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.
Revision control

From Wikipedia, the free encyclopedia

For the Wikipedia's revision control system, see Wikipedia:Revision control.

Revision control (also known as version control (system) (VCS), source control or (source) code management (SCM)) is the management of multiple revisions of the same unit of information. It is most commonly used in engineering and software development to manage ongoing development of digital documents like application source code, art resources such as blueprints or electronic models, and other critical information that may be worked on by a team of people. Changes to these documents are usually identified by incrementing an associated number or letter code, termed the "revision number", "revision level", or simply "revision" and associated historically with the person making the change. A simple form of revision control, for example, has the initial issue of a drawing assigned the revision number "1". When the first change is made, the revision number is incremented to "2" and so on.

Standalone version control systems mostly come from the software engineering industry, but revision control is also embedded in various types of software like word
List of revision control software

From Wikipedia, the free encyclopedia

This is a list of notable software for revision control.

Contents [hide]

1 Distributed model
   1.1 Open source
   1.2 Proprietary
2 Client-server model
   2.1 Local only
   2.2 Open source
   2.3 Proprietary
3 Notes
4 See also
5 External links
   5.1 Other comparisons
   5.2 Further reading
versions as arbitrary sets of revisions (resp. the deltas associated with them).

**Open source**

- **Concurrent Versions System (CVS)** — Originally built on the Revision Control System.
- **CVSNT** — A cross-platform port of CVS that allows case insensitive file names among other changes.
- **Fossil** — works as a command-line application, standalone HTTP server, or a CGI process launched from an existing HTTP server.
- **OpenCVS** — Compatible with CVS, with emphasis put on security and source code correctness.
- **Subversion** — An open source version control tool started in 2000 to be "a compelling replacement for CVS"[1].
- **Vesta** — A build system with a versioning file system and support for distributed repositories. Used at Intel for microprocessor design.

**Proprietary**

- **AccuRev** — A fast and easy-to-use SCM tool with integrated issue tracking based on "Streams" that efficiently manages parallel and global development. A replication server is also available.
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
Apple Developer Connection

CVS

CVS is the Concurrent Versions System. It is a source control tool which allows multiple people to simultaneously view and edit code.

CVS keeps a history of all changes that have been made to the code, along with who made the change and when it was committed into the repository.

Note: Not all Apple open source projects are available via CVS.

Getting CVS

For Mac OS X users, the CVS client is available after installing the Developer Tools.
Open Visualization Data Explorer

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.
SourceForge.net is the world's largest Open Source software development web site. SourceForge.net provides free hosting to Open Source software development projects with a centralized resource for managing projects, issues, communications, and code.

Registered Projects: **170,662**  Registered Users: **1,799,377**
FreeMind

A mind mapper, and at the same time an easy-to-operate hierarchical editor with strong emphasis on folding. These two are...
Firebird

Firebird RDBMS offers many features & runs on Linux, Windows, and several Unix platforms. Features include concurrency & performance.

YHL Version Control 12
Those doing active development can check out the latest source using CVS. This is the preferred method if you plan to provide patches and fix bugs, as it lets you get up-to-the-minute changes and merge them with your own.

If you want to compile a product for release, it is generally better to Download Mozilla Source Code tarballs.
All programming assignments and lab exercises **must** be submitted using CVS
Create a CVS Repository

Need to do this only once.
YHL Version Control 16

in your ee462xxx account

remember this path

in your ee462xxx account
start Netbeans
create a project
New Project

Steps
1. Choose Project
2. ...

Choose Project

Description:
Creates a new Java SE application in a standard IDE project. You can add files in the project. Standard projects use an IDE-generated Ant build script for your project.
package javaapplication17;

public class Main {

    public static void main(String[] args) {
        // TODO code application logic here
    }
}
package application17;

class Main {
    static void main(String[] args) {
        // TODO code application logic here
    }
}

If you don't see this template, choose Tools | Templates to add the template in the editor.
path from pwd
Import Project Options

CVS Root
Specify location of CVS repository defined by CVS root.

CVS Root: p30@msee190pc5.ecn.purdue.edu:/home/shay/a/ee462b30/projects
(:ext:username@hostname:/repository_path)

- Use Internal SSH
  Password: *********
  Remember Password
  [ ] Proxy Configuration...

- Use External Shell
  Shell Command: [ ] Browse...

