


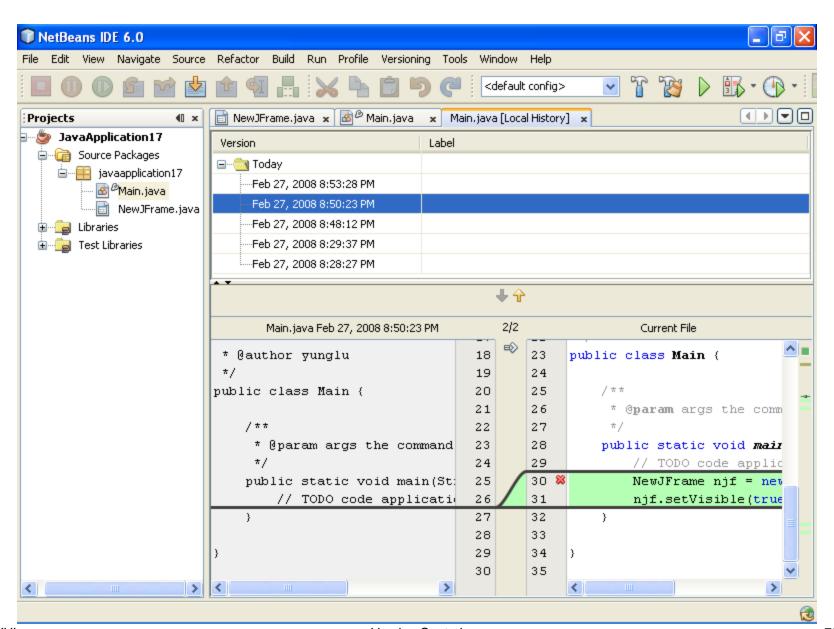


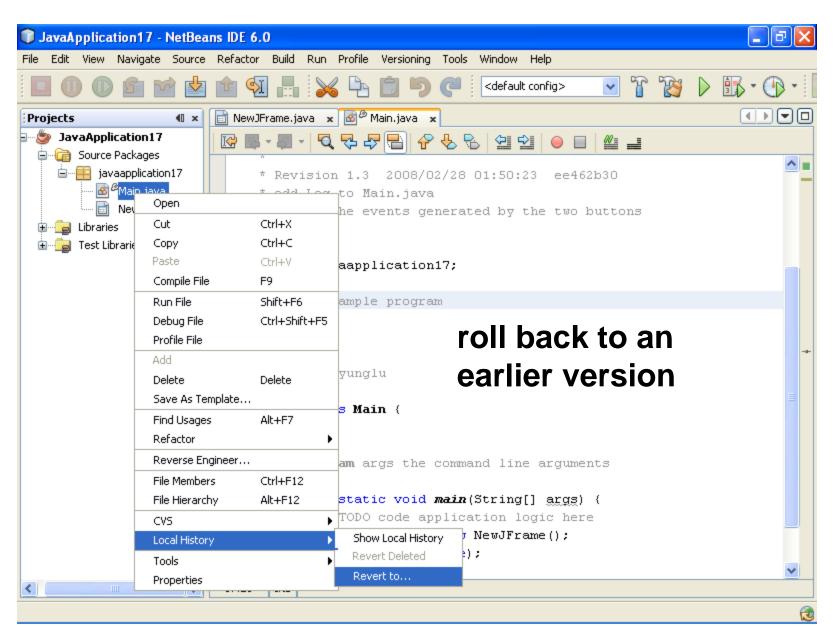


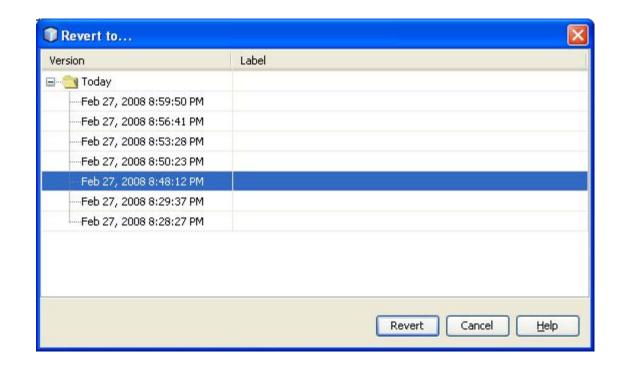


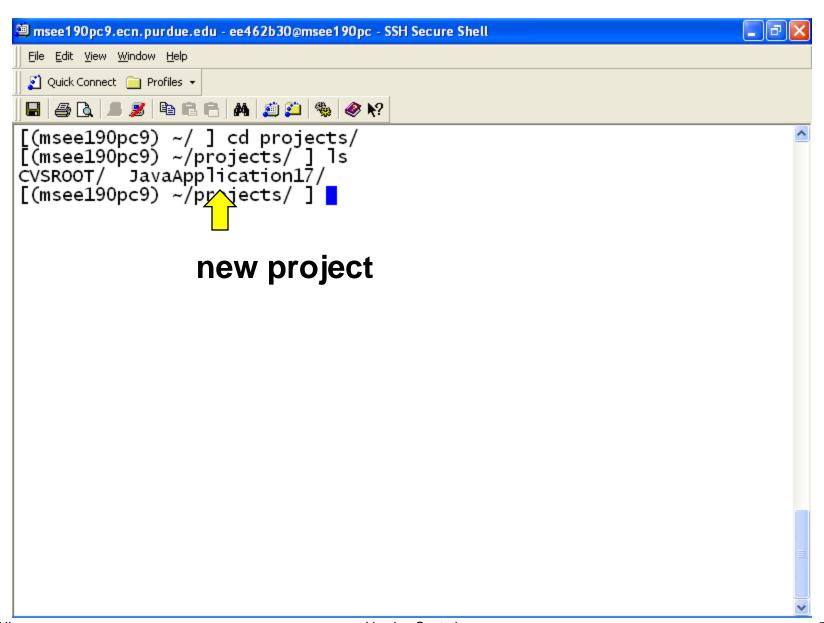


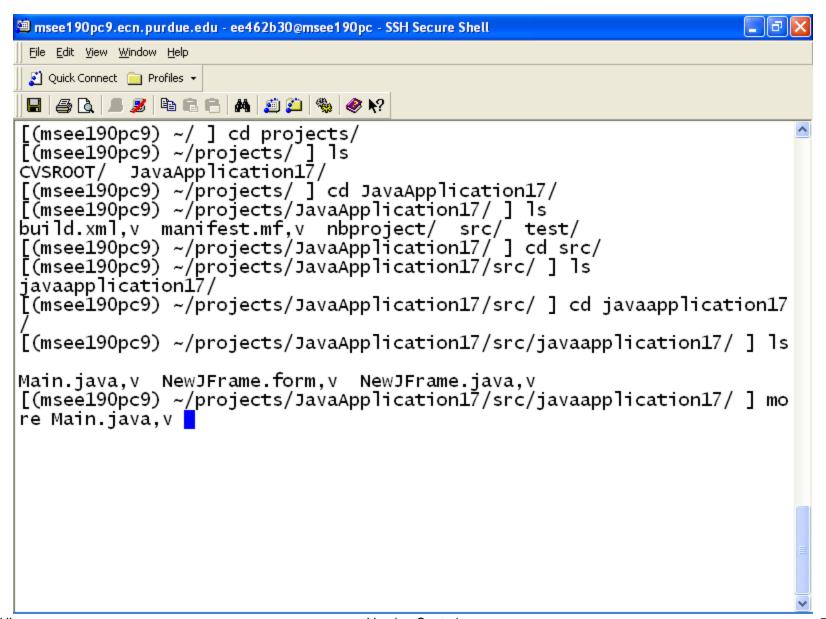






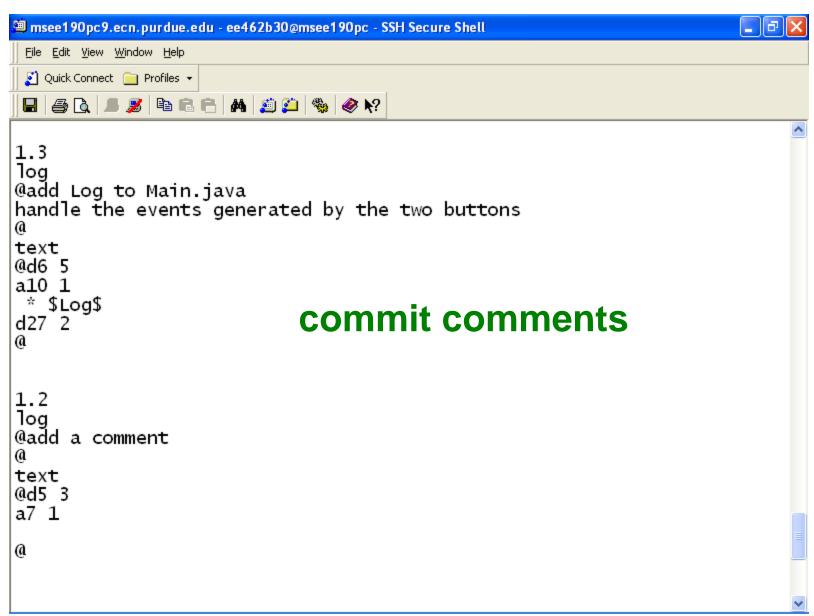






```
msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
 File Edit View Window Help
🗾 Quick Connect 📋 Profiles 🕶
re Main. java, v head 1.4; information about the revision history
access:
symbols
       default_release:1.1.1.1 default_vendor:1.1.1;
locks: strict:
comment @# @:
1.4
date
       2008.02.28.01.56.41; author ee462b30; state Exp;
branches;
       1.3:
next
1.3
date
       2008.02.28.01.50.23; author ee462b30;
                                                      state Exp;
branches:
next
1.2
date
       2008.02.28.01.33.17; author ee462b30; state Exp;
branches:
       1.1:
next
--More--(23%)
```

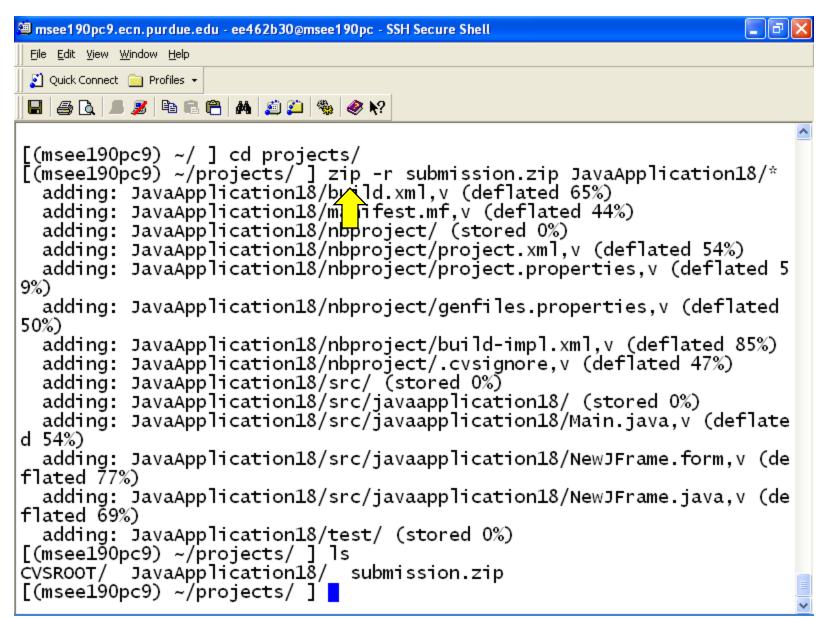
YHL Version Control 78



Do not copy / move / delete files in the repository directly. Always use CVS commands.

CVS adds additional information to each file.

Submission Instruction Create a Zip file of the repository (not the source code)



YHL Version Control 82

Self Test

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

Instantiation

Yung-Hsiang Lu yunglu@purdue.edu

Instantiation

- instantiation: create an object of a class
- Java:

```
ClassName objectName = new ClassName(...);

JLabel jLabel1 = new javax.swing.JLabel();
```

• C++

```
ClassName * objectName = new ClassName(...);
or
```

ClassName objectName (...);

QVBoxLayout *layout = new QVBoxLayout; classNameLabel = new QLabel(tr("&Class name:"));

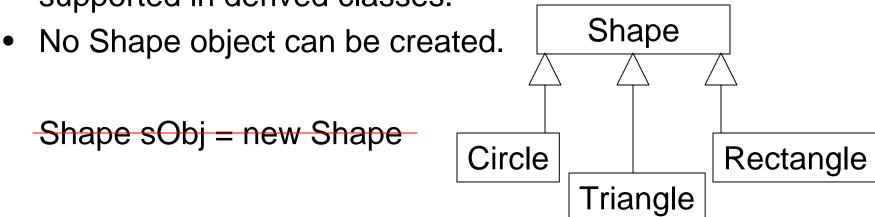
Abstract Class = Cannot Create Objects of *This* Class

But objects can be created for derived classes

Abstract Java Class and Method

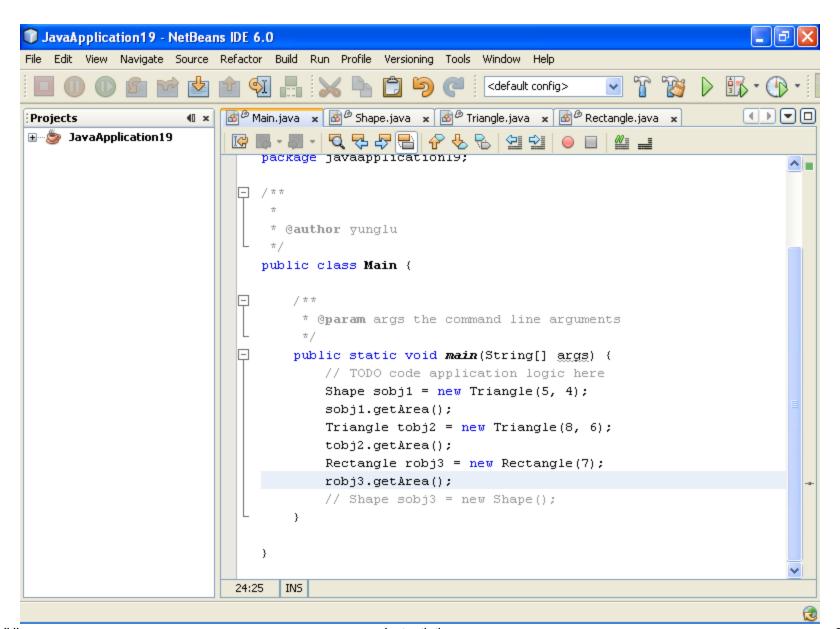
Abstract Class

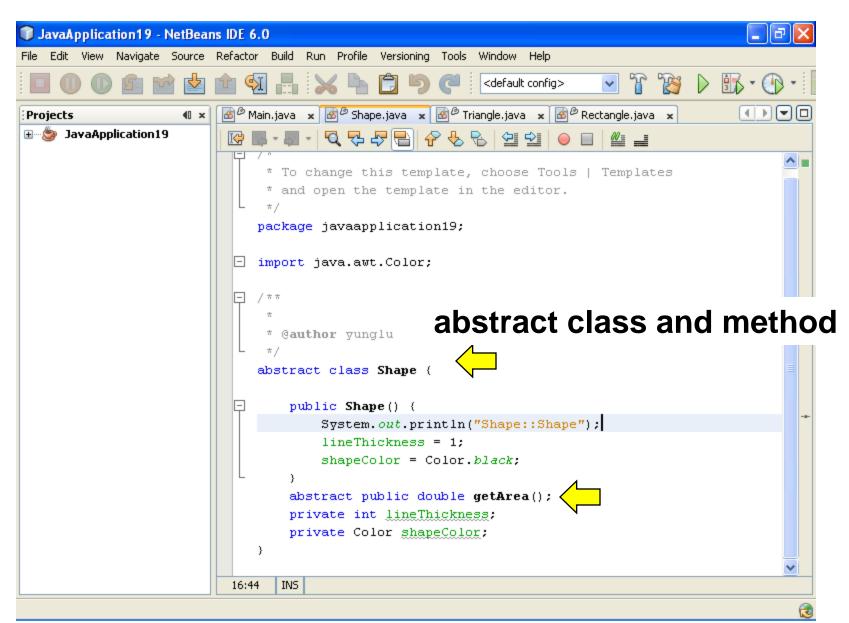
- Some classes do not allow instantiation. These classes serve as base classes for other classes.
- Shape can contain common features, including attributes (such as color, line thickness) and methods (such as setColor and getLineThickness).
- Shape can also promise methods (getArea) that must be supported in derived classes.

