

# **ECE 462**

# **Object-Oriented Programming**

# **using C++ and Java**

## **Draw Game**

Yung-Hsiang Lu  
yunglu@purdue.edu

# document generated by javadoc

**Package** [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)  
PREV PACKAGE NEXT PACKAGE [FRAMES](#) [NO FRAMES](#)

## Package

### com.brackeen.javagamebook.tilegame.sprites

#### Class Summary

<a href="#">Creature</a>	A Creature is a Sprite that is affected by gravity and can die.
<a href="#">Fly</a>	A Fly is a Creature that fly slowly in the air.
<a href="#">Grub</a>	A Grub is a Creature that moves slowly on the ground.
<a href="#">Player</a>	The Player.
<a href="#">PowerUp</a>	A PowerUp class is a Sprite that the player can pick up.
<a href="#">PowerUp.Goal</a>	A Goal PowerUp.
<a href="#">PowerUp.Music</a>	A Music PowerUp.
<a href="#">PowerUp.Star</a>	A Star PowerUp.

PowerUp - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

**All Classes**

- [Creature](#)
- [Fly](#)
- [Grub](#)
- [Player](#)
- [PowerUp](#)
- [PowerUp.Goal](#)
- [PowerUp.Music](#)
- [PowerUp.Star](#)

**Package** **Class** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)

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

---

**com.brackeen.javagamebook.tilegame.sprites**

## Class PowerUp

java.lang.Object  
└ Sprite  
    └ com.brackeen.javagamebook.tilegame.sprites.PowerUp

**Direct Known Subclasses:**

[PowerUp.Goal](#), [PowerUp.Music](#), [PowerUp.Star](#)

---

```
public abstract class PowerUp
extends Sprite
```

A PowerUp class is a Sprite that the player can pick up.

---

### Nested Class Summary

Firefox browser window titled "Creature - Mozilla Firefox". The address bar is empty. The menu bar includes File, Edit, View, History, Bookmarks, Yahoo!, Tools, and Help. The left sidebar shows "All Classes" with a list of links: Creature (selected), Fly, Grub, Player, PowerUp, PowerUp.Goal, PowerUp.Music, and PowerUp.Star.

The main content area displays the JavaDoc page for the `com.brackeen.javagamebook.tilegame.sprites.Creature` class. The page has a navigation bar with links for Package, Class (selected), Tree, Deprecated, Index, and Help. Below this are links for PREVIOUS CLASS, NEXT CLASS, FRAMES, NO FRAMES, and SUMMARY: NESTED | FIELD | CONSTR | METHOD. There are also links for DETAIL: FIELD | CONSTR | METHOD.

The class name `com.brackeen.javagamebook.tilegame.sprites.Creature` is shown. Below it is the class hierarchy:

```
java.lang.Object
  ↳ Sprite
    ↳ com.brackeen.javagamebook.tilegame.sprites.Creature
```

**Direct Known Subclasses:**

[Fly](#), [Grub](#), [Player](#)

---

```
public abstract class Creature
extends Sprite
```

A Creature is a Sprite that is affected by gravity and can die. It has four Animations: moving left, moving right, dying on the left, and dying on the right.