[ ] Next > [ ] Finish [ ] Cancel [ ] Help
always give a meaningful message
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package javaapplication17;

/*
 * ECE462 sample program
 */

/***
 *
 * @author yunigu
 */

public class Main {

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

}
one file has been changed
show the difference
package javaapplication17;

/* ECE462 sample program */

public class Main {
    /* @param args the command line arguments */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
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.
package javaapplication17;

/**
 * ECE462 sample program
 */

public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
Change this template, choose Tools | Templates to open the template in the editor.
```java
package javaapplication17;
/
/*
 * ECE462 sample program
 */
/**
 */
```

no difference

<No Local/Remote Changes>
public static void main(String[] args) {
    // TODO code application logic here
}
The Preview Design button (in the toolbar) enables you to test the design of the form.
Use the Connection Mode button (in the toolbar) to establish a connection between components.
<table>
<thead>
<tr>
<th>Properties</th>
<th>Binding</th>
<th>Events</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>action</td>
<td>null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>background</td>
<td>[236,233,216]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>componentPopupMenu</td>
<td>&lt;none&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>font</td>
<td>Tahoma 11 Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>foreground</td>
<td>[0,0,0]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>icon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mnemonic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>text</strong></td>
<td>hello</td>
<td></td>
<td></td>
</tr>
<tr>
<td>toolTipText</td>
<td>null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Other Properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UIclassID</td>
<td>ButtonU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>actionCommand</td>
<td>jButton1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alignmentX</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alignmentY</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>autoscrolls</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>border</td>
<td>[XPEmptyBorder]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>borderPainted</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**text**

(instance of java.lang.String) The button's text.
### Variable Name

The name of the global variable generated for this component.
The Inspector window displays a tree hierarchy of components in the opened form.
The Inspector window displays a tree hierarchy of components in the opened form.
The Inspector window displays a tree hierarchy of components in the opened form.
/*
 * JFrame.java
 * Created on February 27, 2008, 8:37 PM
 */

package javaapplication17;

/**
 * @author yunglu
 */

public class NewJFrame extends javax.swing.JFrame {

/** Creates new form NewJFrame */
    public NewJFrame() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialise the form.
     * WARNING: Do NOT modify this code. The content of this method
     * always regenerated by the Form Editor */
}
```java
/*
 * always regenerated by the form editor.
 * /

// Generated Code

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

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new JFrame().setVisible(true);
        }
    });

    // Variables declaration - do not modify
    private javax.swing.JButton esc462Button;
    private javax.swing.JButton helloButton;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```
The Tools > Palette Manager > Swing/AWT Components menu item allows you to modify the...
```java
private void helloButtonActionPerformed(ActionEvent event) {
    // TODO add your handling code here:
    JLabel1.setText("Hello");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            NewJFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton ece462Button;
private javax.swing.JButton helloButton;
```
private void helloButtonActionPerformed(ActionEvent e) {
    // TODO add your handling code here:
    System.out.println("Hello" + e.getActionCommand());
}

private javax.swing.JButton helloButton;

private javax.swing.JButton ece462Button;
```java
private void helloButtonActionPerformed(ActionEvent e)
    // TODO add your handling code here:
    jLabel1.setText("Hello");
}

private void ece462ButtonActionPerformed(ActionEvent e)
    // TODO add your handling code here:
    jLabel1.setText("ECE462");

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new JFrame().setVisible(true);
        }
    });
}
```
CVS will replace $Log$ by the change history.
YHL Version Control 60
Commit Message:

add Log to Main.java
handle the events generated by the two buttons

Files to Commit:

<table>
<thead>
<tr>
<th>File</th>
<th>Status</th>
<th>Commit</th>
<th>Repository path</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewJFrame.java</td>
<td>Locally</td>
<td>Commit</td>
<td>.../src/javaapplication17</td>
</tr>
<tr>
<td>NewJFrame.form</td>
<td>Locally</td>
<td>Commit</td>
<td>.../src/javaapplication17</td>
</tr>
<tr>
<td>Main.java</td>
<td>Locally</td>
<td>Commit</td>
<td>.../src/javaapplication17</td>
</tr>
</tbody>
</table>
$Log$ replaced by the commit comments

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

/*
 * $Log: Main.java,v $
 * Revision 1.3  2006/02/28 01:50:23  ee462b30
 * add Log to Main.java
 * handle the events generated by the two buttons
 */

package

/**
 * ECE
 */

/**
 * @author yunghu
 */

public class Main {

/**
 * @param args the command line arguments
 */
create NewJFrame object
ECE462
package application17;

public class Main {

  public static void main(String[] args) {
    // TODO code application logic here
    JFrame frame = new JFrame();
    frame.add(new JButton());
    String message = args[0];
    System.out.println(message);

    // Example of handling events generated by the two buttons
    frame.add(new JButton() {

      public void actionPerformed(ActionEvent e) {
        // Handle event here
      }
    });
  }

  public static void main(String[] args) {
    // TODO code application logic here
    JFrame frame = new JFrame();
    frame.add(new JButton());
  }

  public static void main(String[] args) {
    // TODO code application logic here
    JFrame frame = new JFrame();
  }

  // ...
YHL Version Control 72
roll back to an earlier version
<table>
<thead>
<tr>
<th>Version</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:59:50 PM</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:56:41 PM</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:53:28 PM</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:50:23 PM</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:48:12 PM</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:49:37 PM</td>
<td></td>
</tr>
<tr>
<td>Feb 27, 2008 8:28:27 PM</td>
<td></td>
</tr>
</tbody>
</table>
new project
(msee190pc9) ~ [/] cd projects/
(msee190pc9) ~/projects/ ] ls
CVSROOT/ JavaApplication17/
(msee190pc9) ~/projects/ ] cd JavaApplication17/
(msee190pc9) ~/projects/JavaApplication17/ ] ls
build.xml,v  manifest.mf,v  nbproject/ src/ test/
(msee190pc9) ~/projects/JavaApplication17/ ] cd src/
(msee190pc9) ~/projects/JavaApplication17/src/ ] ls
javaapplication17/
(msee190pc9) ~/projects/JavaApplication17/src/ ] cd javaapplication17
/
(msee190pc9) ~/projects/JavaApplication17/src/javaapplication17/ ] ls
Main.java,v  NewJFrame.form,v  NewJFrame.java,v
(msee190pc9) ~/projects/JavaApplication17/src/javaapplication17/ ] more Main.java,v
information about the revision history