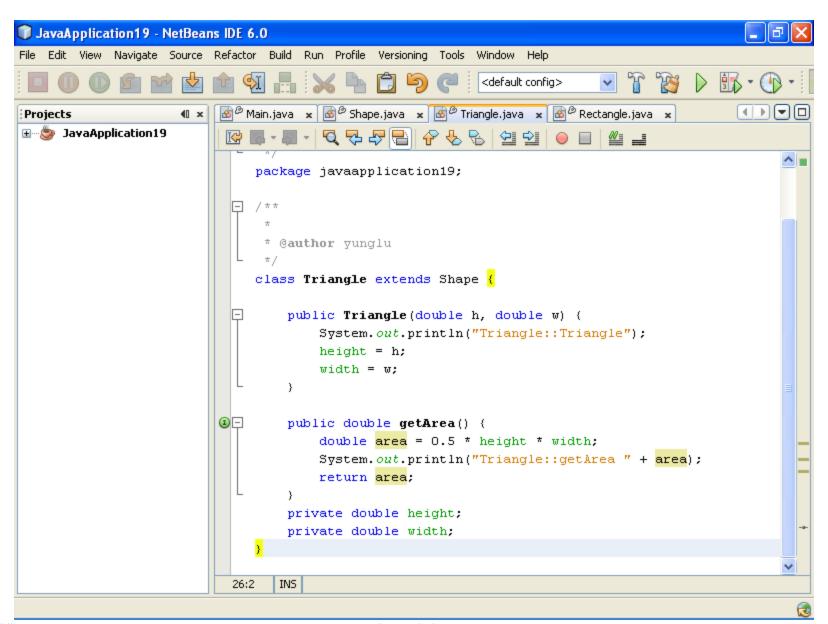


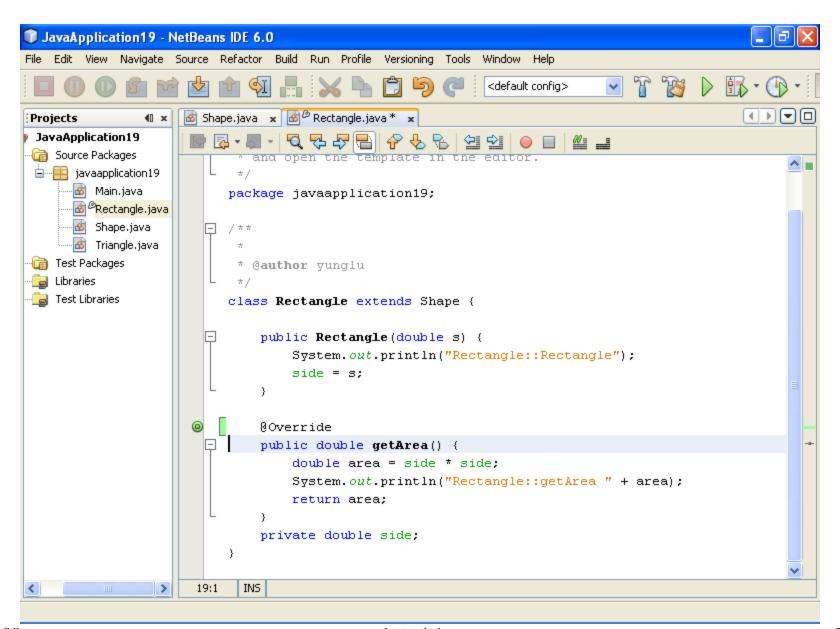
Instantiation

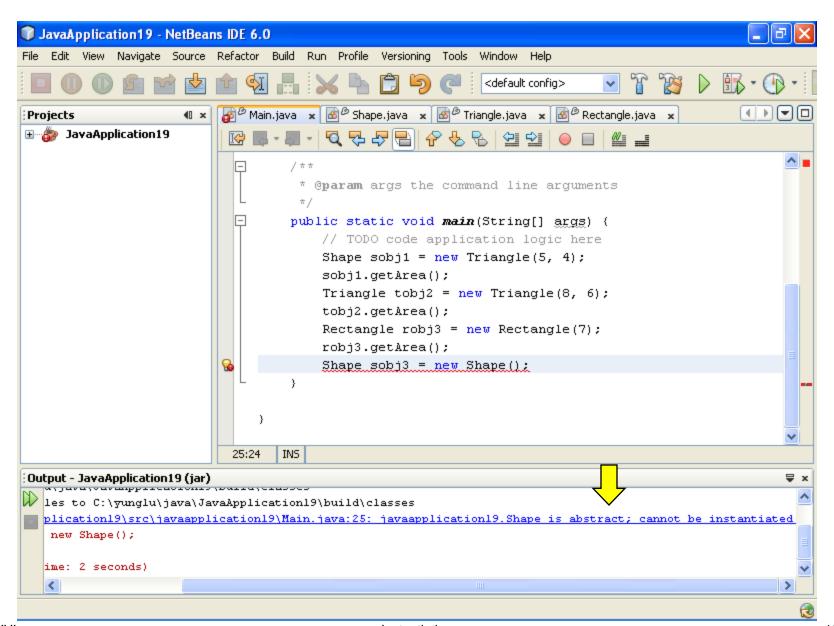
5

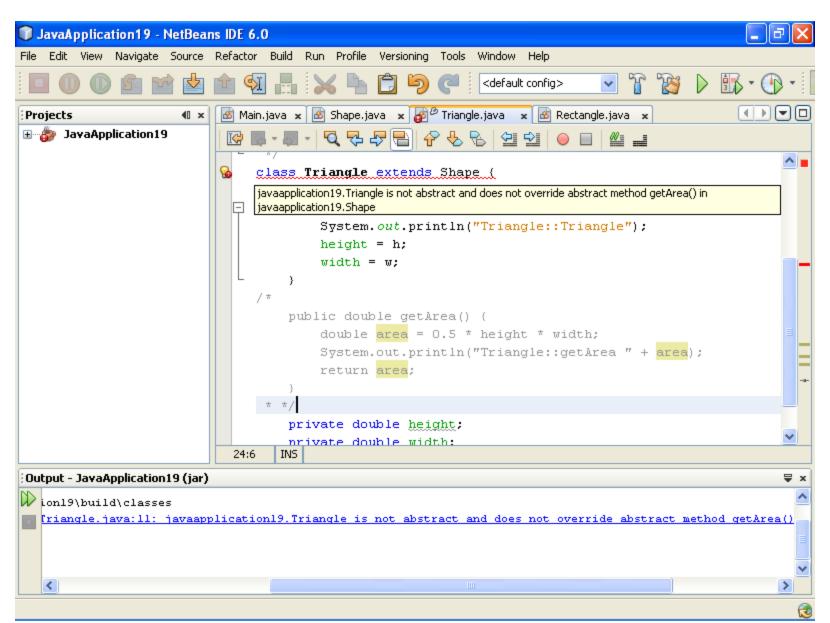


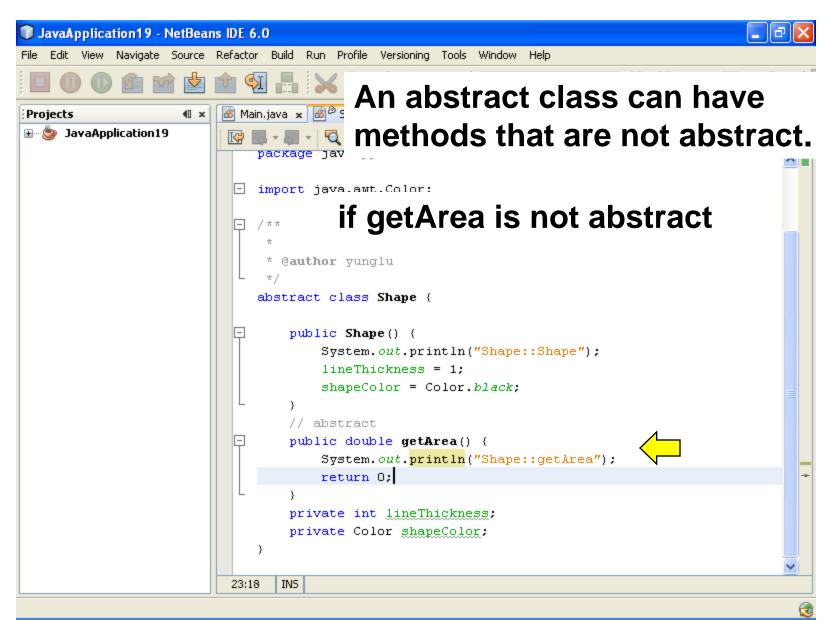


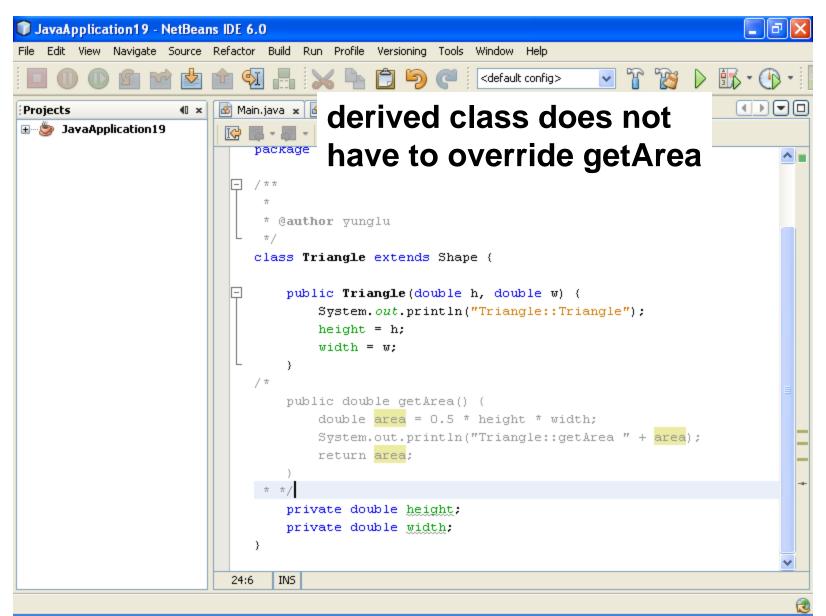


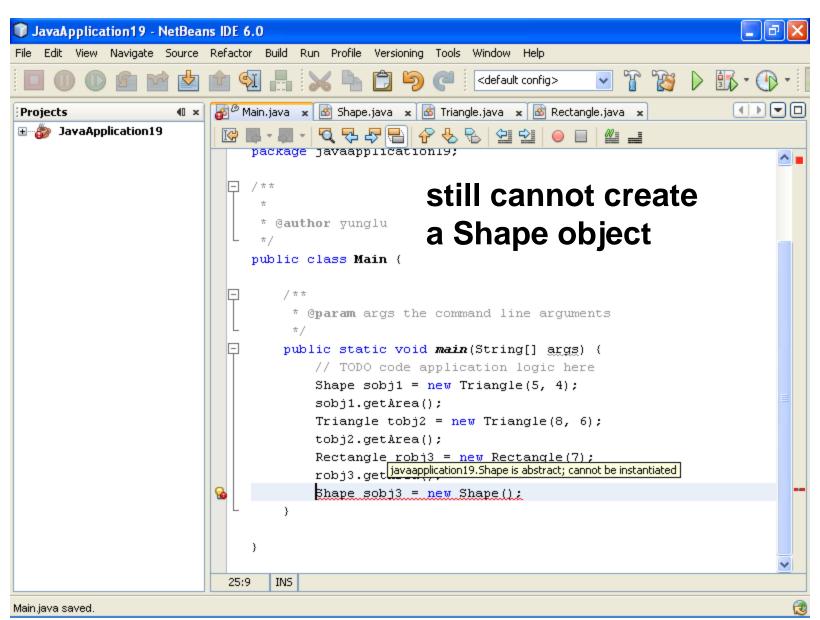


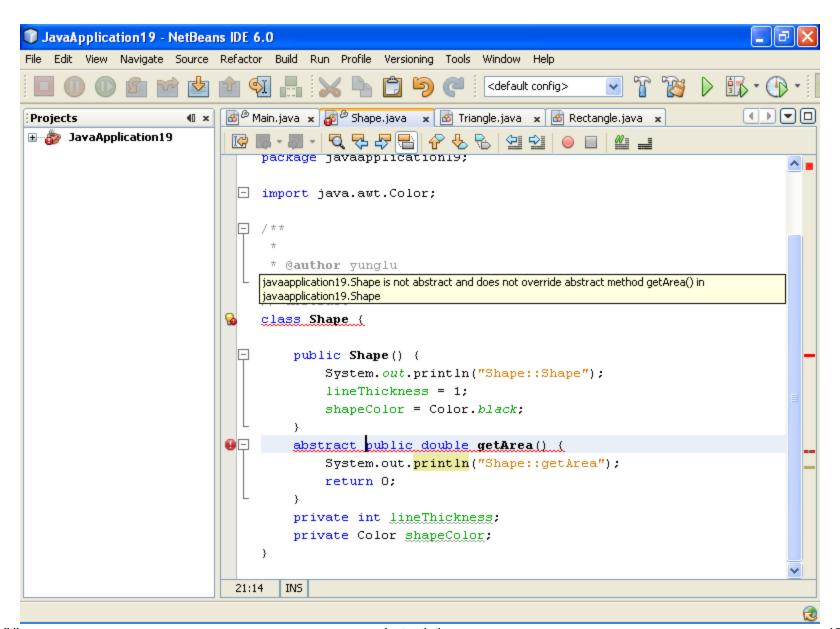


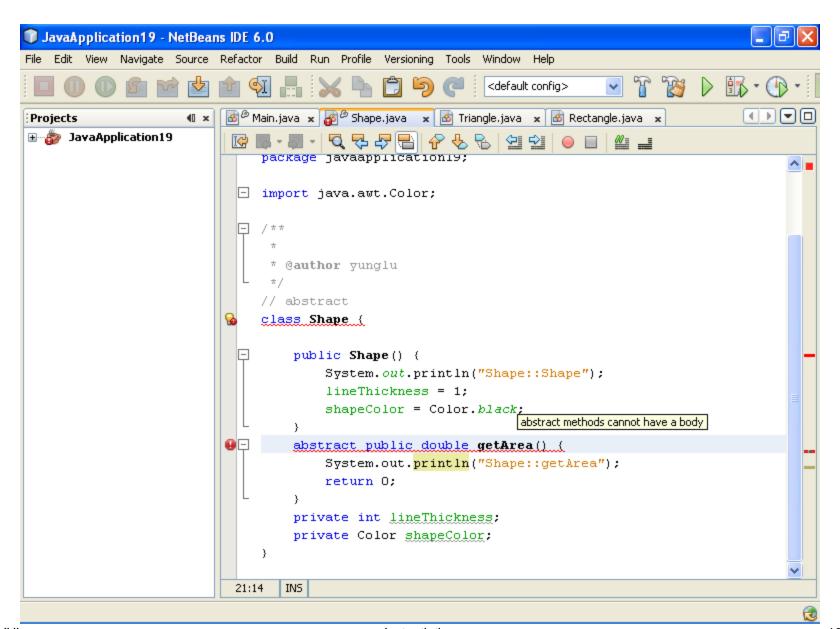


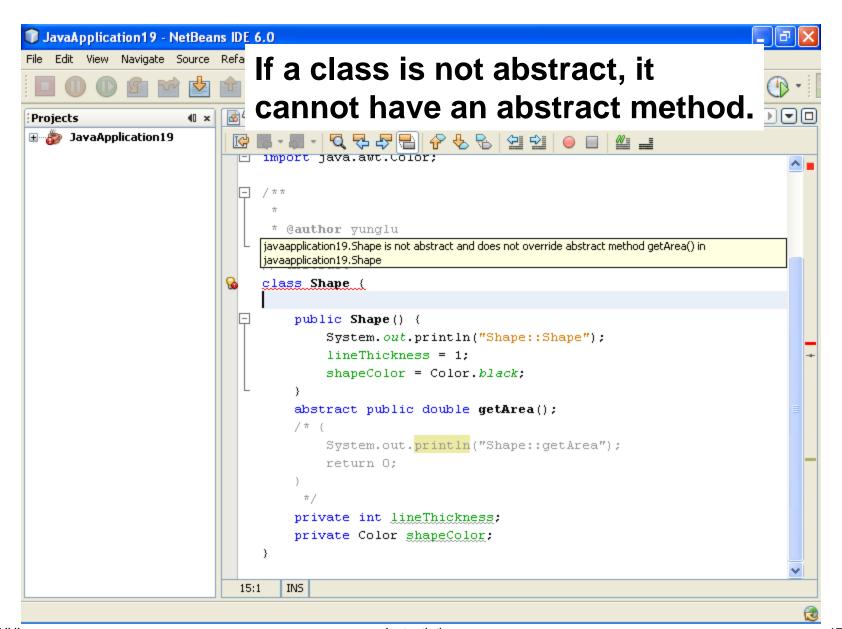


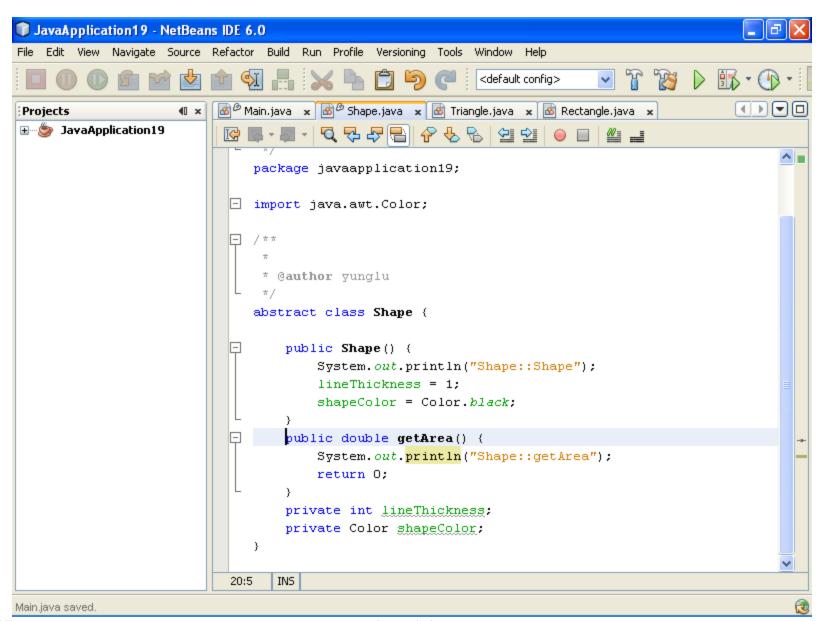






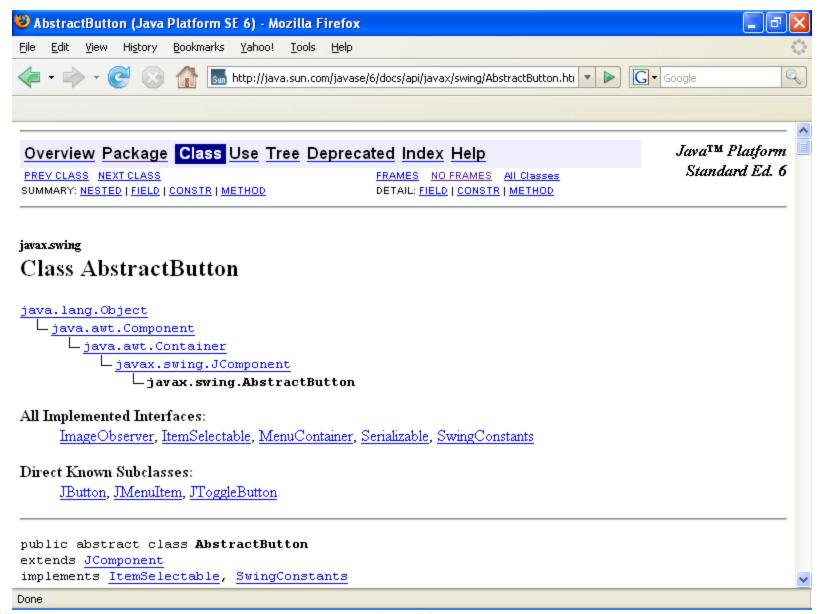




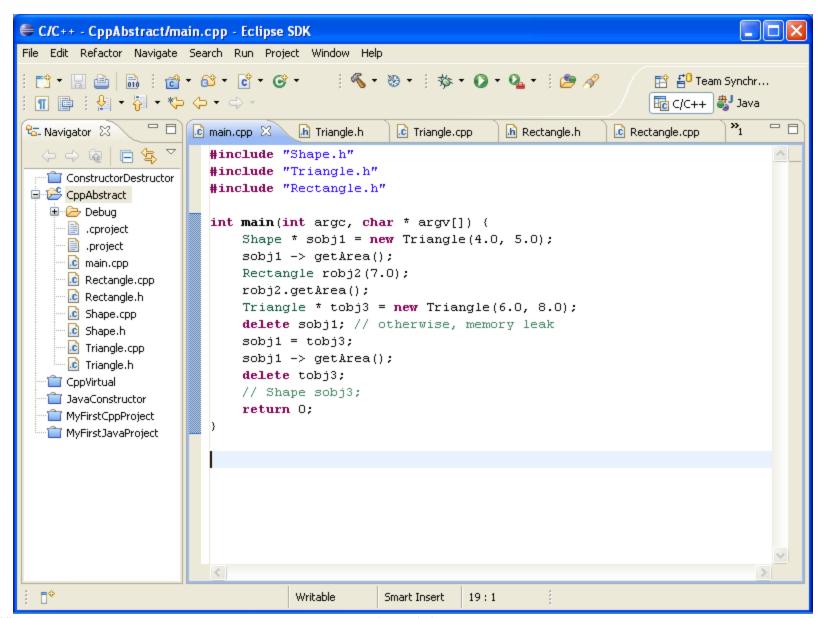


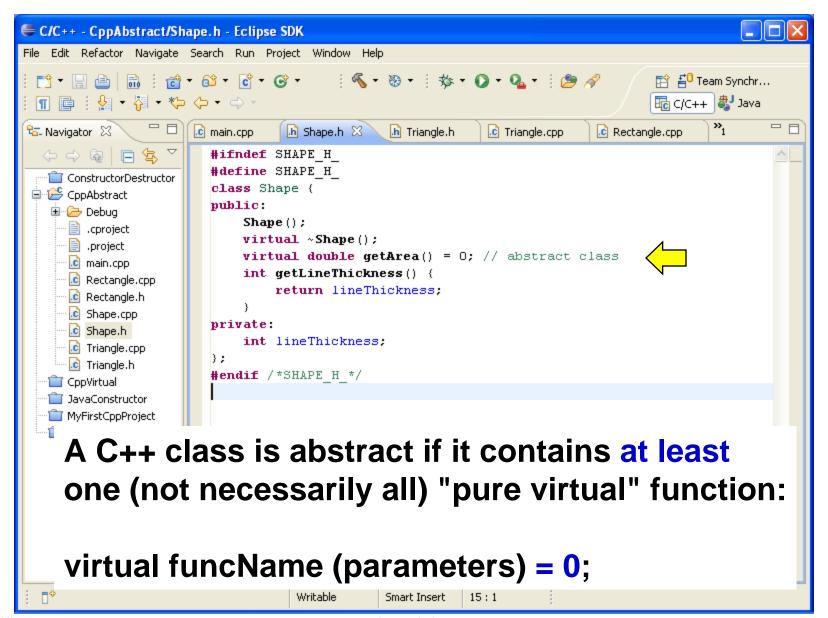
class	method	create object	override method in derived class
not abstract	not abstract	Y	not necessary
not abstract	abstract	Error	Error
abstract	abstract	No	Must
abstract	not abstract	No	not necessary

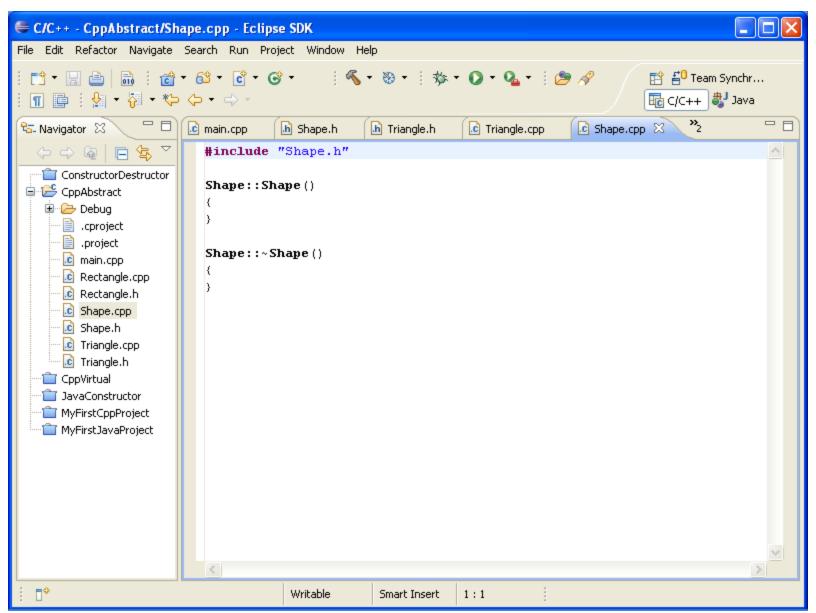
Examples of Abstract Classes



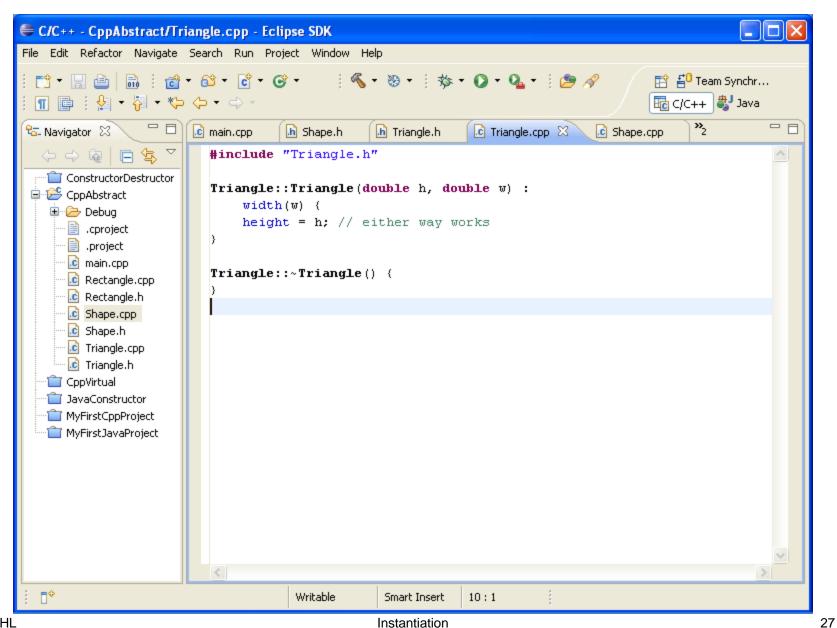
Abstract C++ Class

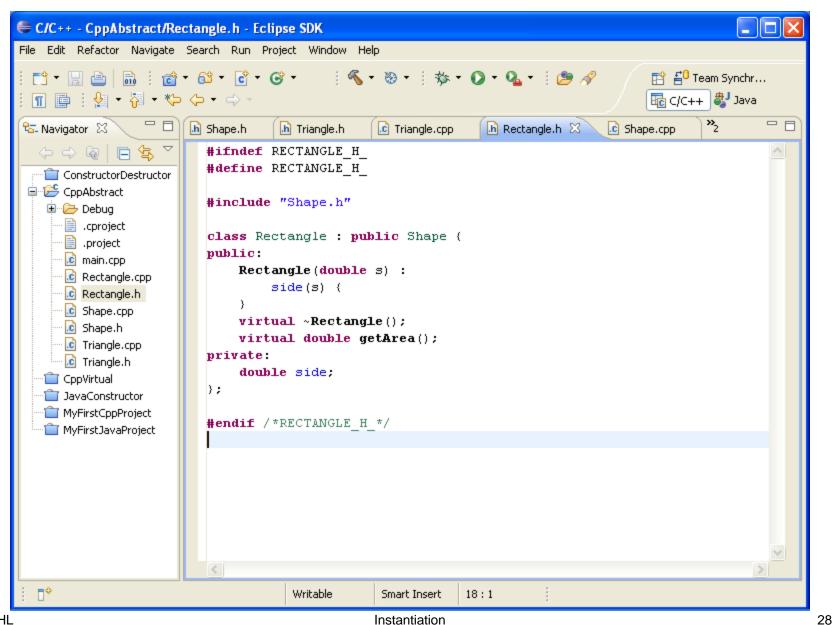


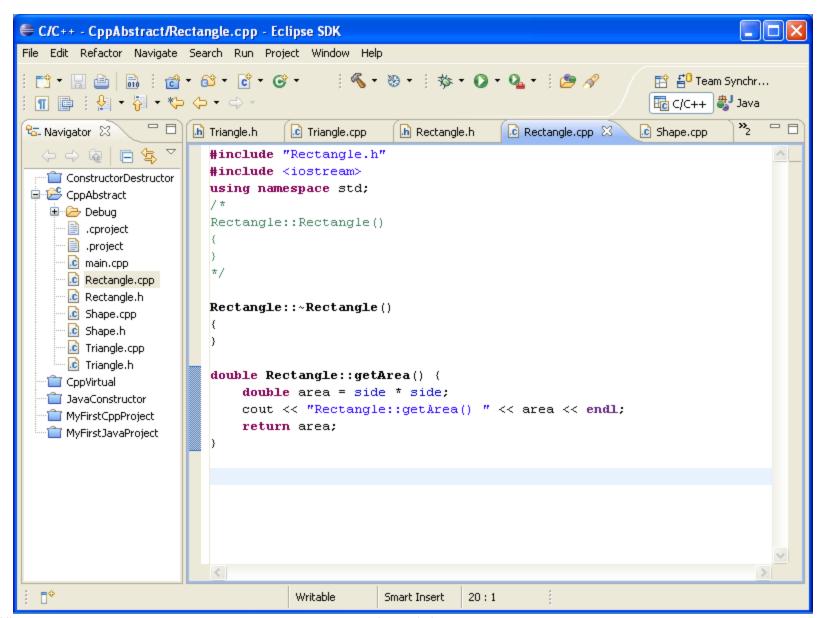


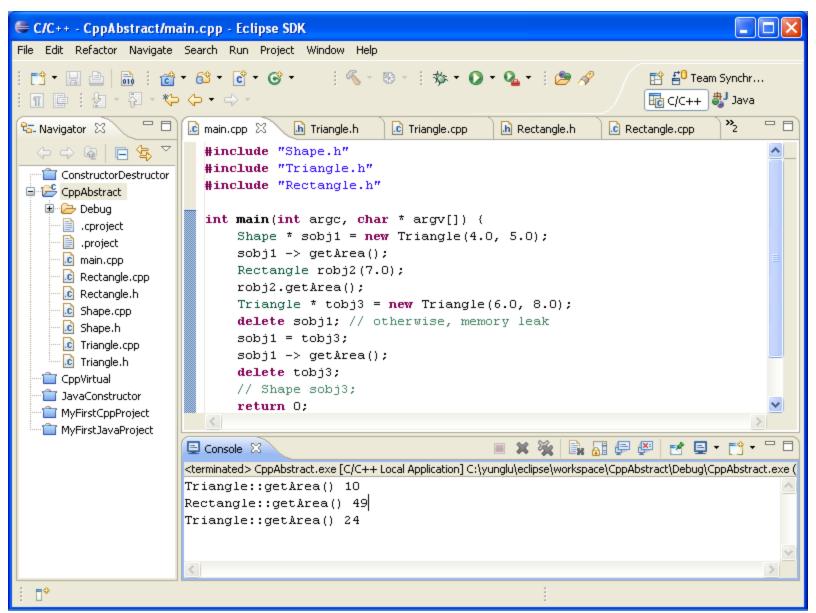


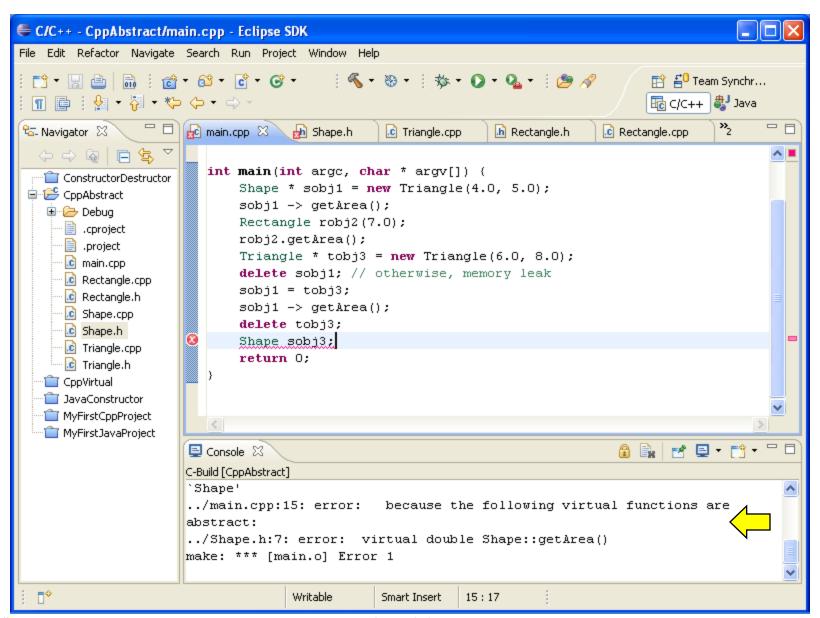
```
€ C/C++ - CppAbstract/Triangle.h - Eclipse SDK
                                                                                        File Edit Refactor Navigate Search Run Project Window Help
🔛 🖆 Team Synchr...
                                                                            ि C/C++ 🐉 Java
 \neg \sqcap
🔽 Navigator 🔀
                     main.cpp
                                h Shape.h
                                           h Triangle.h ≅
                                                        Triangle.cpp