---

**Field Summary**

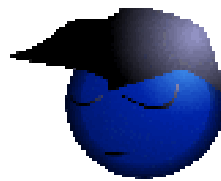
```
emacs@HELPSTABLET2
File Edit Options Buffers Tools Java Help
public abstract class Creature extends Sprite {

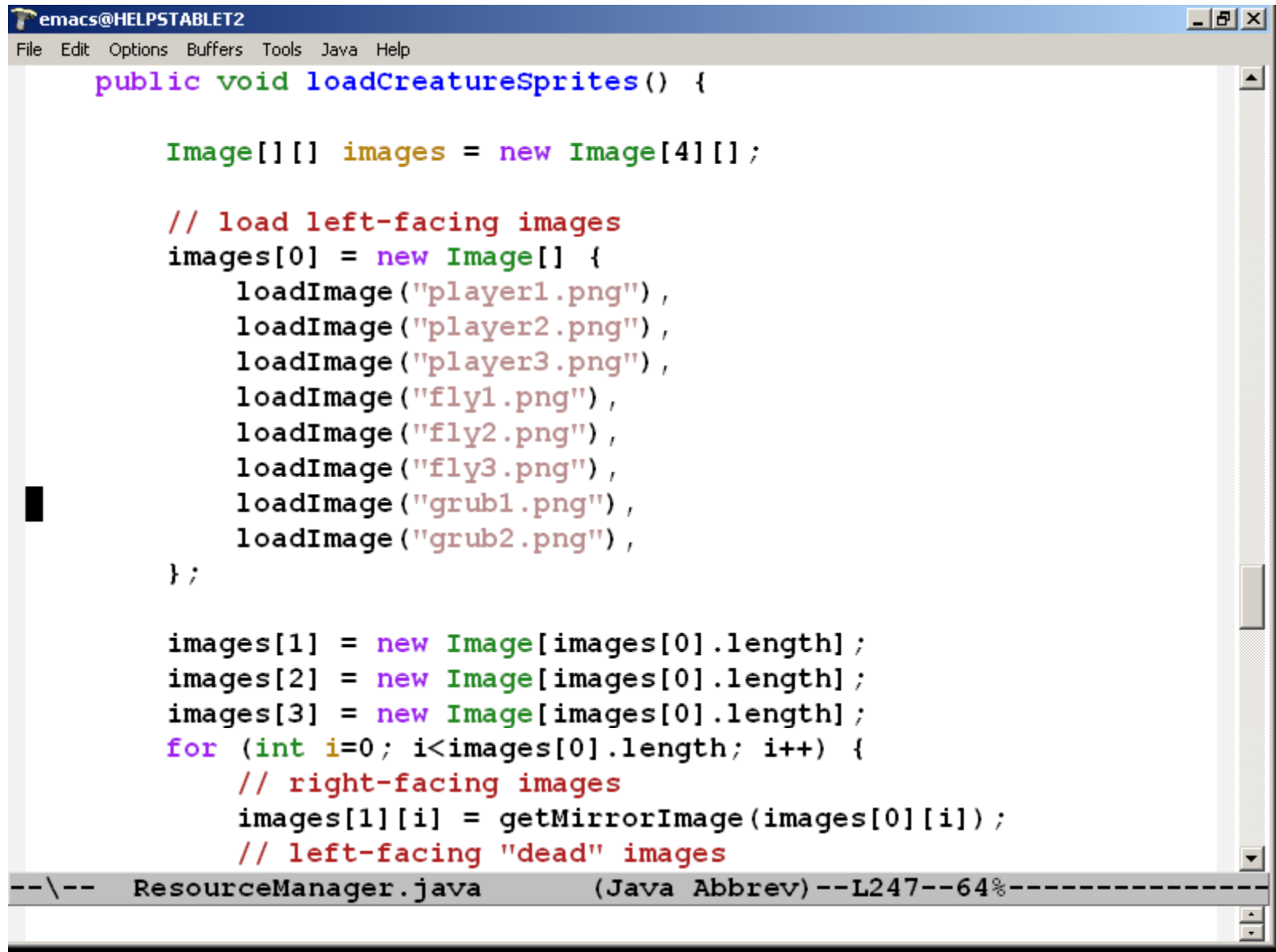
    /**
     * Amount of time to go from STATE_DYING to STATE_DEAD.
     */
    private static final int DIE_TIME = 1000;

    public static final int STATE_NORMAL = 0;
    public static final int STATE_DYING = 1;
    public static final int STATE_DEAD = 2;

    private Animation left;
    private Animation right;
    private Animation deadLeft;
    private Animation deadRight;
    private int state;
    private long stateTime;

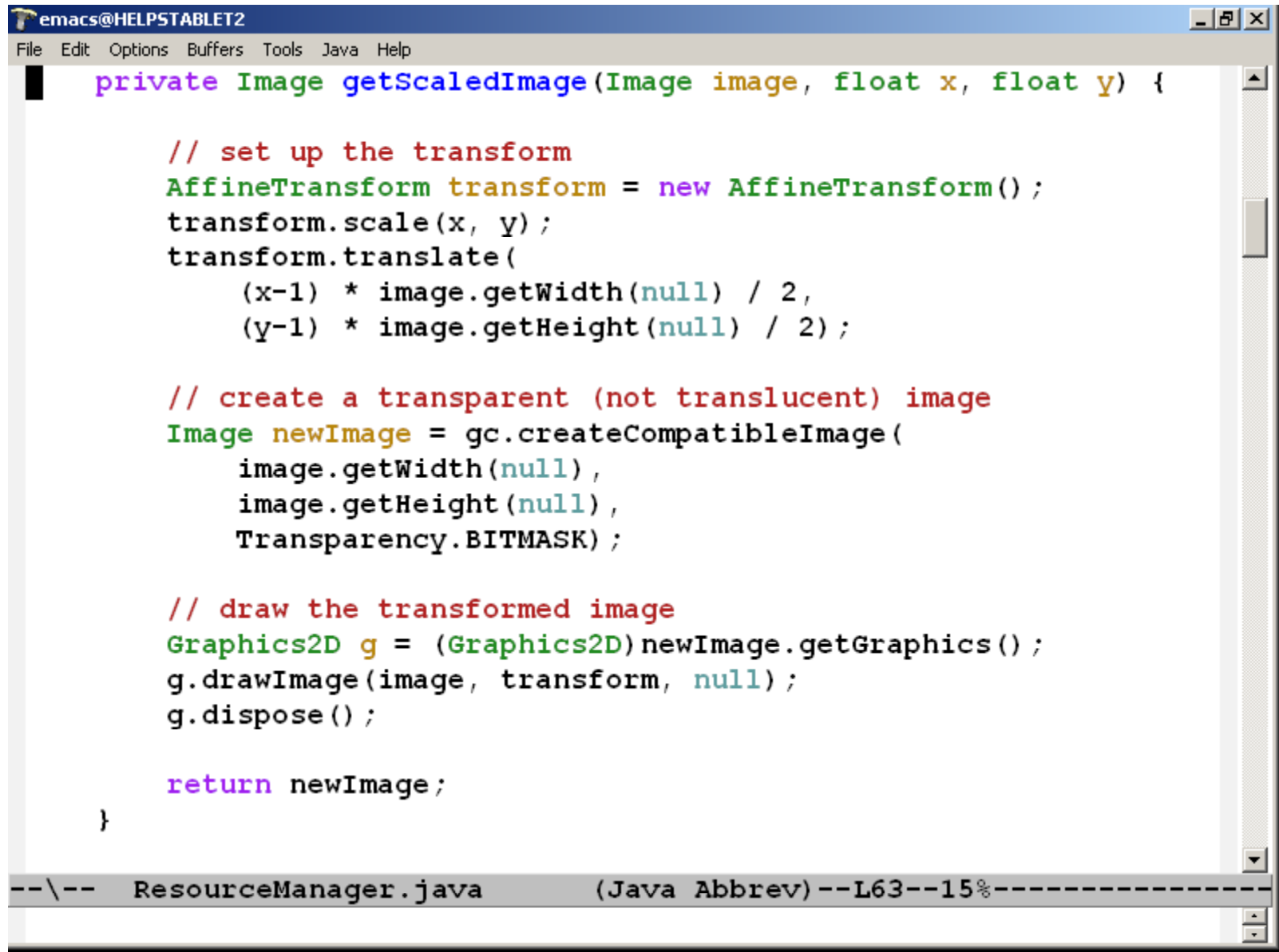
    /**
     * Creates a new Creature with the specified Animations.
     */
    public Creature(Animation left, Animation right,
                   Animation deadLeft, Animation deadRight)
}
--\-- Creature.java (Java Abbrev)--L13-- 8%-----
```