```java
re Main.java;
head 1.4;
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;
next 1.3;

1.3
date 2008.02.28.01.50.23; author ee462b30; state Exp;
branches;
next 1.2;

1.2
date 2008.02.28.01.33.17; author ee462b30; state Exp;
branches;
next 1.1;
```

--More--(23%)
1.3
log
@add Log to Main.java
handle the events generated by the two buttons
@
text
@d6 5
a10 1
* $Log$
@d7 2
@

1.2
log
@add a comment
@
text
@d5 3
a7 1
@

commit comments
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)
cd projects/

zip -r submission.zip JavaApplication18/*
adding: JavaApplication18/build.xml,v (deflated 65%)
adding: JavaApplication18/manifest.mf,v (deflated 44%)
adding: JavaApplication18/nbproject/ (stored 0%)
adding: JavaApplication18/nbproject/project.xml,v (deflated 54%)
adding: JavaApplication18/nbproject/project.properties,v (deflated 59%)
adding: JavaApplication18/nbproject/genfiles.properties,v (deflated 50%)
adding: JavaApplication18/nbproject/build-impl.xml,v (deflated 85%)
adding: JavaApplication18/nbproject/.cvsignore,v (deflated 47%)
adding: JavaApplication18/src/ (stored 0%)
adding: JavaApplication18/src/javaapplication18/ (stored 0%)
adding: JavaApplication18/src/javaapplication18/Main.java,v (deflated 54%)
adding: JavaApplication18/src/javaapplication18/NewJFrame.form,v (deflated 77%)
adding: JavaApplication18/src/javaapplication18/NewJFrame.java,v (deflated 69%)
adding: JavaApplication18/test/ (stored 0%)
ls
CVSROOT/ JavaApplication18/ submission.zip
ls
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:
  ```java
  ClassName objectName = new ClassName(...);
  JLabel jLabel1 = new javax.swing.JLabel();
  ```
- C++
  ```cpp
  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.
• No Shape object can be created.

Shape sObj = new Shape
package javaapplication19;

/**
   * @author yunglu
   */

class Main {

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

    public static void main(String[] args) {
        // TODO code application logic here
        Shape subj1 = new Triangle(5, 4);
        subj1.getArea();
        Triangle tobj2 = new Triangle(8, 6);
        tobj2.getArea();
        Rectangle robj3 = new Rectangle(7);
        robj3.getArea();
        // Shape subj3 = new Shape();
    }
}
abstract class and method

abstract class Shape {
    public Shape() {
        System.out.println("Shape::Shape");
        lineThickness = 1;
        shapeColor = Color.black;
    }
    abstract public double getArea();
    private int lineThickness;
    private Color shapeColor;
}
package javaapplication19;

/**
 * @author yunlu
 */

class Triangle extends Shape {

    public Triangle(double h, double w) {
        System.out.println("Triangle::Triangle");
        height = h;
        width = w;
    }

    public double getArea() {
        double area = 0.5 * height * width;
        System.out.println("Triangle::getArea " + area);
        return area;
    }

    private double height;
    private double width;
}
```java
package javaapplication19;

/**
 * @author yunqlu
 */

class Rectangle extends Shape {

    public Rectangle(double s) {
        System.out.println("Rectangle::Rectangle");
        side = s;
    }

    @Override
    public double getArea() {
        double area = side * side;
        System.out.println("Rectangle::getArea " + area);
        return area;
    }

    private double side;
}
```
```java
/**
 * @param args the command line arguments
 */

public static void main(String[] args) {
    // TODO code application logic here
    Shape obj1 = new Triangle(5, 4);
    obj1.getArea();
    Triangle obj2 = new Triangle(8, 6);
    obj2.getArea();
    Rectangle obj3 = new Rectangle(7);
    obj3.getArea();
    Shape obj3 = new Shape();
}
```

```
[les to C:\yunglu\java\JavaApplication19\build\classes
application\src\javaapplication19\Main.java:25: javaapplication19.Shape is abstract; cannot be instantiated
new Shape();]
```
```java
class Triangle extends Shape {

    Triangle() {
        javaapplication19.Triangle is not abstract and does not override abstract method getArea() in javaapplication19.Shape
    }

    System.out.println("Triangle::Triangle");
    height = h;
    width = w;
}

/**
 * public double getArea() {
 *     double area = 0.5 * height * width;
 *     System.out.println("Triangle::getArea " + area);
 *     return area;
 * }
 */

private double height;
private double width;
```
An abstract class can have methods that are not abstract.