                                                                      Shape.cpp
  #ifndef TRIANGLE H
                       #define TRIANGLE_H_
   ConstructorDestructor
 #include "Shape.h"
   🗓 🧁 Debug
                       #include <iostream>
     - cproject
                       using namespace std;
     project .
                       class Triangle : public Shape
     · 🚺 main.cpp
                       {
     · 🚺 Rectangle.cpp
                       public:
     · 🚺 Rectangle.h
                           Triangle (double h, double w);
     Shape.cpp
                           virtual ~Triangle();
     - 🔝 Shape.h
                           virtual double getArea() {
    Triangle.cpp
                               double area = 0.5 * height * width;
     🖟 🚺 Triangle.h
                               cout << "Triangle::getArea() " << area << endl;
   CppVirtual
                               return area:
   JavaConstructor
   MyFirstCppProject
                       private:
   MyFirstJavaProject
                           double height;
                           double width:
                       );
                       #endif /*TRIANGLE H */
  ₽
                                 Writable
                                            Smart Insert
                                                      23:1
```











Abstract Class

Java

C++

- abstract class
- abstract method (must be in an abstract class)
- does not need abstract method

 must have at least one pure virtual method

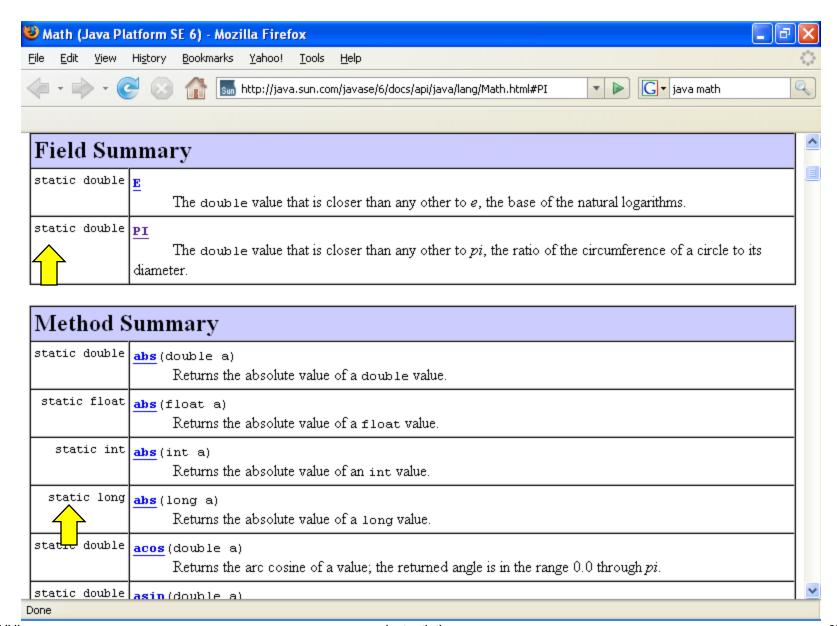
Static Member Unnecessary to Create an Object