The image shows a screenshot of an Emacs editor window. The title bar reads "emacs@HELPSTABLET2". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Java", and "Help". The main text area contains the following Java code:

```
public void loadCreatureSprites() {  
  
    Image[][] images = new Image[4][];  
  
    // load left-facing images  
    images[0] = new Image[] {  
        loadImage("player1.png"),  
        loadImage("player2.png"),  
        loadImage("player3.png"),  
        loadImage("fly1.png"),  
        loadImage("fly2.png"),  
        loadImage("fly3.png"),  
        loadImage("grub1.png"),  
        loadImage("grub2.png"),  
    };  
  
    images[1] = new Image[images[0].length];  
    images[2] = new Image[images[0].length];  
    images[3] = new Image[images[0].length];  
    for (int i=0; i<images[0].length; i++) {  
        // right-facing images  
        images[1][i] = getMirrorImage(images[0][i]);  
        // left-facing "dead" images  
    }  
}
```

The status bar at the bottom of the window shows "--\-- ResourceManager.java (Java Abbrev) --L247--64%-----".

The image shows a screenshot of an Emacs editor window. The title bar reads "emacs@HELPSTABLET2". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Java", and "Help". The main text area contains the following Java code:

```
private Image getScaledImage(Image image, float x, float y) {  
  
    // set up the transform  
    AffineTransform transform = new AffineTransform();  
    transform.scale(x, y);  
    transform.translate(  
        (x-1) * image.getWidth(null) / 2,  
        (y-1) * image.getHeight(null) / 2);  
  
    // create a transparent (not translucent) image  
    Image newImage = gc.createCompatibleImage(  
        image.getWidth(null),  
        image.getHeight(null),  
        Transparency.BITMASK);  
  
    // draw the transformed image  
    Graphics2D g = (Graphics2D) newImage.getGraphics();  
    g.drawImage(image, transform, null);  
    g.dispose();  
  
    return newImage;  
}
```

The status bar at the bottom of the window shows "--\-- ResourceManager.java (Java Abbrev) --L63--15%--".



A screenshot of a Mozilla Firefox browser window displaying the Java Platform Standard Ed. 6 documentation for the `AffineTransform` class. The browser title is "AffineTransform (Java Platform SE 6) - Mozilla Firefox". The address bar shows "File Edit View History Bookmarks Yahoo! Tools Help". The page content includes navigation links for "Overview", "Package", "Class", "Use Tree", "Deprecated", "Index", and "Help". The "Class" link is highlighted. Below these links are "PREV CLASS" and "NEXT CLASS" links, and "SUMMARY: NESTED | FIELD | CONSTR | METHOD" and "DETAIL: FIELD | CONSTR | METHOD" links. The page title is "Class AffineTransform" under the package "java.awt.geom". It lists the superclass "java.lang.Object" and the implemented interfaces "Serializable" and "Cloneable". The class signature is shown as "public class AffineTransform extends Object implements Cloneable, Serializable". A description follows: "The AffineTransform class represents a 2D affine transform that performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the 'straightness' and 'parallelness' of lines. Affine transformations can be constructed using sequences of translations, scales, flips, rotations, and shears." The browser status bar at the bottom shows "YHL" and "Draw Game".