if getArea is not abstract
YHL Instantiation 13

derived class does not have to override getArea

```java
package

public class Triangle extends Shape {

    public Triangle(double h, double w) {
        System.out.println("Triangle::Triangle");
        height = h;
        width = w;
    }

    public double getArea() {
        double area = 0.5 * height * width;
        System.out.println("Triangle::getArea " + area);
        return area;
    }

    private double height;
    private double width;
}
```
still cannot create a Shape object
package javaapplication19;

import java.awt.Color;

/**
 * @author yunjiu
 * javaapplication19.Shape is not abstract and does not override abstract method getArea() in javaapplication19.Shape
 *
 * class Shape {
 * 
 * public Shape() {
 *     System.out.println("Shape::Shape");
 *     lineThickness = 1;
 *     shapeColor = Color.black;
 * }
 *
 * abstract public double getArea();
 * 
 * System.out.println("Shape::getArea");
 * return 0;
 * }
 * 
 * private int lineThickness;
 * private Color shapeColor;
 * }
 * 
*
package javaapplication19;

import java.awt.Color;

/**
 * @author yunlu
 */
// abstract

class Shape {

    public Shape() {
        System.out.println("Shape::Shape");
        lineThickness = 1;
        shapeColor = Color.black;
    }

    abstract public double getArea();

    System.out.println("Shape::getArea");
    return 0;
}

private int lineThickness;
private Color shapeColor;
If a class is not abstract, it cannot have an abstract method.
package javaapplication19;

import java.awt.Color;

/**
 * @author yunglu
 */
abstract class Shape {

    public Shape() {
        System.out.println("Shape::Shape");
        lineThickness = 1;
        shapeColor = Color.black;
    }

    public double getArea() {
        System.out.println("Shape::getArea");
        return 0;
    }

    private int lineThickness;
    private Color shapeColor;
}
<table>
<thead>
<tr>
<th>class</th>
<th>method</th>
<th>create object</th>
<th>override method in derived class</th>
</tr>
</thead>
<tbody>
<tr>
<td>not abstract</td>
<td>not abstract</td>
<td>Y</td>
<td>not necessary</td>
</tr>
<tr>
<td>not abstract</td>
<td>abstract</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>abstract</td>
<td>abstract</td>
<td>No</td>
<td>Must</td>
</tr>
<tr>
<td>abstract</td>
<td>not abstract</td>
<td>No</td>
<td>not necessary</td>
</tr>
</tbody>
</table>
Examples of Abstract Classes
Overview | Package | Class | Use | Tree | Deprecated | Index | Help

PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | ALL CLASSES | DETAIL: FIELD | CONSTR | METHOD

`javax.swing`

**Class AbstractButton**

`java.lang.Object`  
`java.awt.Component`  
`java.awt.Container`  
`javax.swing.JComponent`  
`javax.swing.AbstractButton`

All Implemented Interfaces:
`
ImageObserver, ItemSelectable, MenuContainer, Serializable, SwingConstants`

Direct Known Subclasses:
`
JButton, JMenuItem, JToggleButton`

```java
public abstract class AbstractButton  
extends JComponent  
implements ItemSelectable, SwingConstants
```
Abstract C++ Class
```cpp
#include "Shape.h"
#include "Triangle.h"
#include "Rectangle.h"

int main(int argc, char * argv[]) {
    Shape * subj1 = new Triangle(4.0, 5.0);
    subj1 -> getArea();
    Rectangle obj2(7.0);
    obj2.getArea();
    Triangle * obj3 = new Triangle(6.0, 8.0);
    delete * obj3 = new Triangle(6.0, 8.0); // otherwise, memory leak
    subj1 = obj3;
    subj1 -> getArea();
    delete obj3;
    // Shape subj3;
    return 0;
}
```
A C++ class is abstract if it contains at least one (not necessarily all) "pure virtual" function:

```
virtual funcName (parameters) = 0;
```
#include "Shape.h"

Shape::Shape ()
{
}

Shape::~Shape ()
{
}
```cpp
#ifndef TRIANGLE_H
#define TRIANGLE_H

#include "Shape.h"
#include <iostream>

using namespace std;

class Triangle : public Shape
{
public:
    Triangle(double h, double w);
    virtual ~Triangle();
    virtual double getArea() {
        double area = 0.5 * height * width;
        cout << "Triangle::getArea() " << area << endl;
        return area;
    }

private:
    double height;
    double width;
};

#endif /* TRIANGLE_H */
```
```cpp
#include "Triangle.h"

Triangle::Triangle(double h, double w) :
    width(w) {
    height = h; // either way works
}

Triangle::~Triangle() {
}
```
```cpp
#ifndef RECTANGLE_H
#define RECTANGLE_H

#include "Shape.h"

class Rectangle : public Shape {
public:
    Rectangle(double s) :
        side(s) {} 
    virtual ~Rectangle();
    virtual double getArea();
private:
    double side;
};
#endif /*RECTANGLE_H */
```
```cpp
#include "Rectangle.h"
#include <iostream>
using namespace std;

Rectangle::Rectangle()
{
}

Rectangle::~Rectangle()
{
}

double Rectangle::getArea() {
    double area = side * side;
    cout << "Rectangle::getArea() " << area << endl;
    return area;
}
```
```cpp
#include "Shape.h"
#include "Triangle.h"
#include "Rectangle.h"

int main(int argc, char * argv[]) {
    Shape * subj1 = new Triangle(4.0, 5.0);
    subj1 -> getArea();
    Rectangle obj2(7.0);
    obj2.getArea();
    Triangle * obj3 = new Triangle(6.0, 8.0);
    delete subj1; // otherwise, memory leak
    subj1 = obj3;
    subj1 -> getArea();
    delete obj3;
    // Shape subj3;
    return 0;
}
```