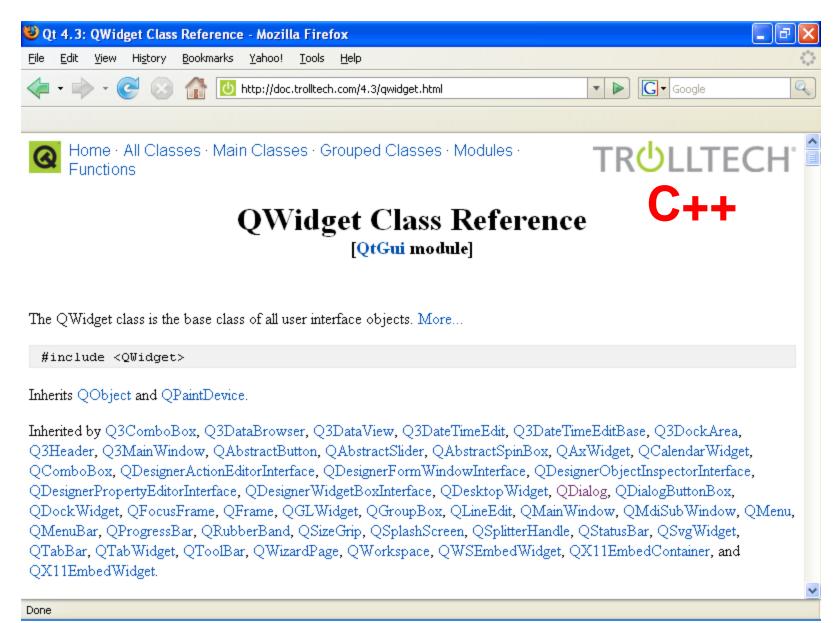
Static Member (Attribute or Method)

- A static attribute is shared by all objects of this class.
- A static function can be called without creating an object.
- applications:
 - constant, e.g. Math.PI in Java
 - global function, e.g. Math.random() and main in Java
 - shared attribute, e.g. interest rate for all SavingAccount objects
 - object counter (how many objects of this class has been created)

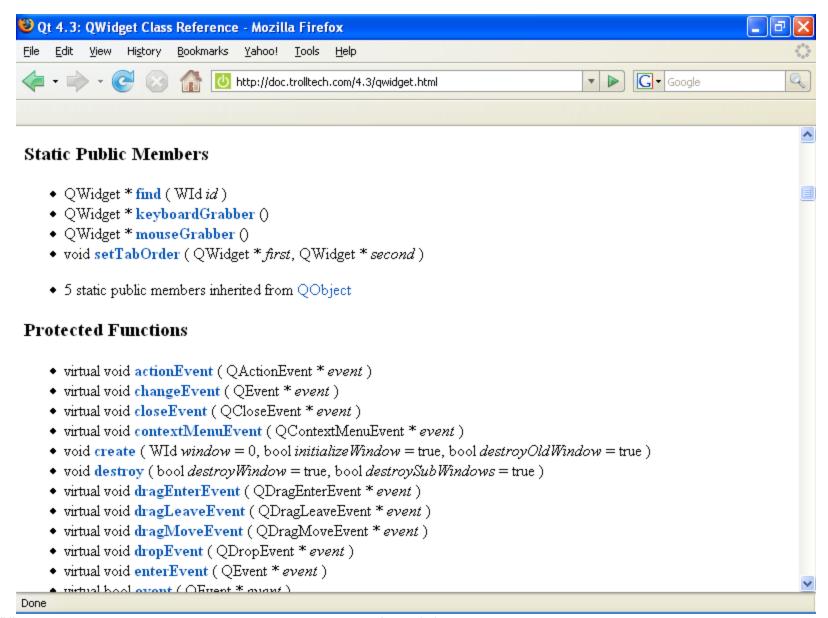
Examples of Static Members

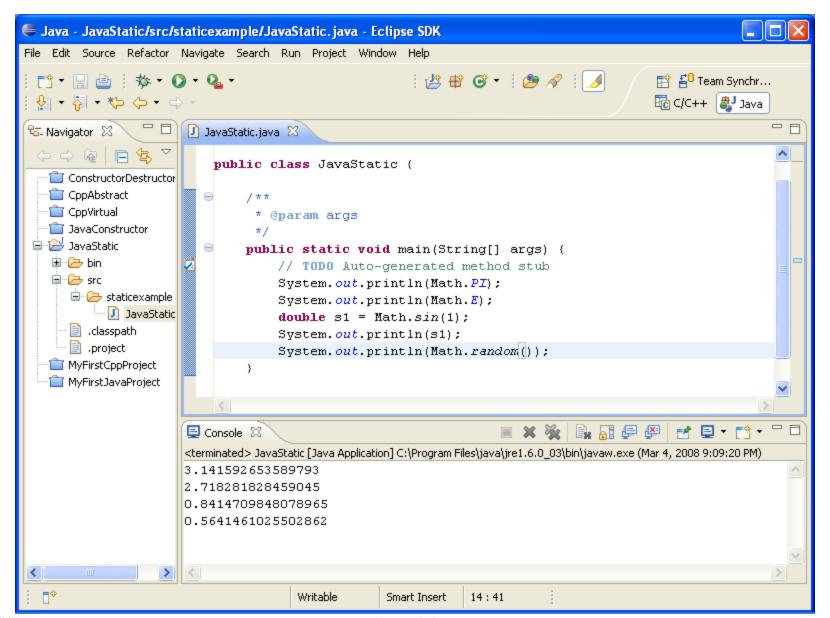




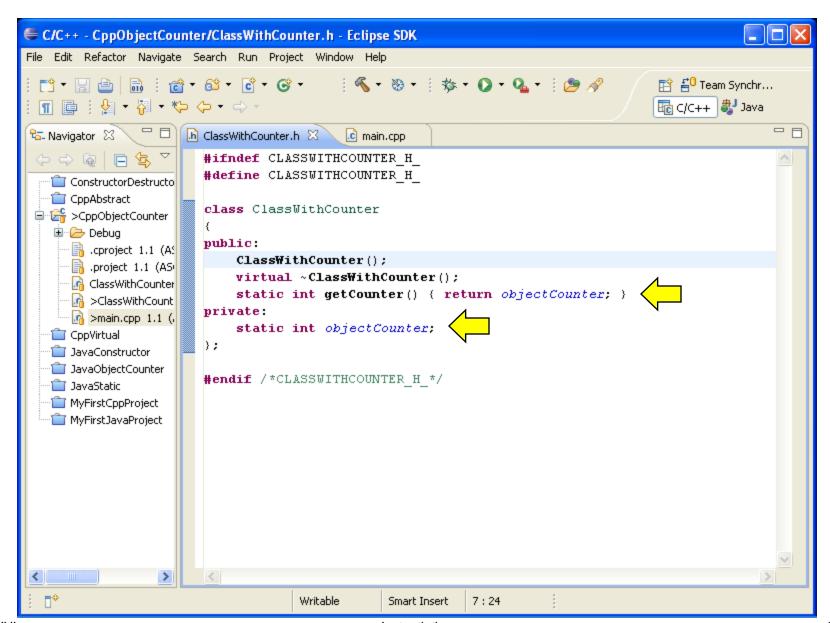


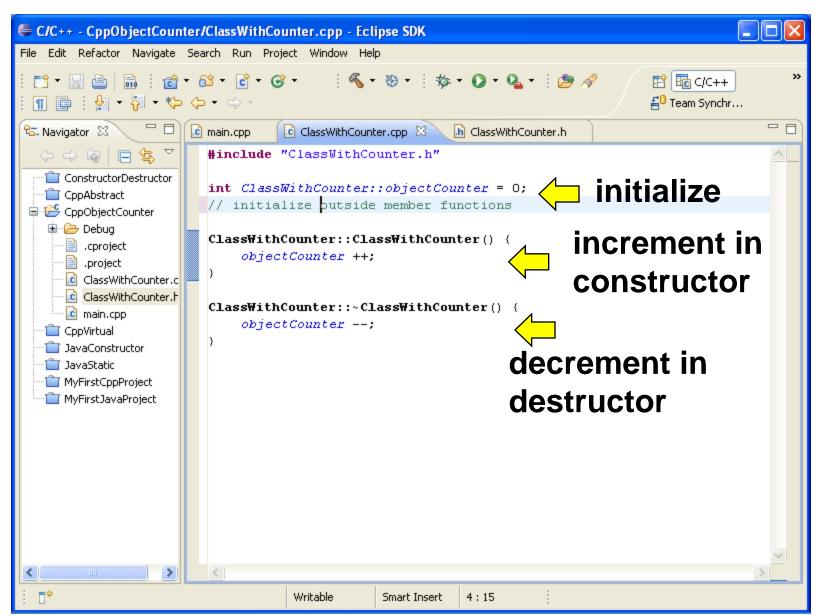
YHL

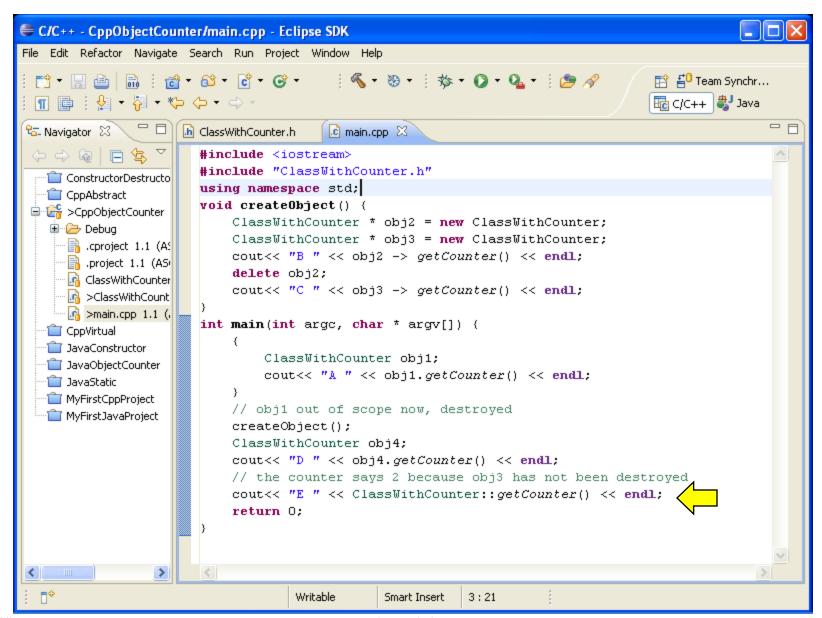


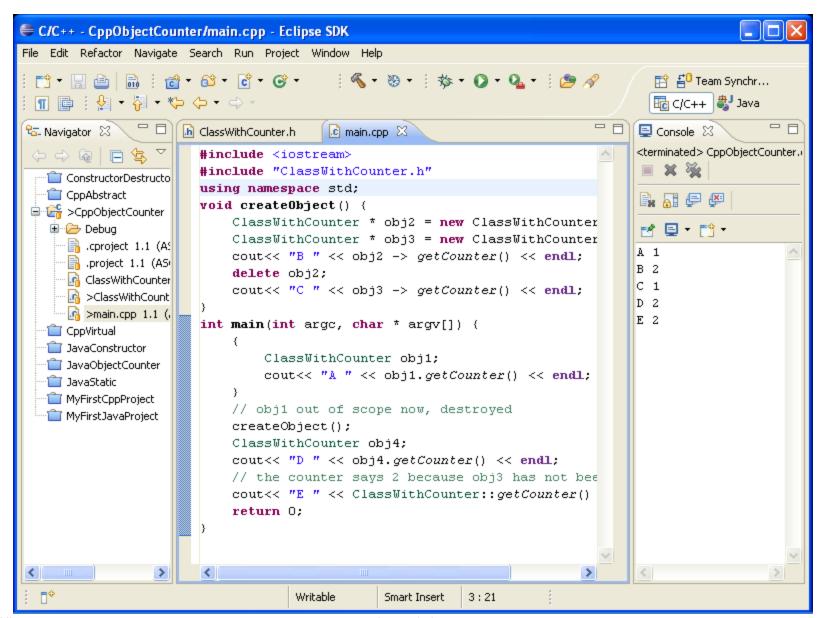


Object Counter in C++





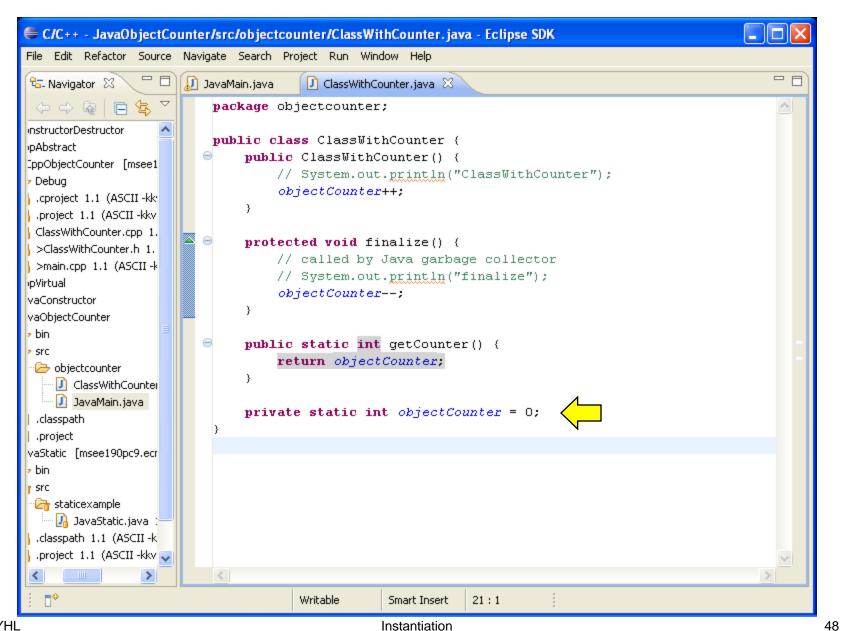


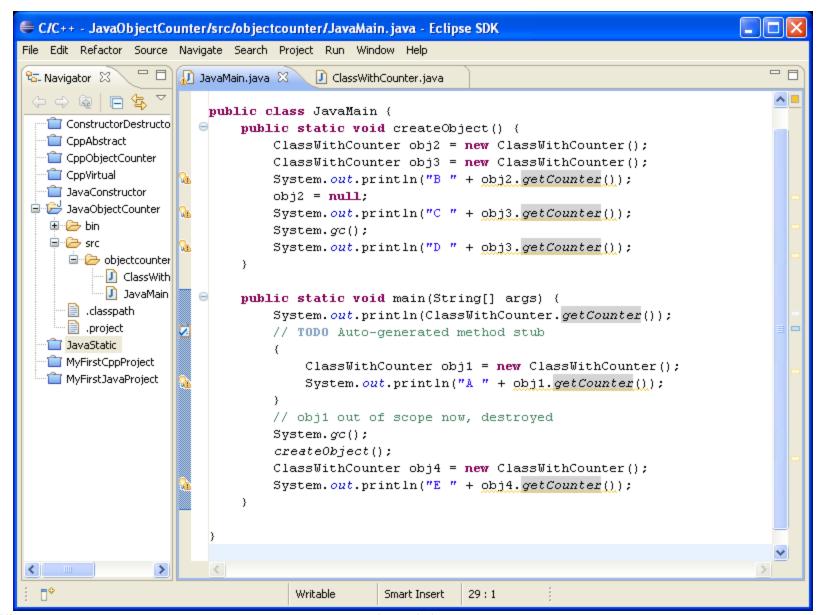


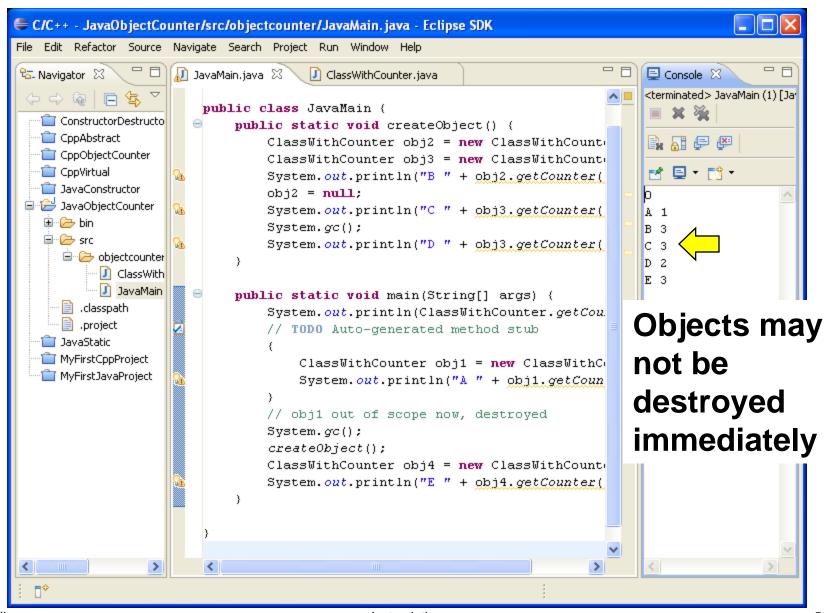
Object Counter in Java

Static Attribute in Java

- can be initialized when it is declared
- System.gc(); ⇒ encourage, but does not force, Java to collect garbage
- finalize() is called when an object is garbage collected









static function can access only static attributes non-static function can also access static attributes



YHL Instantiation 51

Self Test

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

Submission Procedure

Yung-Hsiang Lu yunglu@purdue.edu

Submission and Grading

- All submissions must be the CVS repository. Your submission will not be graded if "cvs co" fails.
- All submissions are graded in MSEE 190. Your code must be able to run on these machines.
- Crashed programs (including run-time exception) will not be graded.
- 20% penalty if memory leak is detected in C++ programming assignments (checked by valgrind).
- Late submission not accepted. Email submission not accepted. No excuse and no exception.
- All rules are firm and not negotiable.