AffineTransform (Java Platform SE 6) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

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.awt.geom

## Class AffineTransform

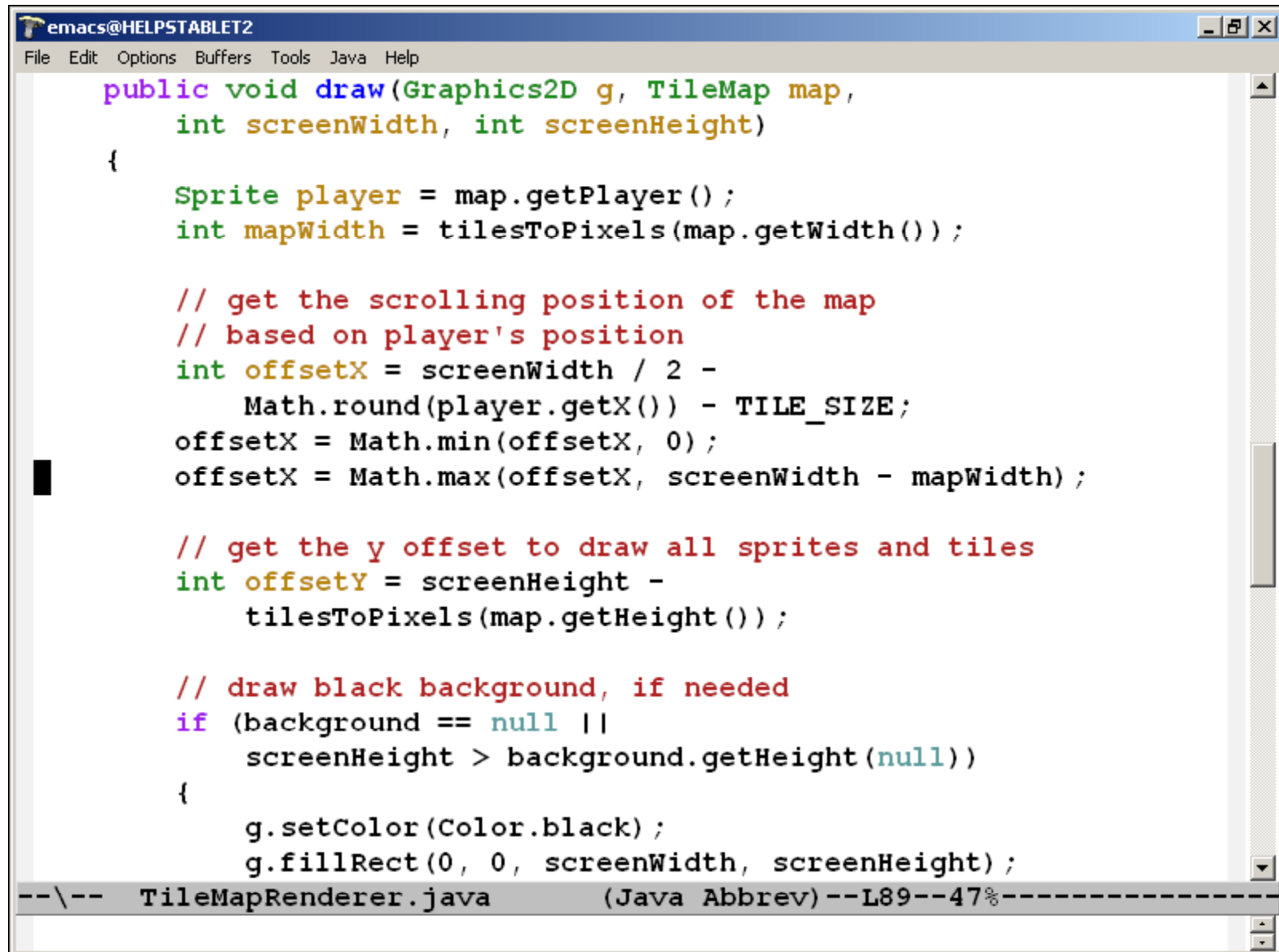
[java.lang.Object](#)  
└ java.awt.geom.AffineTransform

**All Implemented Interfaces:**  
[Serializable](#), [Cloneable](#)

```
public class AffineTransform
extends Object
implements Cloneable, Serializable
```

The `AffineTransform` class represents a 2D affine transform that performs a linear mapping from 2D coordinates to other 2D coordinates that preserves the "straightness" and "parallelness" of lines. Affine transformations can be constructed using sequences of translations, scales, flips, rotations, and shears.

YHL Draw Game

The image shows a screenshot of an Emacs editor window. The title bar reads "emacs@HELPSTABLET2". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Java", and "Help". The main text area contains Java code for a method named "draw". The code is color-coded: keywords are purple, class names and variables are green, integers are orange, and comments are red. The code defines a method that takes a Graphics2D object and a TileMap object, and returns void. It calculates the map width in pixels, determines the x and y offsets for scrolling based on the player's position, and draws a black background if needed. The status bar at the bottom of the window shows "--\-- TileMapRenderer.java (Java Abbrev)--L89--47%-----".

```
public void draw(Graphics2D g, TileMap map,
    int screenWidth, int screenHeight)
{
    Sprite player = map.getPlayer();
    int mapWidth = tilesToPixels(map.getWidth());

    // get the scrolling position of the map
    // based on player's position
    int offsetX = screenWidth / 2 -
        Math.round(player.getX()) - TILE_SIZE;
    offsetX = Math.min(offsetX, 0);
    offsetX = Math.max(offsetX, screenWidth - mapWidth);

    // get the y offset to draw all sprites and tiles
    int offsetY = screenHeight -
        tilesToPixels(map.getHeight());

    // draw black background, if needed
    if (background == null ||
        screenHeight > background.getHeight(null))
    {
        g.setColor(Color.black);
        g.fillRect(0, 0, screenWidth, screenHeight);
    }
}
```