The output of the program is:

```
<terminated> CppAbstract.exe [C/C++ Local Application] C:\yungu\eclipse\workspace\CppAbstract\Debug\CppAbstract.exe
Triangle::getArea() 10
Rectangle::getArea() 49
Triangle::getArea() 24
```
Abstract Class

**Java**

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

**C++**

- 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
Overview | Package | Class | Use | Tree | Deprecated | Index | Help
PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD |
FRAMES | NO FRAMES | ALL CLASSES | DETAIL: FIELD | CONSTR | METHOD

Java™ Platform
Standard Ed. 6

java.lang

Class Math

java.lang.Object

public final class Math
extends Object

The class Math contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.

Unlike some of the numeric methods of class StrictMath, all implementations of the equivalent functions of class Math are not defined to return the bit-for-bit same results. This relaxation permits better-performing implementations where strict reproducibility is not required.

By default many of the Math methods simply call the equivalent method in StrictMath for their implementation. Code generators are encouraged to use platform-specific native libraries or microprocessor instructions, where available, to provide

Done
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static double $e$</td>
<td>The double value that is closer than any other to $e$, the base of the natural logarithms.</td>
</tr>
<tr>
<td>static double $\pi$</td>
<td>The double value that is closer than any other to $\pi$, the ratio of the circumference of a circle to its diameter.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static double $\text{abs} (\text{double } a)$</td>
<td>Returns the absolute value of a double value.</td>
</tr>
<tr>
<td>static float $\text{abs} (\text{float } a)$</td>
<td>Returns the absolute value of a float value.</td>
</tr>
<tr>
<td>static int $\text{abs} (\text{int } a)$</td>
<td>Returns the absolute value of an int value.</td>
</tr>
<tr>
<td>static long $\text{abs} (\text{long } a)$</td>
<td>Returns the absolute value of a long value.</td>
</tr>
<tr>
<td>static double $\text{acos} (\text{double } a)$</td>
<td>Returns the arc cosine of a value; the returned angle is in the range $0.0$ through $\pi$.</td>
</tr>
<tr>
<td>static double $\text{asin} (\text{double } a)$</td>
<td>Returns the arc sine of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$.</td>
</tr>
</tbody>
</table>
QWidget Class Reference

[QtGui module]

The QWidget class is the base class of all user interface objects. More...

```cpp
#include <QWidget>
```

Inherits QObject and QPaintDevice.

Inherited by QComboBox, QDataBrowser, QDataView, QDateTimeEdit, QDateTimeEditBase, QDockArea, QHeader, QMainWindow, AbstractButton, AbstractSlider, AbstractSpinBox, AxWidget, CalendarWidget, QComboBox, DesignerActionEditorInterface, DesignerFormWindowInterface, DesignerObjectInspectorInterface, DesignerPropertyEditorInterface, DesignerQWidgetBoxInterface, DesktopWidget, Dialog, DialogButtonBox, DockWidget, FocusFrame, Frame, GLLight, GroupBox, LineEdit, QMainWindow, MDIContainer, Menu, MenuBar, ProgressBar, RubberBand, SizeGrip, SplashScreen, SplitterHandle, StatusBar, SvgWidget, TabBar, TabWidget, ToolBar, WizardPage, Workspace, WSObject, XEmbed, X11Embed, and XEmbed.

Done
Static Public Members

- `QWidget *find ( WId id )`
- `QWidget *keyGrabber ()`
- `QWidget *mouseGrabber ()`
- `void setTabOrder ( QWidget *first, QWidget *second )`

5 static public members inherited from `QObject`

Protected Functions

- `virtual void actionEvent ( QActionEvent *event )`
- `virtual void changeEvent ( QEvent *event )`
- `virtual void closeEvent ( QCloseEvent *event )`
- `virtual void contextMenuEvent ( QContextMenuEvent *event )`
- `void create ( WId window = 0, bool initializeWindow = true, bool destroyOldWindow = true )`
- `void destroy ( bool destroyWindow = true, bool destroySubWindows = true )`
- `virtual void dragEnterEvent ( QDragEnterEvent *event )`
- `virtual void dragLeaveEvent ( QDragLeaveEvent *event )`
- `virtual void dragMoveEvent ( QDragMoveEvent *event )`
- `virtual void dropEvent ( QDropEvent *event )`
- `virtual void enterEvent ( QEvent *event )`
- `virtual bool event ( QEvent *event )`
public class JavaStatic {