Team Policy

- You can work with one (and only one) person that is also taking ECE 462 in this semester for a programming assignment. You can have different teammates for different assignments.
- Both must submit identical files and include a file called GROUP (not GROUP.txt nor GROUP.doc)
- GROUP includes the names, PID, and emails of both.
- If you have no teammate, please say so explicitly in README (not README.txt nor README.doc)
- Team is not allowed for lab exercises.

GROUP

Person A	Person B	GROUP	Grade
Submit	Submit Same	Submitted	Same
Submit	Submit Same	Not Submitted	Cheating ⇒ F
Submit	Submit Different	Not Submitted	work alone
Submit	Submit Different	Submitted	20% penalty of the average
Submit	Not Submitted	Submitted by A	50% penalty both
Submit	Not Submitted	Not Submitted	B receives 0

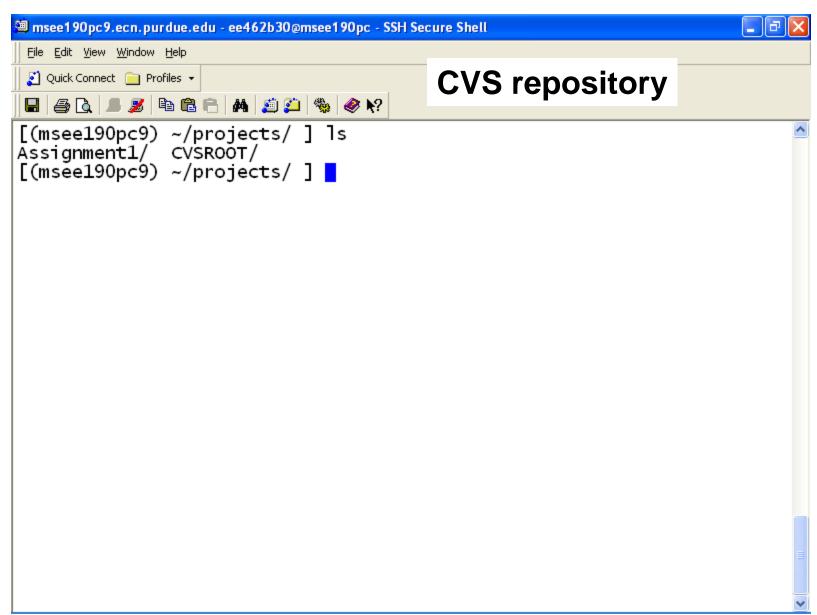
Policy Handling Dishonesty (Zero Tolerance)

- Code can be shared by teammates only.
- Copying code is strictly forbidden. Code similarity will be checked.
- Dishonesty, if discovered and verified
 - ⇒ Everyone involved will receive F for the course (not just the assignment).
 - \Rightarrow The case will be reported to the associate head.
 - \Rightarrow No exception will be given for any reason.
 - ⇒ Repeated offenders may be expelled from Purdue.
 International students may be expelled from US.

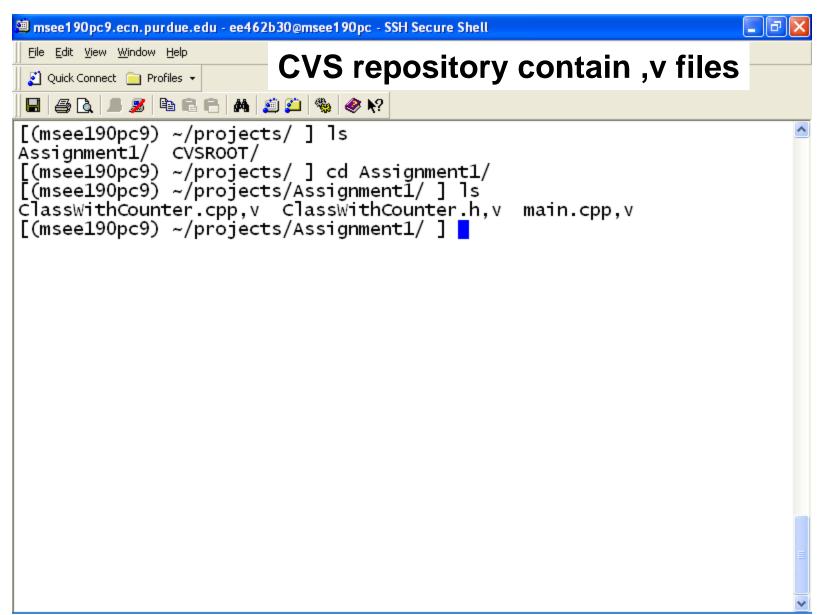
Submit Zip of CVS Repository

Procedure

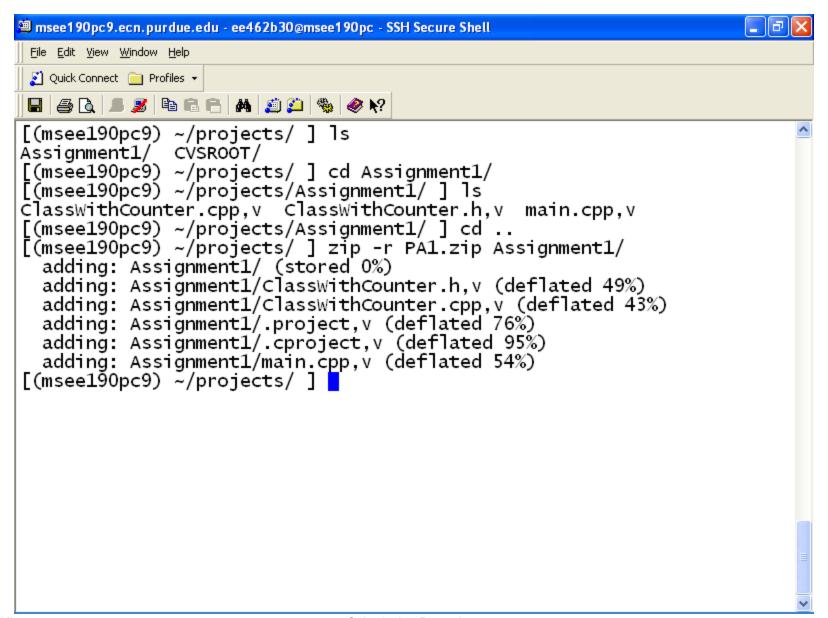
- enter your CVS repository
- find the assignment's name
- zip -r PAx.zip assignmentx
- upload PAx.zip in Blackboard
- submit