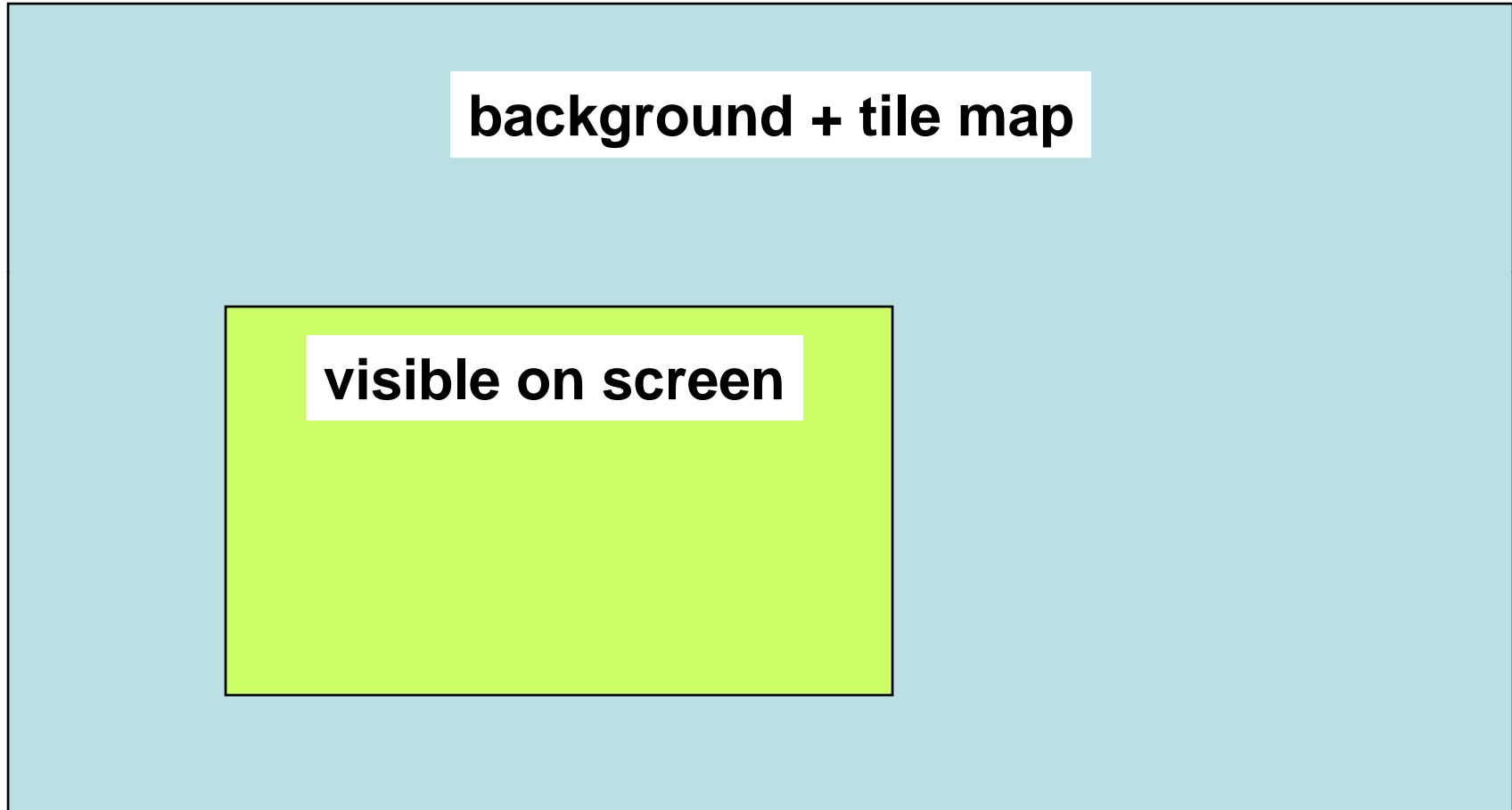
```
emacs@HELPSTABLET2
File Edit Options Buffers Tools Java Help

// draw the visible tiles
int firstTileX = pixelsToTiles(-offsetX);
int lastTileX = firstTileX +
    pixelsToTiles(screenWidth) + 1;
for (int y=0; y<map.getHeight(); y++) {
    for (int x=firstTileX; x <= lastTileX; x++) {
        Image image = map.getTile(x, y);
        if (image != null) {
            g.drawImage(image,
                tilesToPixels(x) + offsetX,
                tilesToPixels(y) + offsetY,
                null);
        }
    }
}

// draw player
g.drawImage(player.getImage(),
    Math.round(player.getX()) + offsetX,
    Math.round(player.getY()) + offsetY,
    null);

// draw sprites
--\-- TileMapRenderer.java (Java Abbrev) --L122--72%-----
```

# Large Background



# Manage Visible Objects



- An object is visible if the location is between  $\text{offsetX}$  and  $\text{offsetX} + \text{xresolution}$
- In a game with few moving objects, it is acceptable to draw every object. If it is outside the visible region, it cannot be seen.
- If the background moves slowly, it gives the impression of distance. This allows a smaller image for the background and saves memory.