    /**
     * @param args
     */

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println(Math.PI);
        System.out.println(Math.E);
        double s1 = Math.sin(1);
        System.out.println(s1);
        System.out.println(Math.random());
    }
}
Object Counter in C++
```cpp
#ifndef CLASSWITHCOUNTER_H_
#define CLASSWITHCOUNTER_H_

class ClassWithCounter
{
public:
    ClassWithCounter();
    virtual ~ClassWithCounter();
    static int getCounter() { return objectCounter; }
private:
    static int objectCounter;
};

#endif /*CLASSWITHCOUNTER_H_*/
```
initialize

increment in constructor

decrement in destructor
```cpp
#include <iostream>
#include "ClassWithCounter.h"

using namespace std;

void createObject() {
    ClassWithCounter * obj2 = new ClassWithCounter;
    ClassWithCounter * obj3 = new ClassWithCounter;
    cout << "B " << obj2 -> getCounter() << endl;
    delete obj2;
    cout << "C " << obj3 -> getCounter() << endl;
}

int main(int argc, char * argv[]) {
    
    ClassWithCounter obj1;
    cout << "A " << obj1.getCounter() << endl;
}

// obj1 out of scope now, destroyed
createObject();
ClassWithCounter obj4;
cout << "D " << obj4.getCounter() << endl;
// the counter says 2 because obj3 has not been destroyed
cout << "E " << ClassWithCounter::getCounter() << endl;
return 0;
```
```cpp
#include <iostream>
#include "ClassWithCounter.h"
using namespace std;

void createObject() {
    ClassWithCounter * obj2 = new ClassWithCounter;
    ClassWithCounter * obj3 = new ClassWithCounter;
    cout << "B " << obj2 -> getCount() << endl;
    delete obj2;
    cout << "C " << obj3 -> getCount() << endl;
}

int main(int argc, char * argv[]) {
{
    ClassWithCounter obj1;
    cout << "A " << obj1.getCount() << endl;
}
// obj1 out of scope now, destroyed
createObject();
ClassWithCounter obj4;
cout << "D " << obj4.getCount() << endl;
// the counter says 2 because obj3 has not been counted
cout << "E " << ClassWithCounter::getCount() << endl;
return 0;
```
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
package objectcounter;

public class ClassWithCounter {
    public ClassWithCounter() {
        // System.out.println("ClassWithCounter");
        objectCounter++;
    }

    protected void finalize() {
        // called by Java garbage collector
        System.out.println("finalize");
        objectCounter--;
    }

    public static int getCount() {
        return objectCounter;
    }

    private static int objectCounter = 0;
}
```java
public class JavaMain {
    public static void createObject() {
        ClassWithCounter obj2 = new ClassWithCounter();
        ClassWithCounter obj3 = new ClassWithCounter();
        System.out.println("B " + obj2.getCounter());
        obj2 = null;
        System.out.println("C " + obj3.getCounter());
        System.gc();
        System.out.println("D " + obj3.getCounter());
    }

    public static void main(String[] args) {
        System.out.println(ClassWithCounter.getCounter());
        // TODO Auto-generated method stub
        ClassWithCounter obj1 = new ClassWithCounter();
        System.out.println("A " + obj1.getCounter());
    }

    // obj1 out of scope now, destroyed
    System.gc();
    createObject();
    ClassWithCounter obj4 = new ClassWithCounter();
    System.out.println("E " + obj4.getCounter());
}
```
Objects may not be destroyed immediately.
static function can access only static attributes
non-static function can also access static attributes

```java
public class MyClassWithCounter {
    private static int objectCounter = 0;
    private int stateValue;

    public classWithCounter() {
        // System.out.println("ClassWithCounter");
        objectCounter++;
    }

    protected void finalize() {
        // called by Java garbage collector
        // System.out.println("finalize");
        objectCounter--;
    }

    public static int getCounter() {
        return objectCounter;
    }

    public static int getState() {
        return stateValue;
    }
}
```
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.
<table>
<thead>
<tr>
<th>Person A</th>
<th>Person B</th>
<th>GROUP</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit</td>
<td>Submit Same</td>
<td>Submitted</td>
<td>Same</td>
</tr>
<tr>
<td>Submit</td>
<td>Submit Same</td>
<td>Not Submitted</td>
<td>Cheating ⇒ F</td>
</tr>
<tr>
<td>Submit</td>
<td>Submit Different</td>
<td>Not Submitted</td>
<td>work alone</td>
</tr>
<tr>
<td>Submit</td>
<td>Submit Different</td>
<td>Submitted</td>
<td>20% penalty of the average</td>
</tr>
<tr>
<td>Submit</td>
<td>Not Submitted</td>
<td>Submitted by A</td>
<td>50% penalty both</td>
</tr>
<tr>
<td>Submit</td>
<td>Not Submitted</td>
<td>Not Submitted</td>
<td>B receives 0</td>
</tr>
</tbody>
</table>
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).
  ⇒ The case will be reported to the associate head.
  ⇒ 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
CVS repository

```
[(msee190pc9) ~/projects/ ] ls
Assignment1/ CVSROOT/
[(msee190pc9) ~/projects/ ]
```
CVS repository contain .v files

```bash
[msee190pc9] ~/projects/ ] ls
Assignment1/ CVSROOT/
[msee190pc9] ~/projects/Assignment1/ ] ls
ClassWithCounter.cpp,v ClassWithCounter.h,v main.cpp,v
[msee190pc9] ~/projects/Assignment1/ ]
```
YHL Submission Procedure