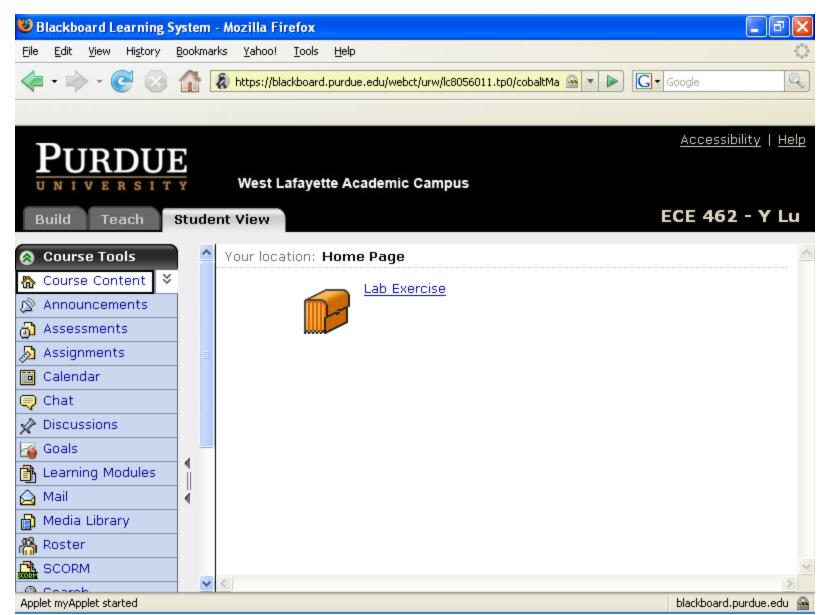
YHL

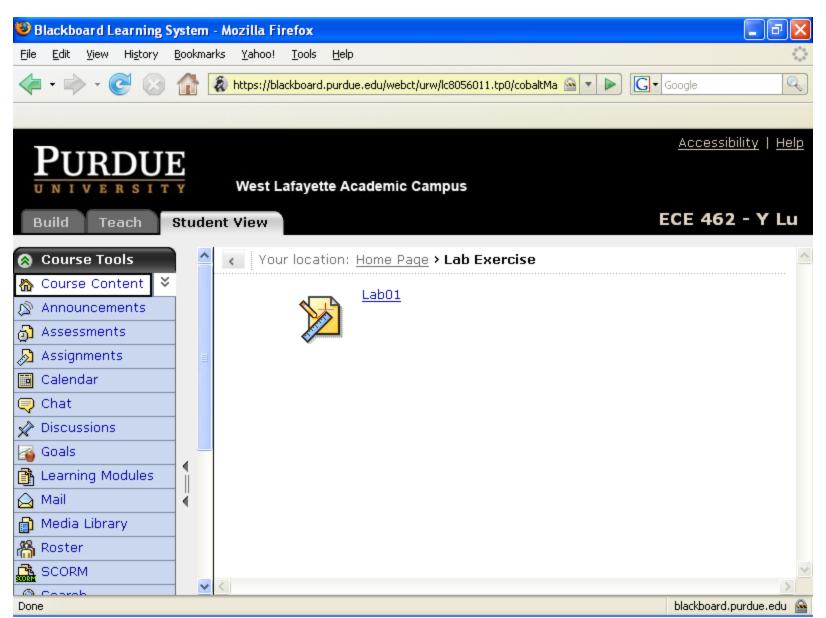


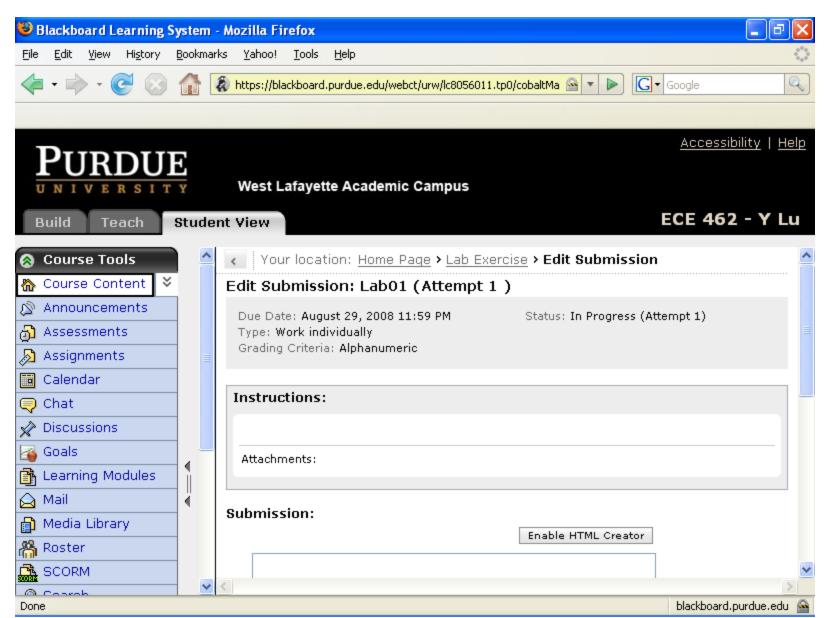
YHL

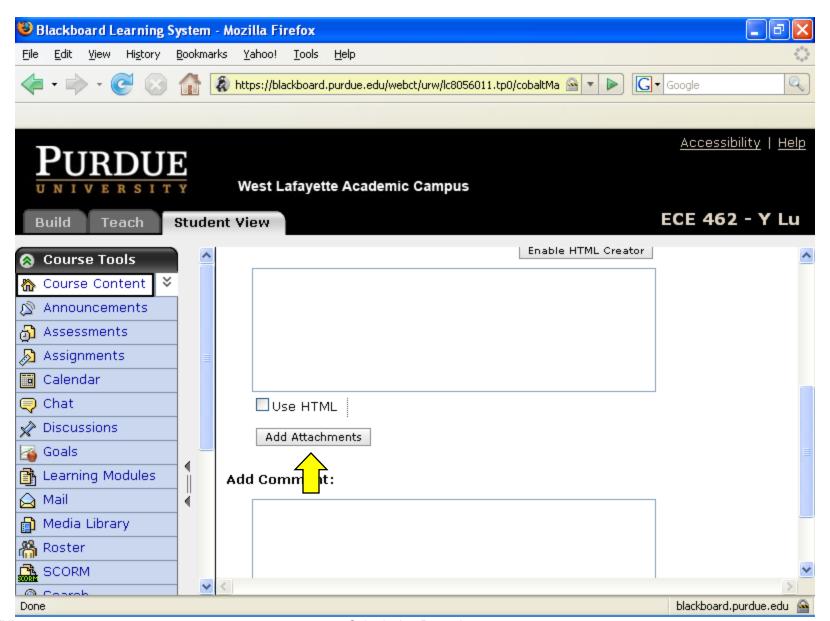


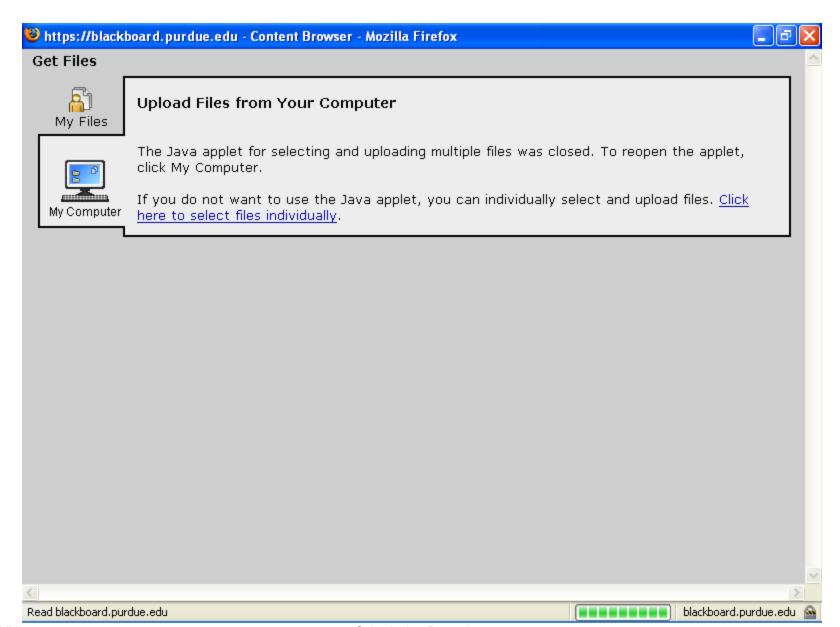
Submit in Blackboard

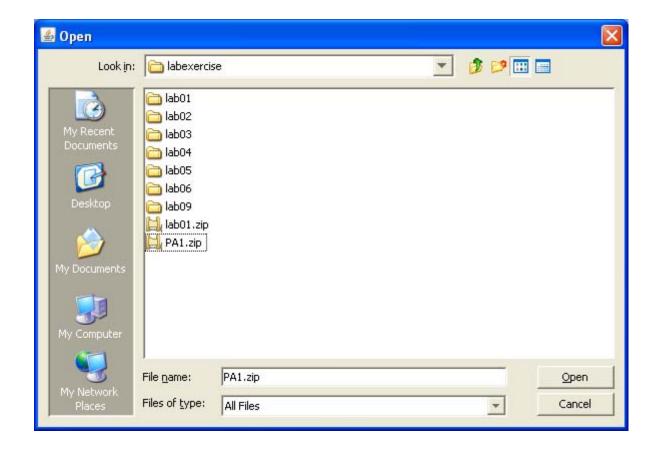


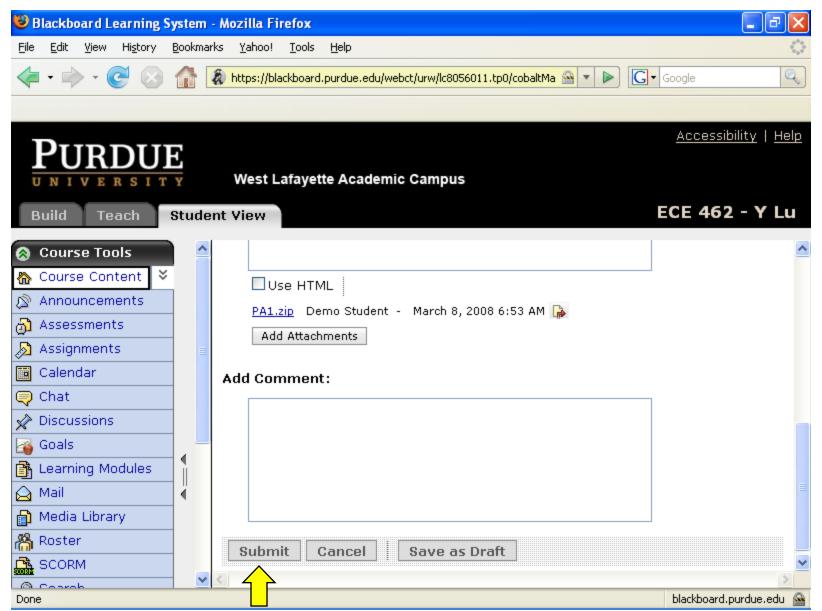


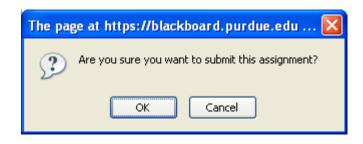


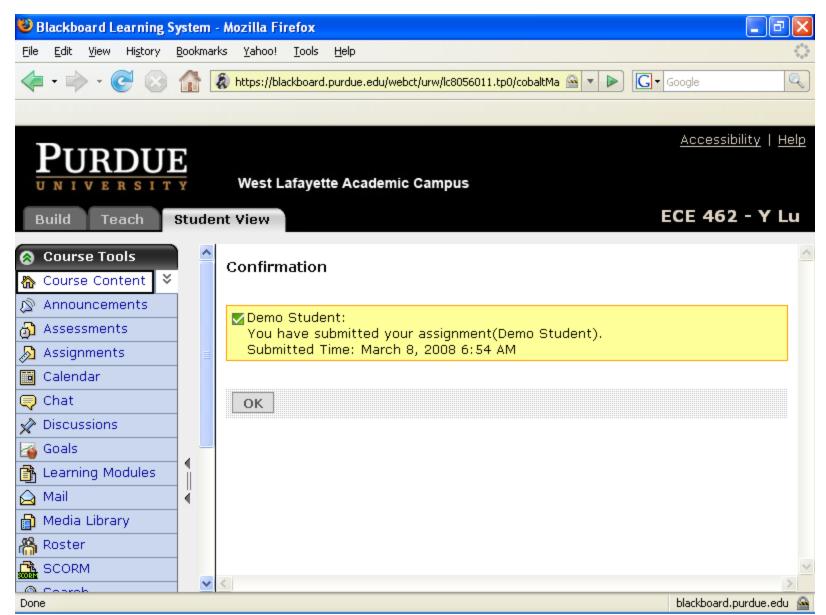




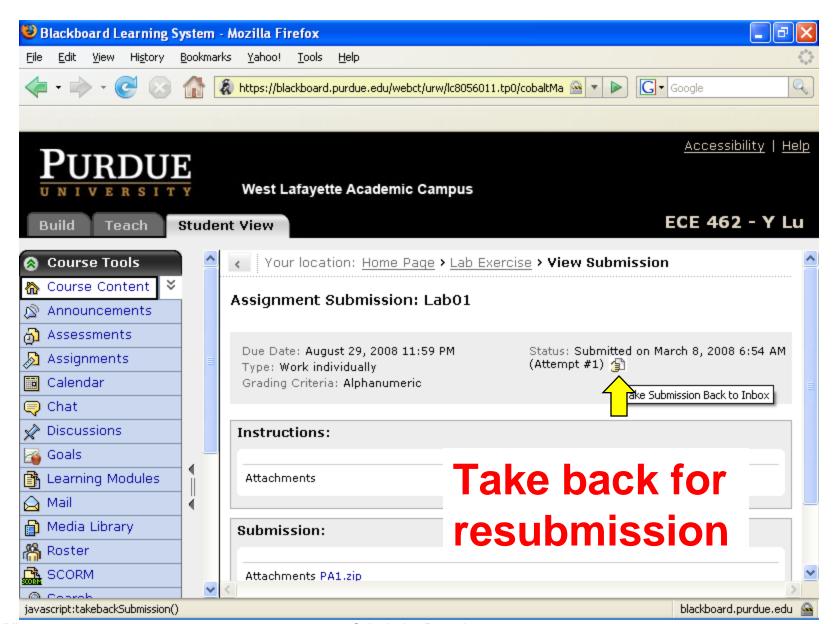


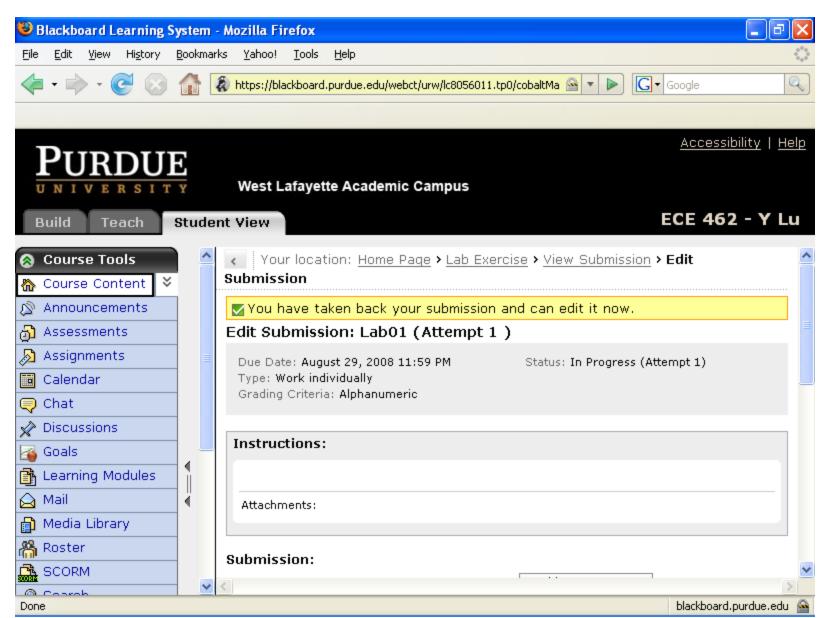


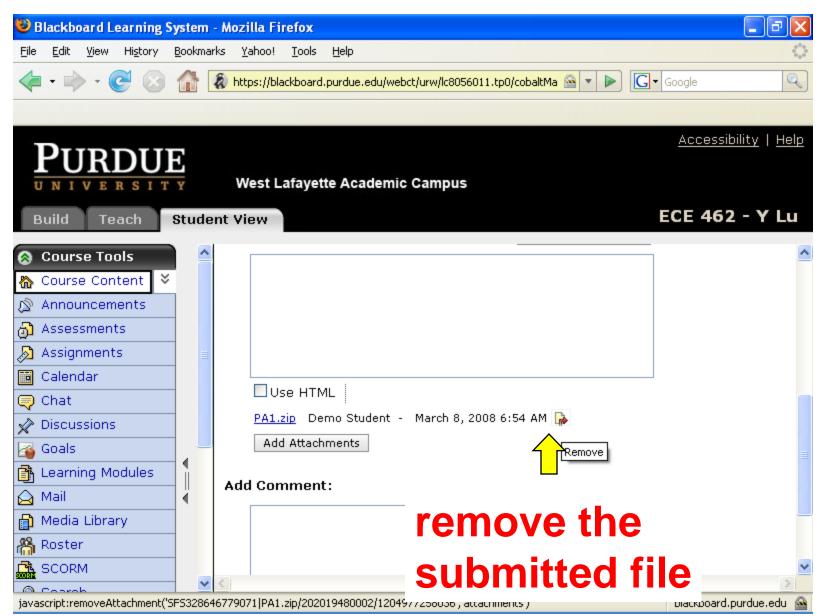


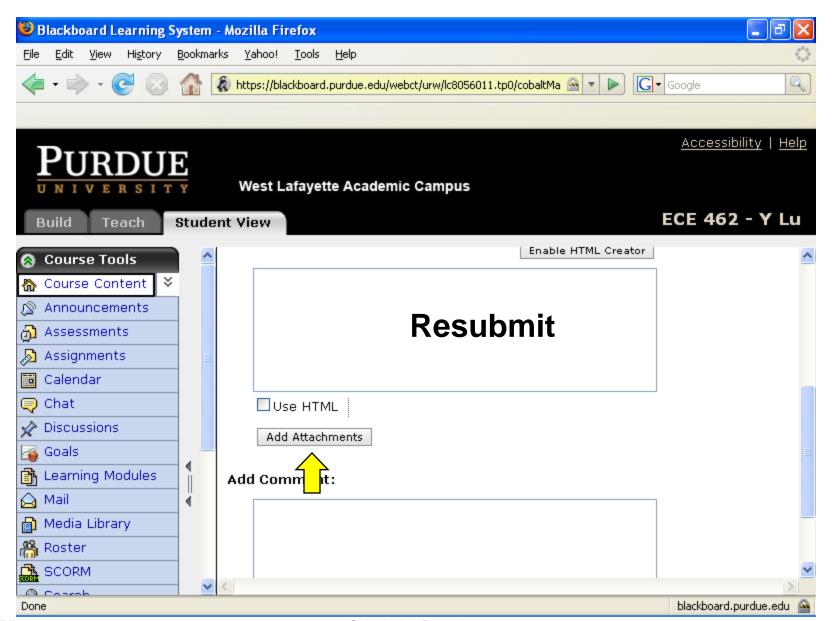


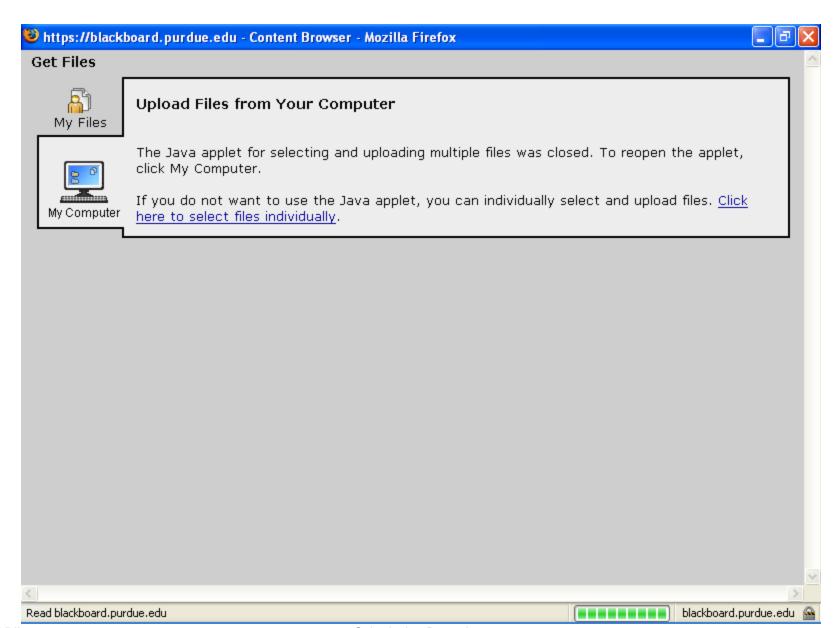
Take Back and Resubmit

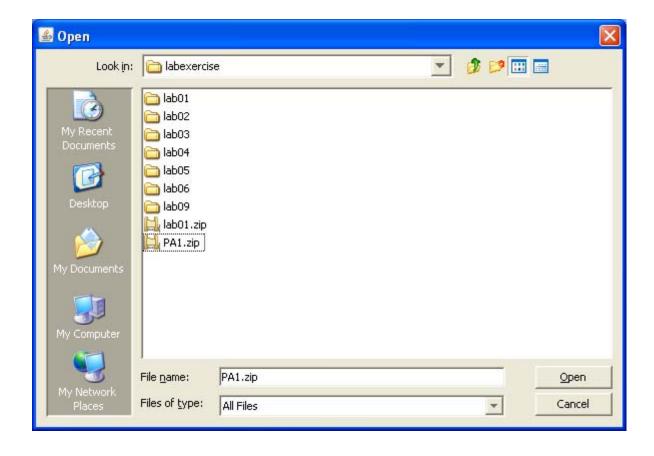


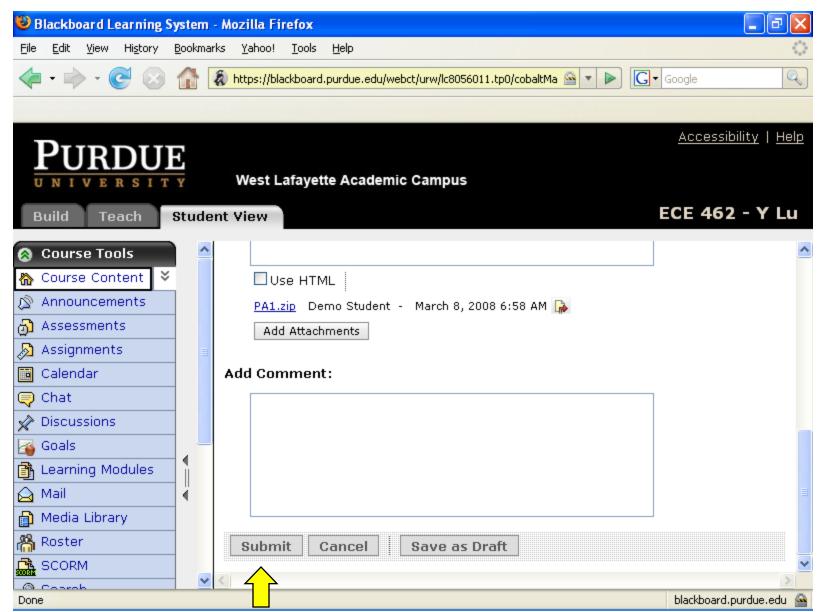


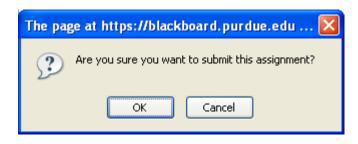








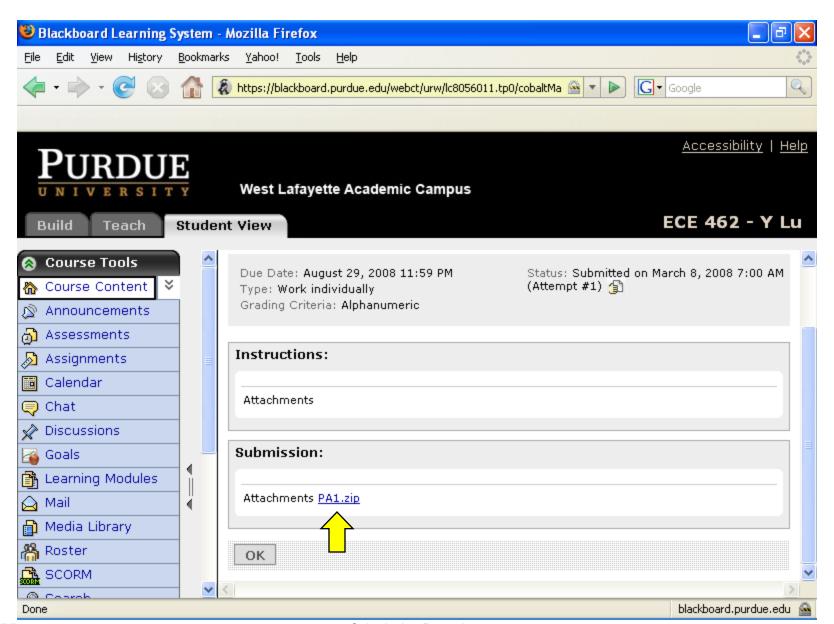




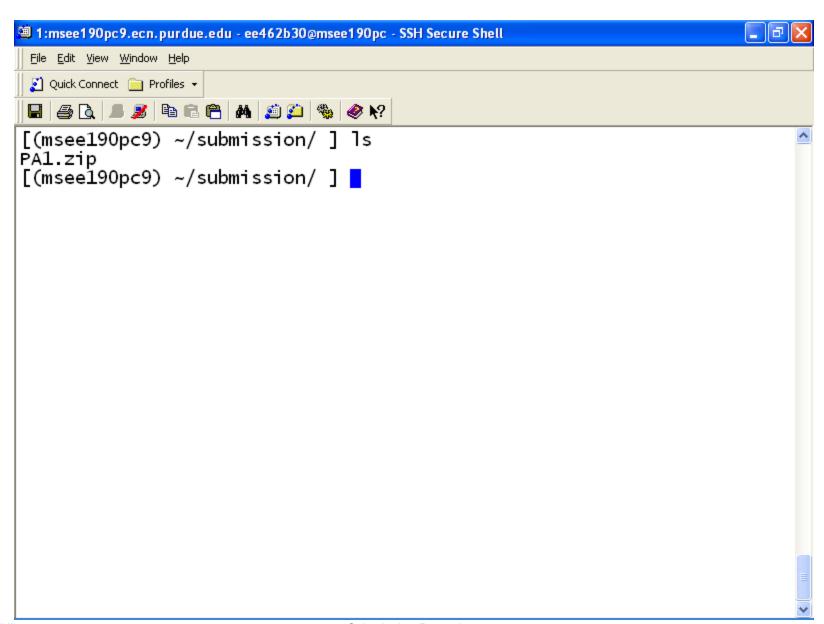
Verify Submission

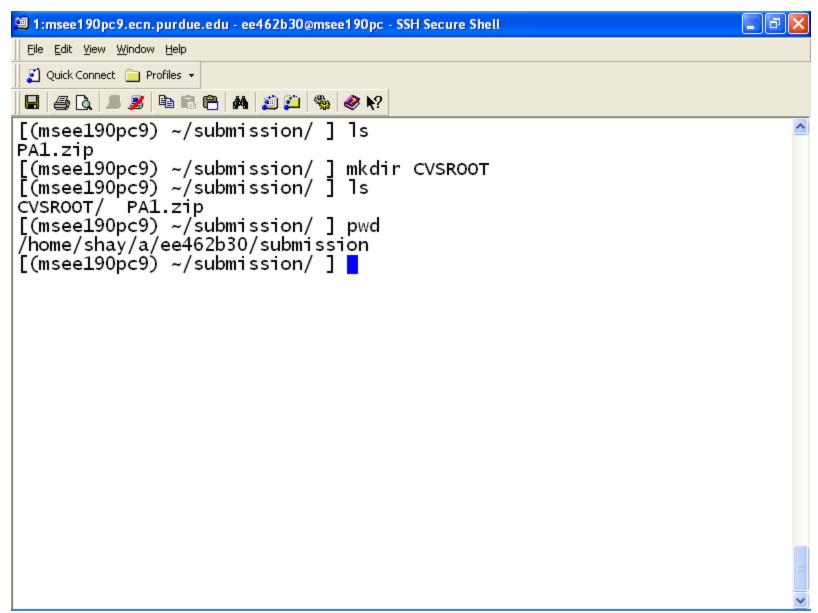
This step is **extremely important**.