# **ECE 462**

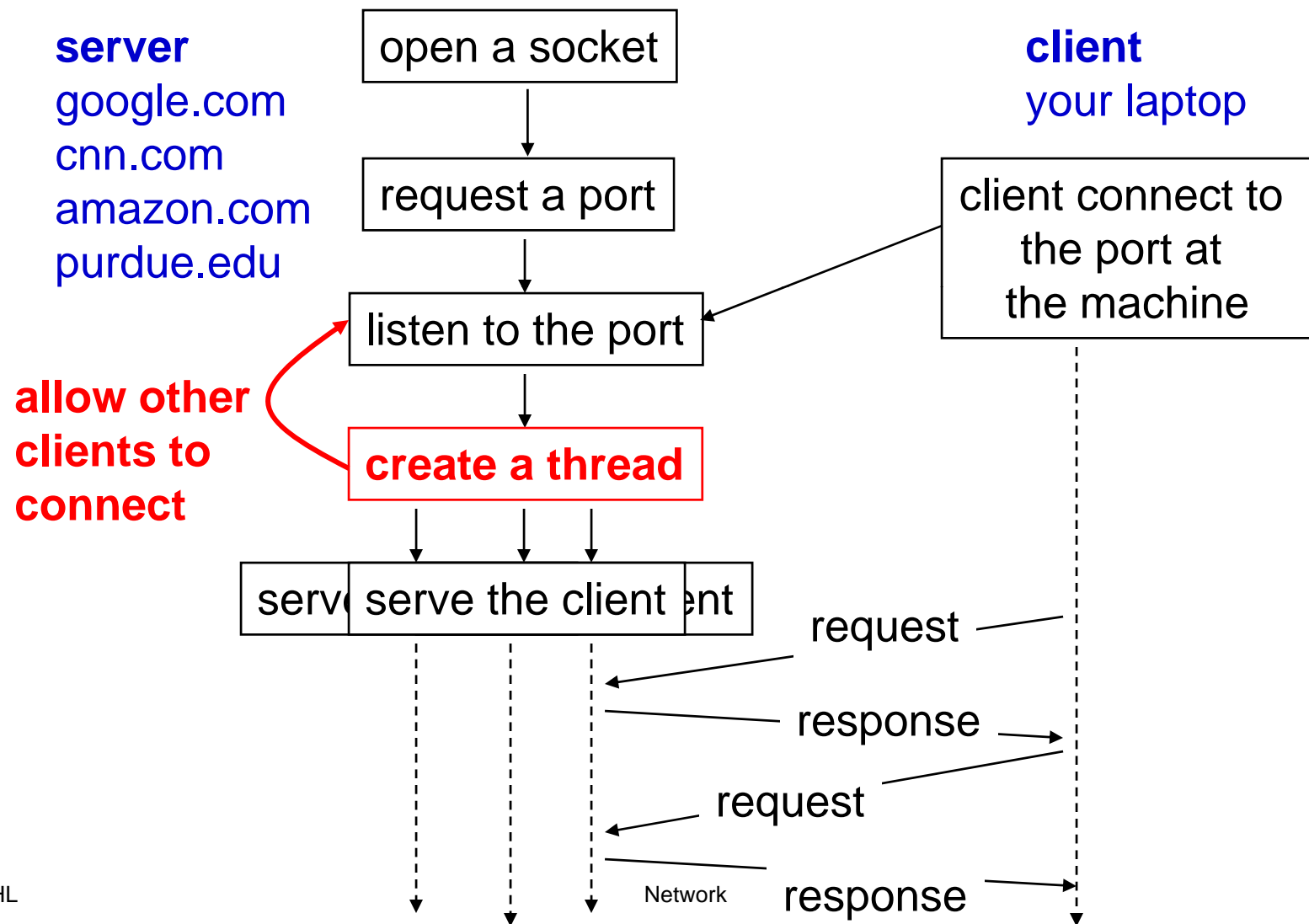
# **Object-Oriented Programming**

# **using C++ and Java**

## **Network Clients**

Yung-Hsiang Lu  
yunglu@purdue.edu

# Networking, Client-Server Model



```
qstruct04.ecn.purdue.edu - ee462b30@qstruct04 - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
[(qstruct04) ~/ ] ping www.google.com
PING www.l.google.com (64.233.167.99) 56(84) bytes of data.
64 bytes from py-in-f99.google.com (64.233.167.99): icmp
p_seq=0 ttl=243 time=12.7 ms
64 bytes from py-in-f99.google.com (64.233.167.99): icmp
p_seq=1 ttl=243 time=12.6 ms
64 bytes from py-in-f99.google.com (64.233.167.99): icmp
p_seq=2 ttl=243 time=12.9 ms
64 bytes from py-in-f99.google.com (64.233.167.99): icmp
p_seq=3 ttl=243 time=12.9 ms

--- www.l.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
 3005ms
rtt min/avg/max/mdev = 12.686/12.832/12.977/0.162 ms, p
ipe 2
[(qstruct04) ~/ ] █
```



```
qstruct04.ecn.purdue.edu - ee462b30@qstruct04 - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
[(qstruct04) ~/ ] traceroute www.mit.edu
traceroute to www.mit.edu (18.7.22.83), 30 hops max, 46
byte packets
 1 msee-b003-c6506-01.ecn-101.ecn.purdue.edu (128.46.1
01.1)  0.370 ms  0.235 ms  0.227 ms
 2 172.19.124.1 (172.19.124.1)  0.635 ms  0.587 ms  0.
281 ms
 3 192.31.0.5 (192.31.0.5)  0.369 ms  0.311 ms  0.317
ms
 4 192.31.0.101 (192.31.0.101)  0.472 ms  5.833 ms  0.
343 ms
 5 192.31.0.13 (192.31.0.13)  0.418 ms  0.373 ms  0.47
5 ms
 6 lynn-b168-m10i-01-campus.tcom.purdue.edu (192.5.40.
58)  3.727 ms  0.471 ms  0.523 ms
 7 gigapop-ctc-re-t640.tcom.purdue.edu (192.5.40.134)
1.989 ms  1.603 ms  1.553 ms
 8 149.165.254.226 (149.165.254.226)  7.608 ms  7.506
ms  6.927 ms
 9 newy-chic-100.layer3.nlr.net (216.24.186.33)  30.29
2 ms  29.759 ms  29.598 ms
```

List of TCP and UDP port numbers - Wikipedia, the free encyclopedia - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

W http://en.wikipedia.org/wiki/List\_of\_TCP\_and\_UDP\_port\_numbers

Learn more about using Wikipedia for research. [Log in / create account](#)

[article](#) [discussion](#) [edit this page](#) [history](#)

## List of TCP and UDP port numbers

From Wikipedia, the free encyclopedia

**Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are transport layer protocols used for communication between computers. The Internet Assigned Numbers Authority (IANA) is responsible for assigning port numbers for specific uses.**

**Contents** [hide]

- 1 Ranges
- 2 Port status colors
- 3 Well Known Ports: 0–1023