[[msee190pc9] ~/projects/ ] ls
Assignment1/  CVSROOT/
[[msee190pc9] ~/projects/ ] cd Assignment1/
[[msee190pc9] ~/projects/Assignment1/ ] ls
ClassWithCounter.cpp,v  ClassWithCounter.h,v  main.cpp,v
[[msee190pc9] ~/projects/Assignment1/ ] cd ..
[[msee190pc9] ~/projects/ ] zip -r PA1.zip Assignment1/
  adding: Assignment1/ (stored 0%)
  adding: Assignment1/ClassWithCounter.h,v (deflated 49%)
  adding: Assignment1/ClassWithCounter.cpp,v (deflated 43%)
  adding: Assignment1/.project,v (deflated 76%)
  adding: Assignment1/.cproject,v (deflated 95%)
  adding: Assignment1/main.cpp,v (deflated 54%)
[[msee190pc9] ~/projects/ ]
Submit in Blackboard
Edit Submission: Lab01 (Attempt 1)

Due Date: August 29, 2008 11:59 PM
Type: Work individually
Grading Criteria: Alphanumeric

Instructions:

Attachments:

Submission:

Enable HTML Creator
Upload Files from Your Computer

The Java applet for selecting and uploading multiple files was closed. To reopen the applet, click My Computer.

If you do not want to use the Java applet, you can individually select and upload files. Click here to select files individually.
Are you sure you want to submit this assignment?
Confirmation

- Demo Student:
  You have submitted your assignment (Demo Student).
  Submitted Time: March 8, 2008 6:54 AM

OK
Take Back and Resubmit
Take back for resubmission
remove the submitted file
Resubmit

Add Comment:

Add Attachments

Use HTML
Upload Files from Your Computer

The Java applet for selecting and uploading multiple files was closed. To reopen the applet, click My Computer.

If you do not want to use the Java applet, you can individually select and upload files. Click here to select files individually.
YHL Submission Procedure

Add Comment:

Submit  Cancel  Save as Draft
Are you sure you want to submit this assignment?

[OK]  [Cancel]
Verify Submission

This step is extremely important. You must check whether you submit correct files.
You have chosen to open

PA1.zip

which is a: UltimateZip File

from: https://blackboard.purdue.edu

What should Firefox do with this file?

- Open with "UltimateZip Executable (default)"
- Save to Disk
- Do this automatically for files like this from now on.

OK | Cancel
YHL Submission Procedure

```
[(msee190pc9) ~/submission/ ] ls
PA1.zip
```

[(msee190pc9) ~/submission/ ] ls
PA1.zip
[(msee190pc9) ~/submission/ ] mkdir CVSROOT
[(msee190pc9) ~/submission/ ] ls
CVSROOT/ PA1.zip
[(msee190pc9) ~/submission/ ] pwd
/home/shay/a/ee462b30/submission
[(msee190pc9) ~/submission/ ]
YHL Submission Procedure

```bash
[msee190pc9] ~ ] cd submission/
[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/submission
[msee190pc9] ~/submission/ ]
```
Are these files expected?
YHL Submission Procedure

```
[msee190pc9] ~/verify/ ] cd Assignment1/
[msee190pc9] ~/verify/Assignment1/ ] ls
ClassWithCounter.cpp  ClassWithCounter.h  CVS/ main.cpp
[msee190pc9] ~/verify/Assignment1/ ] cd
[msee190pc9] ~/ ] cd verify/Assignment1/
[msee190pc9] ~/verify/Assignment1/ ] c
[msee190pc9] ~/verify/Assignment1/ ] ls
ClassWithCounter.cpp  ClassWithCounter.h  CVS/ main.cpp
[msee190pc9] ~/verify/Assignment1/ ] qmake -project
[msee190pc9] ~/verify/Assignment1/ ] qmake
[msee190pc9] ~/verify/Assignment1/ ] 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/QtGui -I.../../sfwtools/public/qt4.3.0/include/QtG
tGui -I.../../sfwtools/public/qt4.3.0/include -I. -I. -I. -o ClassWithCounter.o
ClassWithCounter.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/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
```
Is the output as expected?