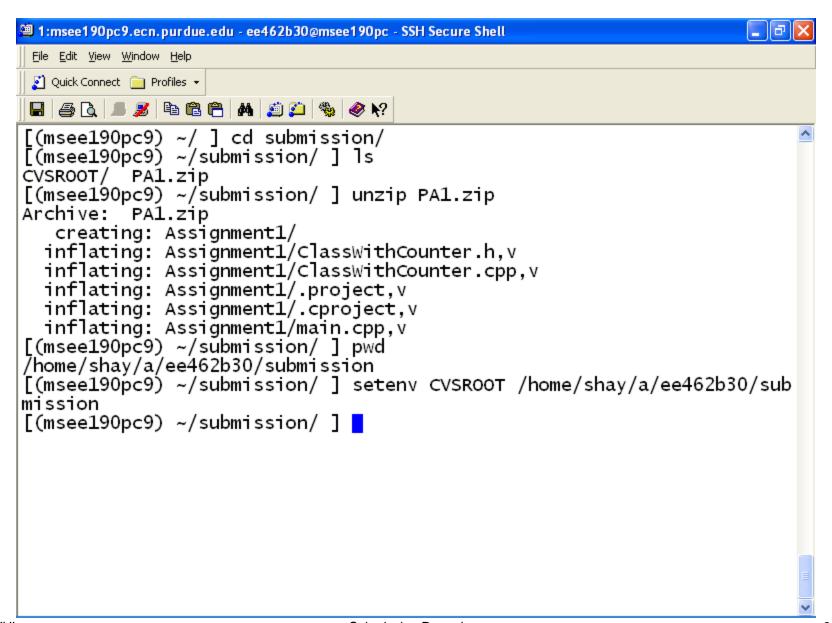
You must check whether you submit correct files.



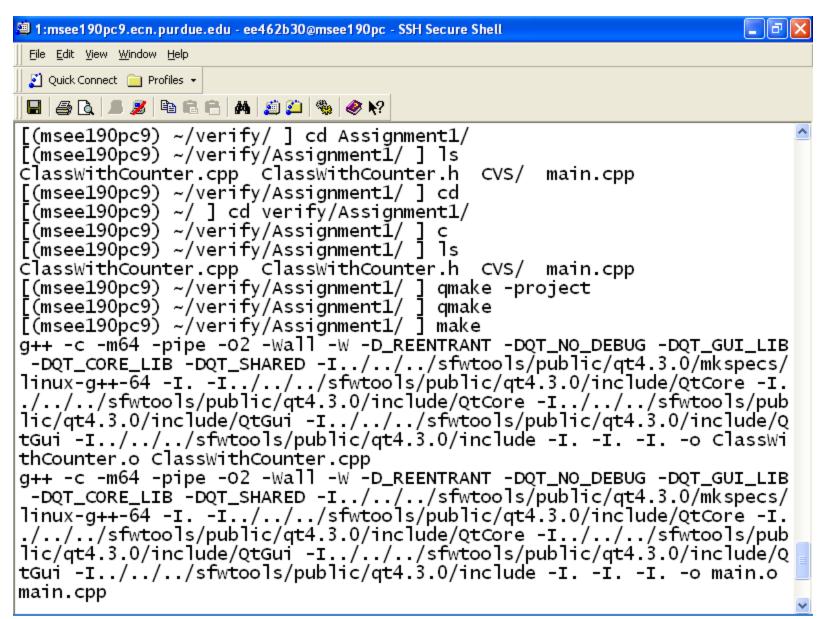




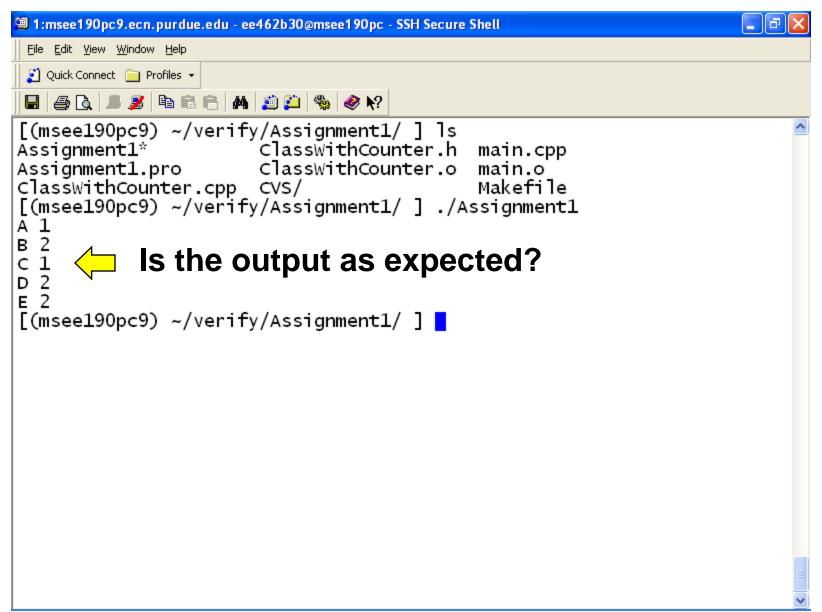




```
🕮 1:msee190pc9.ecn.purdue.edu - ee462b30@msee190pc - SSH Secure Shell
                                                                     _ 0 X
 File Edit View Window Help
 🗾 Quick Connect 📋 Profiles 🕶
[(msee190pc9) ~/submission/ ] ls
CVSROOT/ PA1.zip
[(msee190pc9) ~/submission/ ] unzip PA1.zip
Archive: PA1.zip
   creating: Assignment1/
  inflating: Assignment1/ClassWithCounter.h,v
  inflating: Assignment1/ClassWithCounter.cpp, v
  inflating: Assignment1/.project,v
  inflating: Assignment1/.cproject,v
  inflating: Assignment1/main.cpp,v
[(msee190pc9) ~/submission/ ] pwd
/home/shay/a/ee462b30/submission
[(msee190pc9) ~/submission/ ] setenv CVSROOT /home/shay/a/ee462b30/sub
mission
[(msee190pc9) ~/submission/ ] cd
[(msee190pc9) ~/ ] mkdir verify
[(msee190pc9) ~/ ] cd verify/
[(msee190pc9) ~/verify/ ] cvs co Assignment1
cvs checkout: Updating Assignment1
U Assignment1/.cproject
U Assignment1/.project
U Assignment1/ClassWithCounter.cpp
                                           Are these files expected?
U Assignment1/ClassWithCounter.h
U Assignment1/main.cpp
[(msee190pc9) ~/verify/ ]
```



YHL Submission Procedure 37



Submit Early. Submit (and Resubmit) Often. Do not miss the deadline.

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

Inheritance

Yung-Hsiang Lu yunglu@purdue.edu

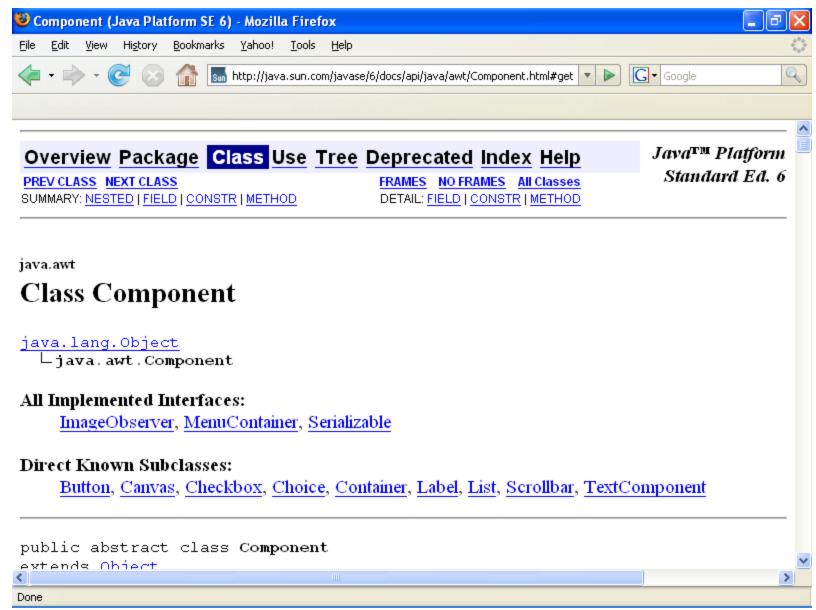
Block Inheritance

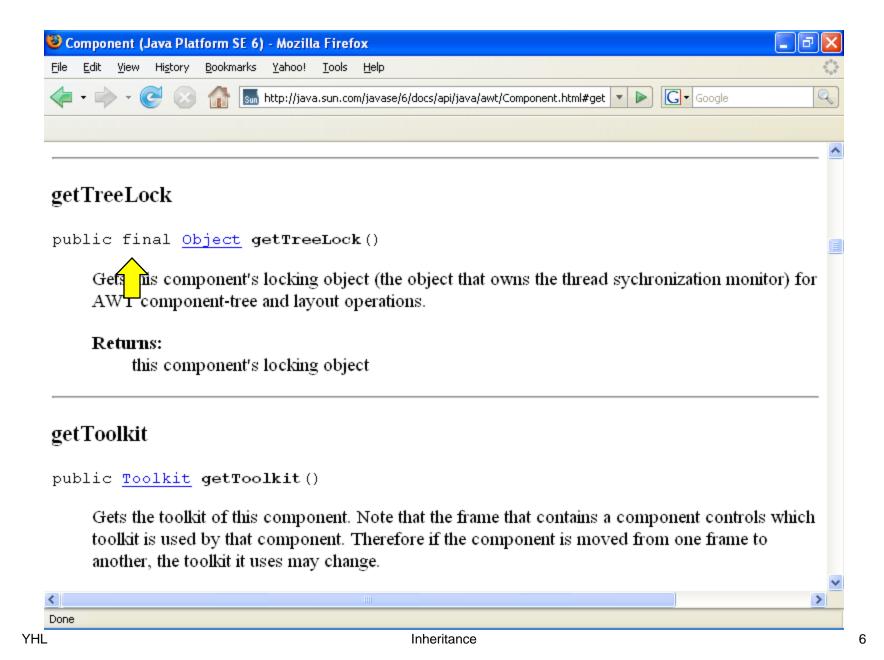
Why to Block Inheritance?

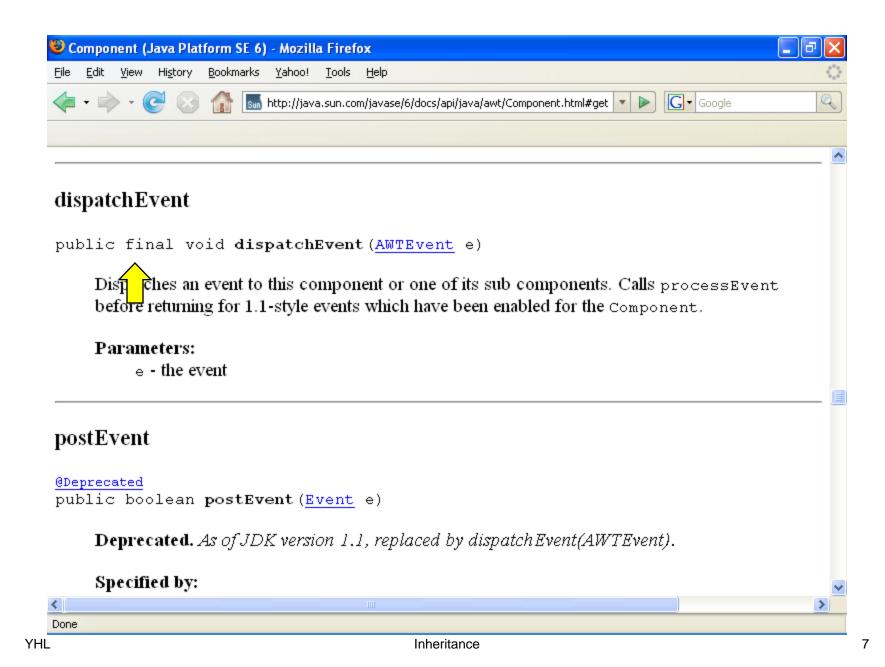
- If a class is a "leaf" in the class hierarchy and it cannot be further specialized:
 - Shape ⇒ Circle, Square, Triangle
 - BankAccount ⇒ CheckingAccount, SavingAccount
 - MotorVehicle ⇒ Sedan
- If a method must not be overridden in derived class:
 - setColor, getColor
 - setInterest

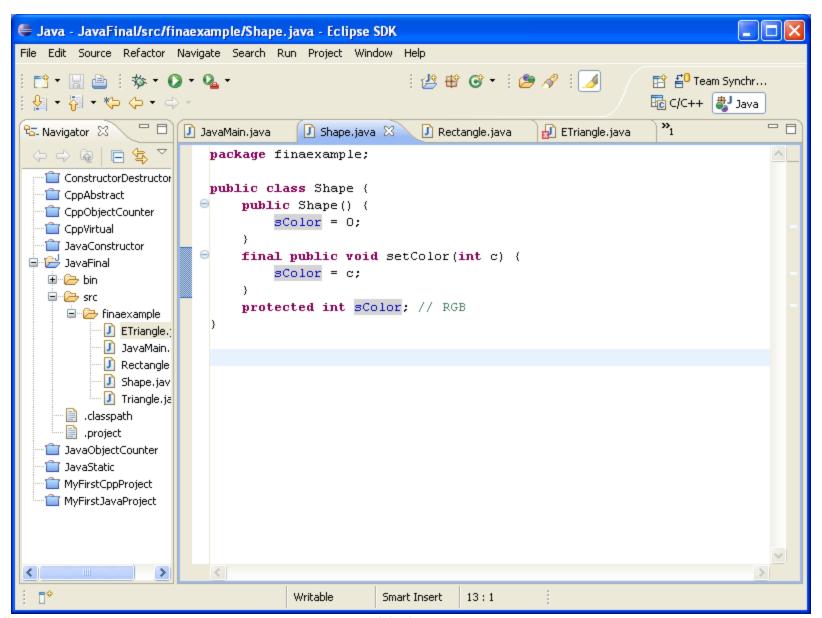
Block Inheritance in Java

- add final before class ⇒ this class cannot be extended
- add final before a method ⇒ this method cannot be overridden in derived classes
- function overloading: several functions have the same name but different parameters (numbers, types, or both)
 ⇒ final blocks only the function with the same parameters