- 4 Registered Ports: 1024–49151
- 5 Dynamic and/or Private Ports: 49152–65535
- 6 References
- 7 External links

### Ranges [edit]

The port numbers are divided into three ranges<sup>[1]</sup>:

- The *Well Known Ports* are those in the range 0–1023. On [Unix-like operating systems](#), opening a port in this range to receive incoming connections requires administrative privileges, or possession

Done

List of TCP and UDP port numbers - Wikipedia, the free encyclopedia - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

W http://en.wikipedia.org/wiki/List\_of\_TCP\_and\_UDP\_port\_numbers

53/TCP,UDP	<a href="#">Domain Name System (DNS)</a>	Official
54/TCP,UDP	XNS ( <a href="#">Xerox Network Services</a> ) Clearinghouse	Official
56/TCP,UDP	XNS ( <a href="#">Xerox Network Services</a> ) Authentication	Official
56/TCP,UDP	RAP ( <a href="#">Route Access Protocol</a> ) <sup>[4]</sup>	Unofficial
57/TCP	MTP, <a href="#">Mail Transfer Protocol</a>	Unofficial
58/TCP,UDP	XNS ( <a href="#">Xerox Network Services</a> ) Mail	Official
67/UDP	<a href="#">Bootstrap Protocol (BOOTP)</a> Server; also used by <a href="#">Dynamic Host Configuration Protocol (DHCP)</a>	Official
68/UDP	<a href="#">Bootstrap Protocol (BOOTP)</a> Client; also used by <a href="#">Dynamic Host Configuration Protocol (DHCP)</a>	Official
69/UDP	<a href="#">Trivial File Transfer Protocol (TFTP)</a>	Official
70/TCP	<a href="#">Gopher protocol</a>	Official
79/TCP	<a href="#">Finger protocol</a>	Official
80/TCP	<a href="#">Hypertext Transfer Protocol (HTTP)</a>	Official
81/TCP	<a href="#">Torpark—Onion routing</a>	Unofficial
82/UDP	<a href="#">Torpark—Control</a>	Unofficial
83/TCP	MIT ML Device	Official
88/TCP	<a href="#">Kerberos—authentication system</a>	Official

Done



# Client States

- Does a later request from the same client depend on an earlier request?  
No  $\Rightarrow$  stateless  
Yes  $\Rightarrow$  stateful
- For stateful requests, the client, the server, or both have to store information based on earlier requests.

HTTP, HyperText Transfer Protocol - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://www.networksorcery.com/enp/protocol/http.htm

## HTTP, HyperText Transfer Protocol

[Description](#)   [Glossary](#)   [RFCs](#)   [Publications](#)   [Obsolete RFCs](#)

**Description:**

*Protocol suite:* [TCP/IP](#).

*Type:* Application layer file transfer protocol.

*Ports:* HTTP: 80, 8008, 8080 (TCP) server.  
S-HTTP: 80 (TCP) server.  
HTTPS: 443 (TCP) server over SSL/TLS.