A 1
B 2
C 1
D 2
E 2
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 $\Rightarrow$ Circle, Square, Triangle
  – BankAccount $\Rightarrow$ CheckingAccount, SavingAccount
  – MotorVehicle $\Rightarrow$ 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
### java.awt

**Class Component**

```java
java.lang.Object
  java.awt.Component
```

**All Implemented Interfaces:**

- `ImageObserver`, `MenuContainer`, `Serializable`

**Direct Known Subclasses:**

- `Button`, `Canvas`, `Checkbox`, `Choice`, `Container`, `Label`, `List`, `Scrollbar`, `TextComponent`

```java
public abstract class Component extends Object
```

## Inheritance

<table>
<thead>
<tr>
<th>Prev Class</th>
<th>Next Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>SUMMARY:</td>
<td>DETAIL:</td>
</tr>
</tbody>
</table>

---

YHL Inheritance 5
**getTreeLock**

`public final Object getTreeLock()`

Gets this component's locking object (the object that owns the thread synchronization monitor) for AWT component-tree and layout operations.

**Returns:**

this component's locking object

**getToolkit**

`public Toolkit getToolkit()`

Gets the toolkit of this component. Note that the frame that contains a component controls which toolkit is used by that component. Therefore if the component is moved from one frame to another, the toolkit it uses may change.
**dispatchEvent**

```java
public final void dispatchEvent(AWTEvent e)
```

Dispatches an event to this component or one of its sub components. Calls `processEvent` before returning for 1.1-style events which have been enabled for the `Component`.

**Parameters:**

- `e` - the event

**postEvent**

```java
@Deprecated
public boolean postEvent(Event e)
```

**Deprecated.** As of JDK version 1.1, replaced by `dispatchEvent(AWTEvent)`.

**Specified by:**
package fnaexample;

public class Shape {
    public Shape() {
        sColor = 0;
    }

    final public void setColor(int c) {
        sColor = c;
    }

    protected int sColor; // RGB
```java
package finaexample;

public final class Triangle extends Shape {

    public void setColor(int c) {
        sColor = c;
    }
}
```
package finaexample;

public class ETriangle extends Triangle {

    The type ETriangle cannot subclass the final class Triangle.

    Press ‘F2’ for focus.
}


package finaexample;

public class Rectangle extends Shape {

    public void setColor(int r, int g, int b) {
        sColor = (r << 16) | (g << 8) | b;
    }
}
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)
```cpp
#ifndef TRIANGLE_H_
#define TRIANGLE_H_

class Triangle
{
private:
    Triangle();
    virtual ~Triangle();
};

#endif /*TRIANGLE_H_*/
```
#ifndef DTRIANGLE_H_
define DTRIANGLE_H_

#include "Triangle.h"

class DTriangle : public Triangle
{
public:
    DTriangle();
    virtual ~DTriangle();
};

#endif /*DTRIANGLE_H__*/
#ifndef TRIANGLE_H
#define TRIANGLE_H

class Triangle
{
    private:
    
    Triangle();
    virtual ~Triangle();
};

#endif /*TRIANGLE_H__*/
"Final" in Java

<table>
<thead>
<tr>
<th>purpose</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>final double PI = 3.1415926535;</td>
</tr>
<tr>
<td>block inheritance</td>
<td>final class Triangle { ...</td>
</tr>
<tr>
<td>block overriding</td>
<td>public final void setColor ( ...</td>
</tr>
<tr>
<td>constant parameter</td>
<td>public void setColor (final int c)</td>
</tr>
</tbody>
</table>

- `finalized()` ⇒ called when a garbage object is collected
package finaexample;

public class Shape {
    public Shape() {
        sColor = 0;
    }

    final public void setColor(int c) {
        sColor = c;
    }

    public void setColor(final int r, final int g, final int b) {
        sColorr = (r << 16) | (g << 8) | b;
        r = g | b;
    }

    protected int sColor; // RGB
package finaexample;

public class Shape {
    public Shape() {
        sColor = 0;
    }

    final public void setColor(int c) {
        sColor = c;
    }

    public void setColor(final int r, final int g, final int b) {
        sColor = (r << 16) | (g << 8) | b;
        r = g | b;
    }

    protected int sColor; // RGB
Constant Parameter in C++
```cpp
#ifndef TRIANGLE_H_
#define TRIANGLE_H_

class Triangle {
public:

    void setColor(const int k)
    {
        k = 5;
    }

private:

    Triangle();

toTriangle();

#elif END/

/*TRIANGLE_H__*/
```
```cpp
#ifndef TRIANGLE_H
#define TRIANGLE_H

class Triangle {
public:
  void setColor(const int k) {
    n = 5;
  }

private:
  Triangle();
  virtual ~Triangle();
};

#endif /*TRIANGLE_H */
```
Constant Function in C++
If "const" is added after the parentheses, the method cannot modify any attribute.
```cpp
#ifndef TRIANGLE_H
#define TRIANGLE_H

class Triangle {
public:
    void setColor(const int r) {
        // r = 5;
    }

    int getColor() const {
        tColor = 0;
        return tColor;
    }

private:
    Triangle();
    virtual ~Triangle();
    int tColor;

#endif /*TRIANGLE_H*/
```
Self Test