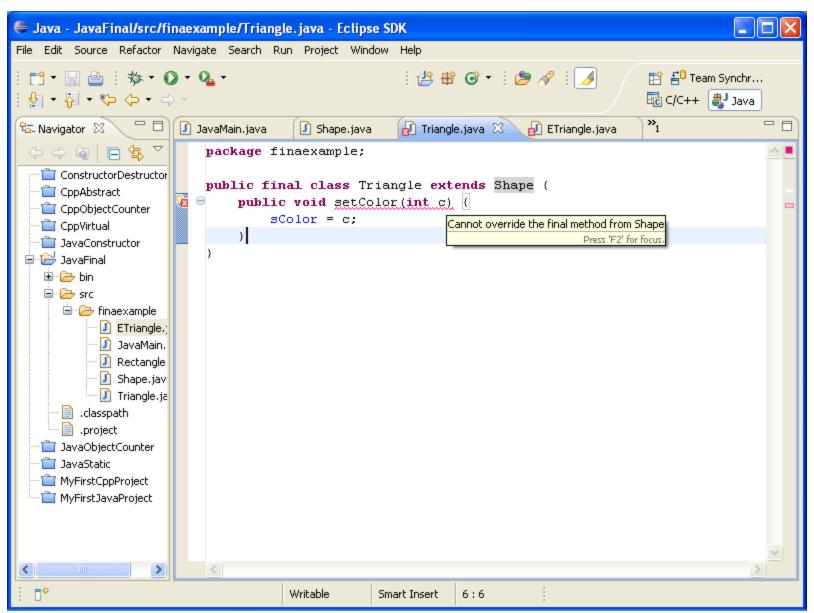


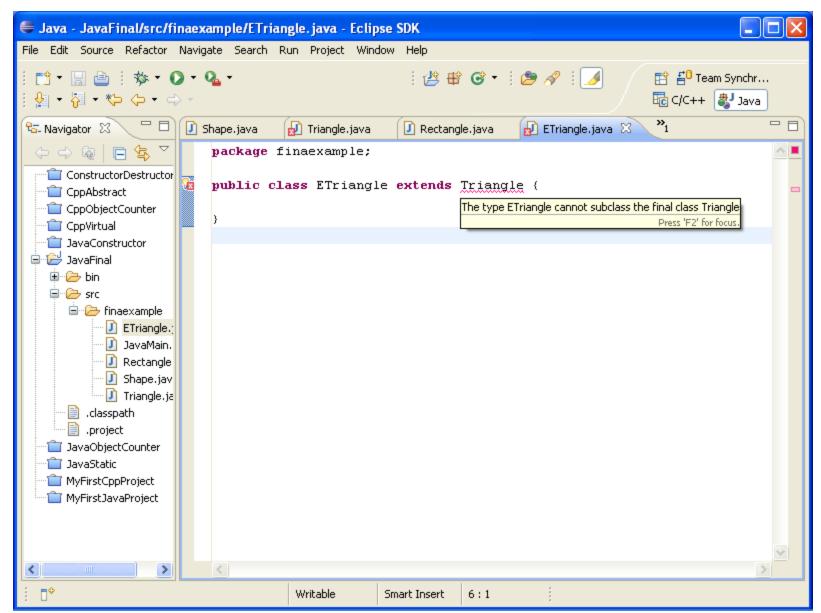


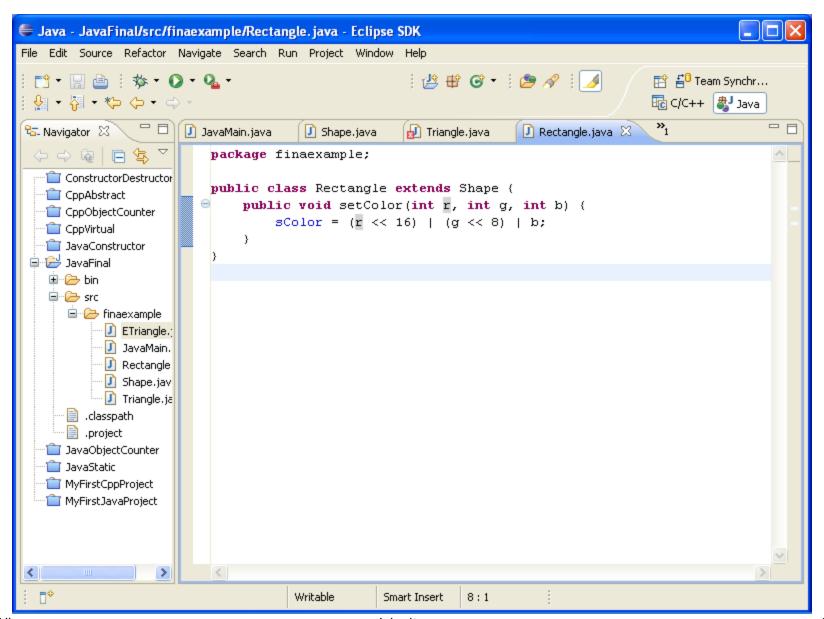




YHL Inheritance 8

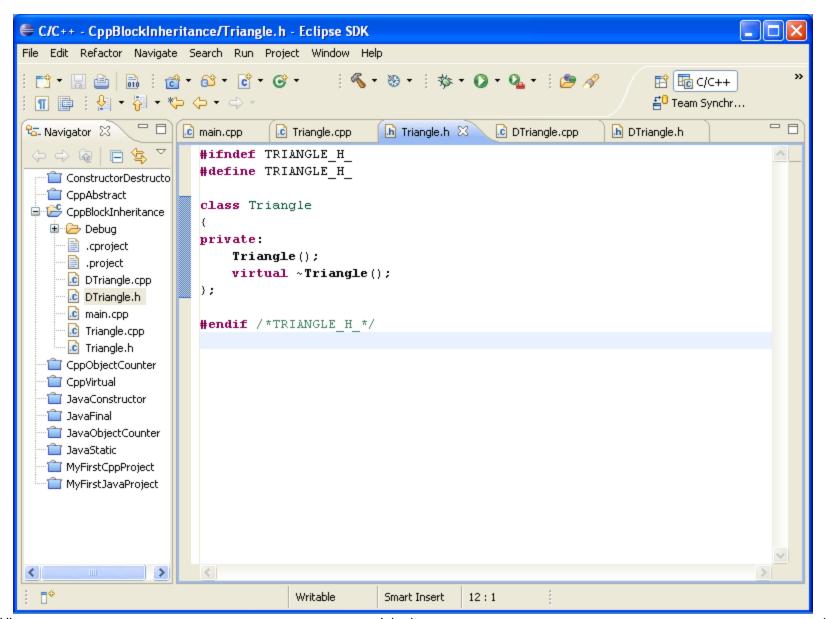


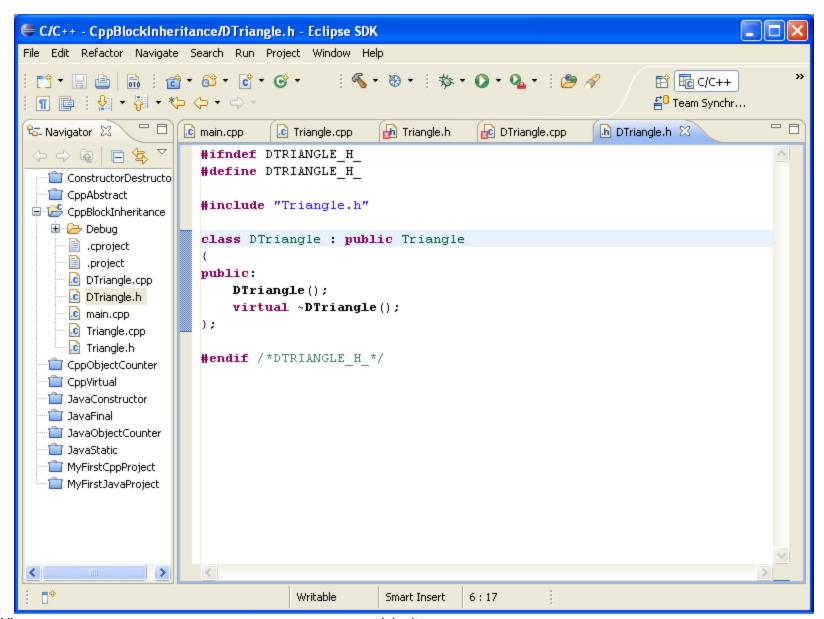


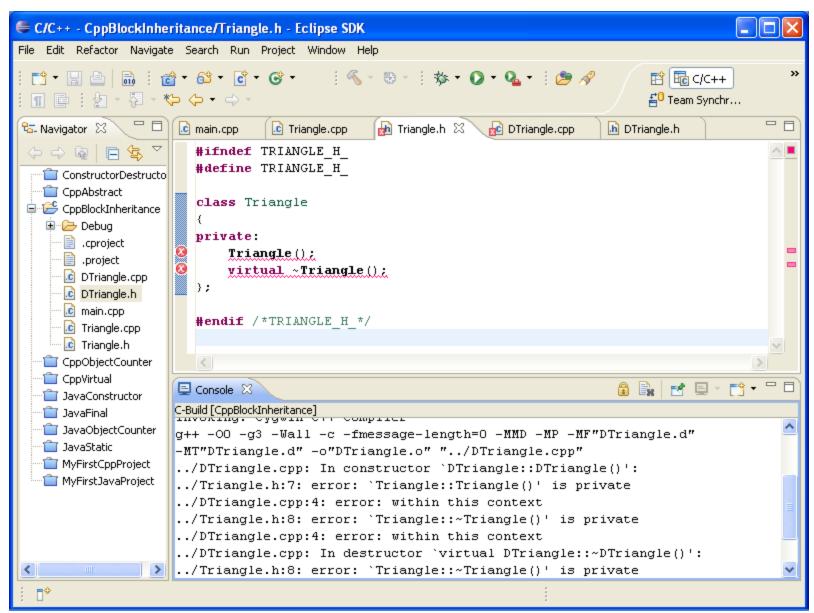


Block Inheritance in C++

- create a private constructor
- If constructors are overloaded, all of them have to be private.
- C++ does not allow blocking individual functions (in the same way as Java's final)



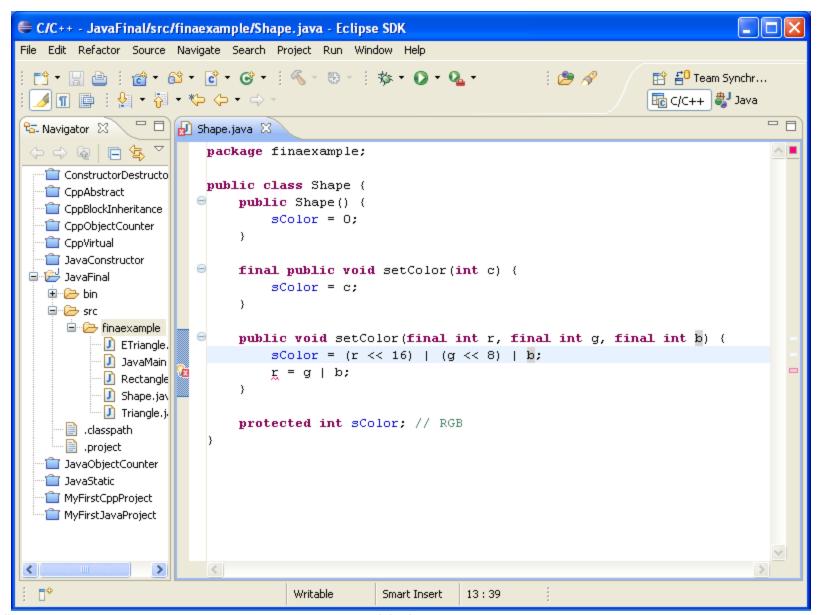


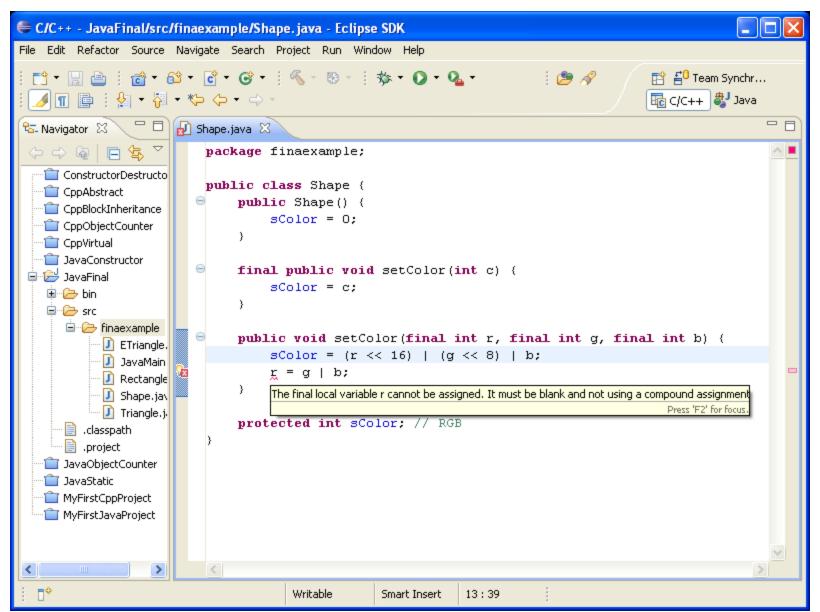


"Final" in Java

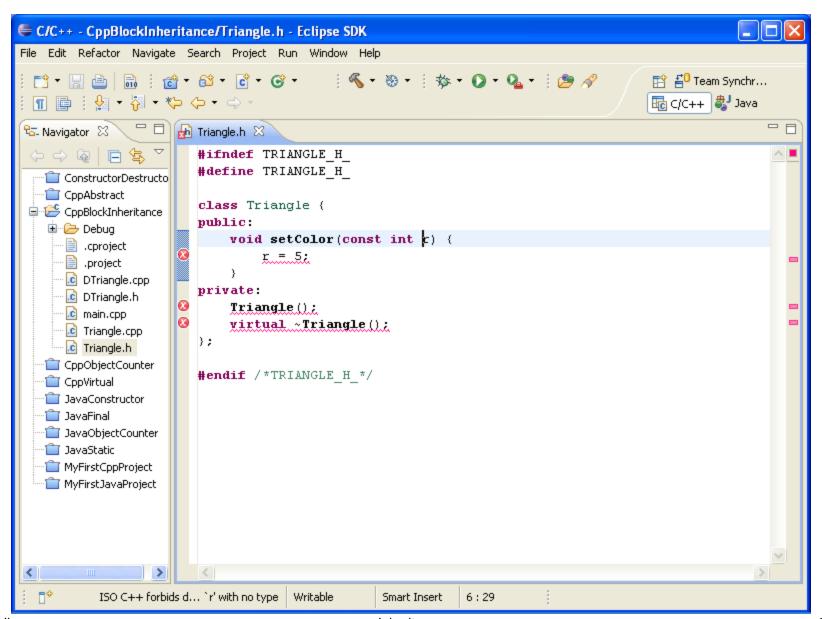
purpose	example
constant	final double PI = 3.1415926535;
block inheritance	final class Triangle {
block overriding	public final void setColor (
constant parameter	public void setColor (final int c)

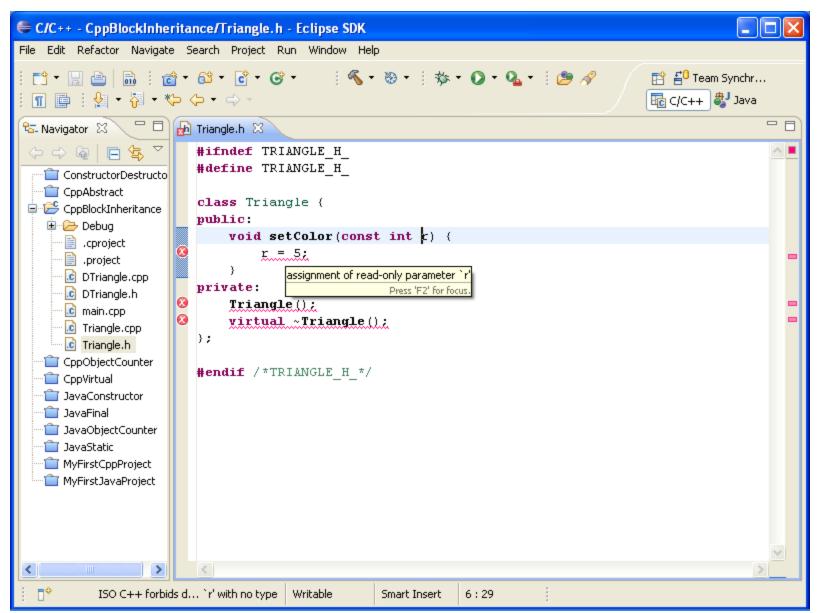
finalized() ⇒ called when a garbage object is collected



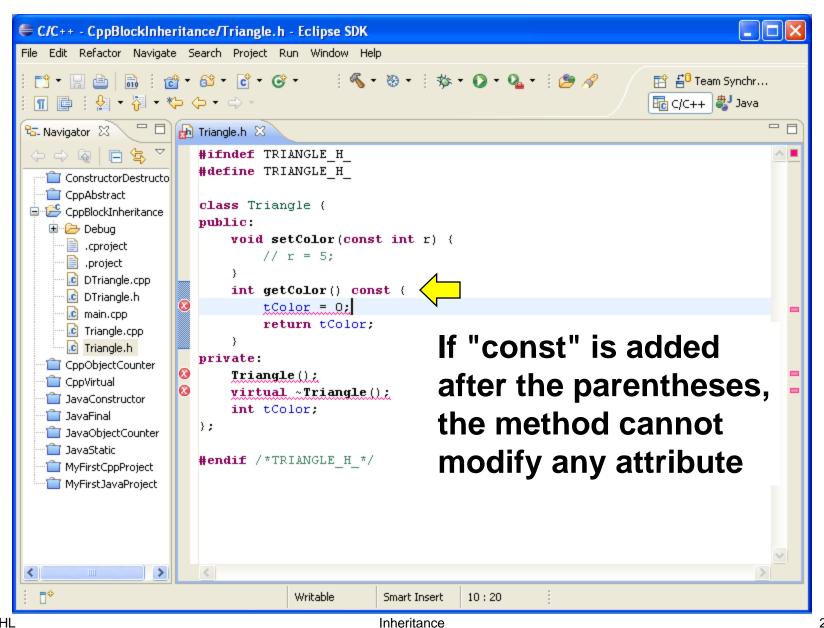


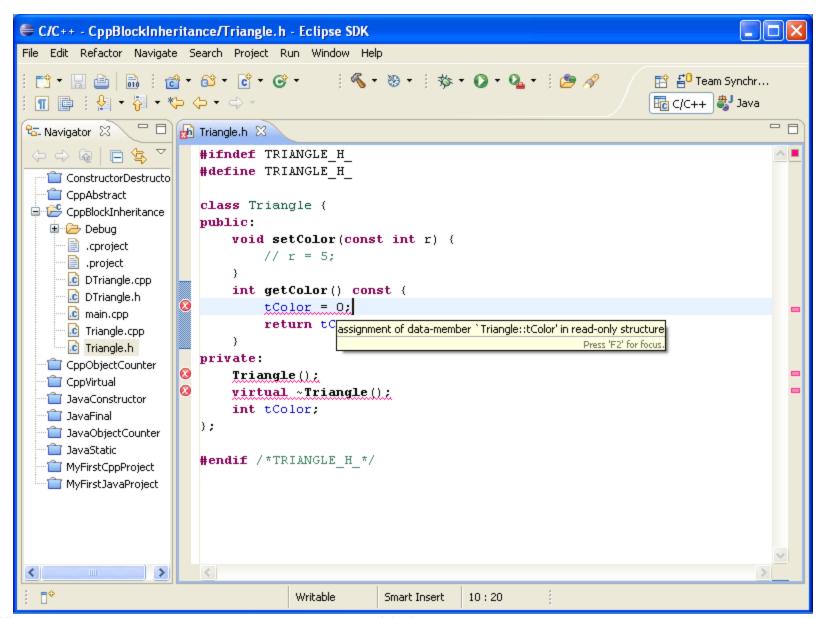
Constant Parameter in C++





Constant Function in C++





Self Test