*Related protocols:* [webDAV](#), Web Distributed Authoring and Versioning.

*URI:* [http:](#), [https:](#)

*MIME subtype:* [application/http](#), [message/http](#), [message/s-http](#).

*Working groups:* [http](#), HyperText Transfer Protocol.  
[httpbis](#), Hypertext Transfer Protocol Bis.  
[webdav](#), WWW Distributed Authoring and Versioning.  
[wts](#), Web Transaction Security.

*Links:* [HTTP Object Header lines](#).  
[HTTP status codes](#).

Done

HTTP, HyperText Transfer Protocol - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://www.networksorcery.com/enp/protocol/http.htm

http command

### HTTP message:

---

### Methods:

Method	References
DELETE	<a href="#">RFC 1945</a>
GET	<a href="#">RFC 1945</a>
HEAD	<a href="#">RFC 1945</a>
LINK	<a href="#">RFC 1945</a>
OPTIONS	<a href="#">RFC 2068</a>
PATCH	<a href="#">RFC 2068</a>
POST	<a href="#">RFC 1945</a>
PUT	<a href="#">RFC 1945</a>
TRACE	<a href="#">RFC 2068</a>
UNLINK	<a href="#">RFC 1945</a>

---

### Header fields:

Header field	Description	References
Done		

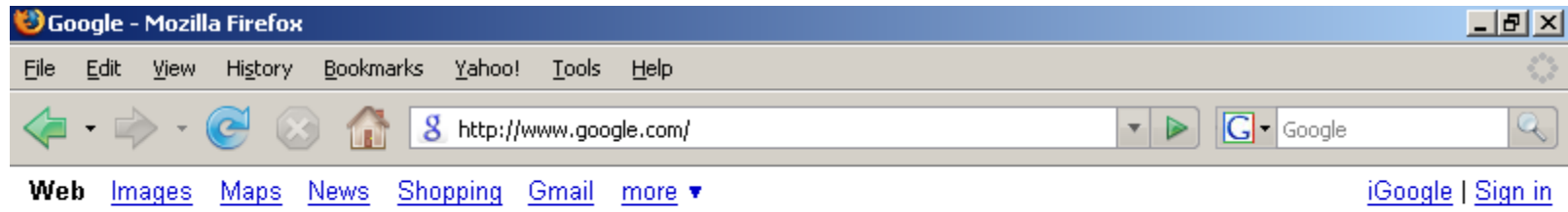
# Java HTTP Client

```
Java - network/ClientSocket.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help

ClientSocket.java
//ClientSocket.java
import java.io.*;
import java.net.*;
class ClientSocket {
    public static void main(String[] args) {
        try {
            String webAddress = args[0];
            String hostHeader = "Host: " + webAddress;
            Socket socket = new Socket(webAddress, 80);
            OutputStream os = socket.getOutputStream();
            PrintStream ps = new PrintStream(os, true);
            InputStream in = socket.getInputStream();
            InputStreamReader in_reader = new InputStreamReader(in);
            BufferedReader b_reader = new BufferedReader(in_reader);
            ps.print("GET / HTTP/1.0\r\n" + hostHeader + "\r\n" + "\r\n");
            boolean more = true;
            while (more) {
                String str = b_reader.readLine();
                if (str == null)
                    more = false;
                System.out.println(str);
            }
        } catch (IOException e) {
            System.out.println("Error: " + e);
        }
    }
}

Writable | Smart Insert | 22 : 1
```





Google™

Google Search | I'm Feeling Lucky

[Advanced Search](#)  
[Preferences](#)  
[Language Tools](#)

[Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2008 Google

Done

YHL

Network

12

```
qstruct04.ecn.purdue.edu - ee462b30@qstruct04 - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
[(qstruct04) ~/lecturecode/1105/java/ ] java ClientSocket
et www.google.com
HTTP/1.0 200 OK
Cache-Control: private, max-age=0
Date: Wed, 02 Jul 2008 14:48:20 GMT
Expires: -1
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=49176524815f28e0:TM=1215010100:LM=1
215010100:s=_kSS5YgXK0Jm8l5y; expires=Fri, 02-Jul-2010
14:48:20 GMT; path=/; domain=.google.com
Server: gws
Connection: close

<html><head><meta http-equiv="content-type" content="te
xt/html; charset=ISO-8859-1"><title>Google</title><styl
e>body,td,a,p,.h{font-family:arial,sans-serif}.h{font-s
ize:20px}.h{color:#3366cc}.q{color:#00c}.ts td{padding:
0}.ts{border-collapse:collapse}.lnc:link,.lnc:visited{c
olor:#00c}.pgtab,.pgtab:hover,.pgtabselected,.pgtabside
{text-align:center;text-decoration:none;color:#00c;disp
lay:block;height:27px;float:left;overflow:hidden;backgr
```



# Network Protocols

- Protocols specify what commands / formats can be sent and the responses / formats.
- Protocols also specify the responses when commands cannot execute.

# C++ Qt HTTP Client

Qt 4.4: Qt Examples - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://doc.trolltech.com/4.4/examples.html

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

# Qt Examples

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

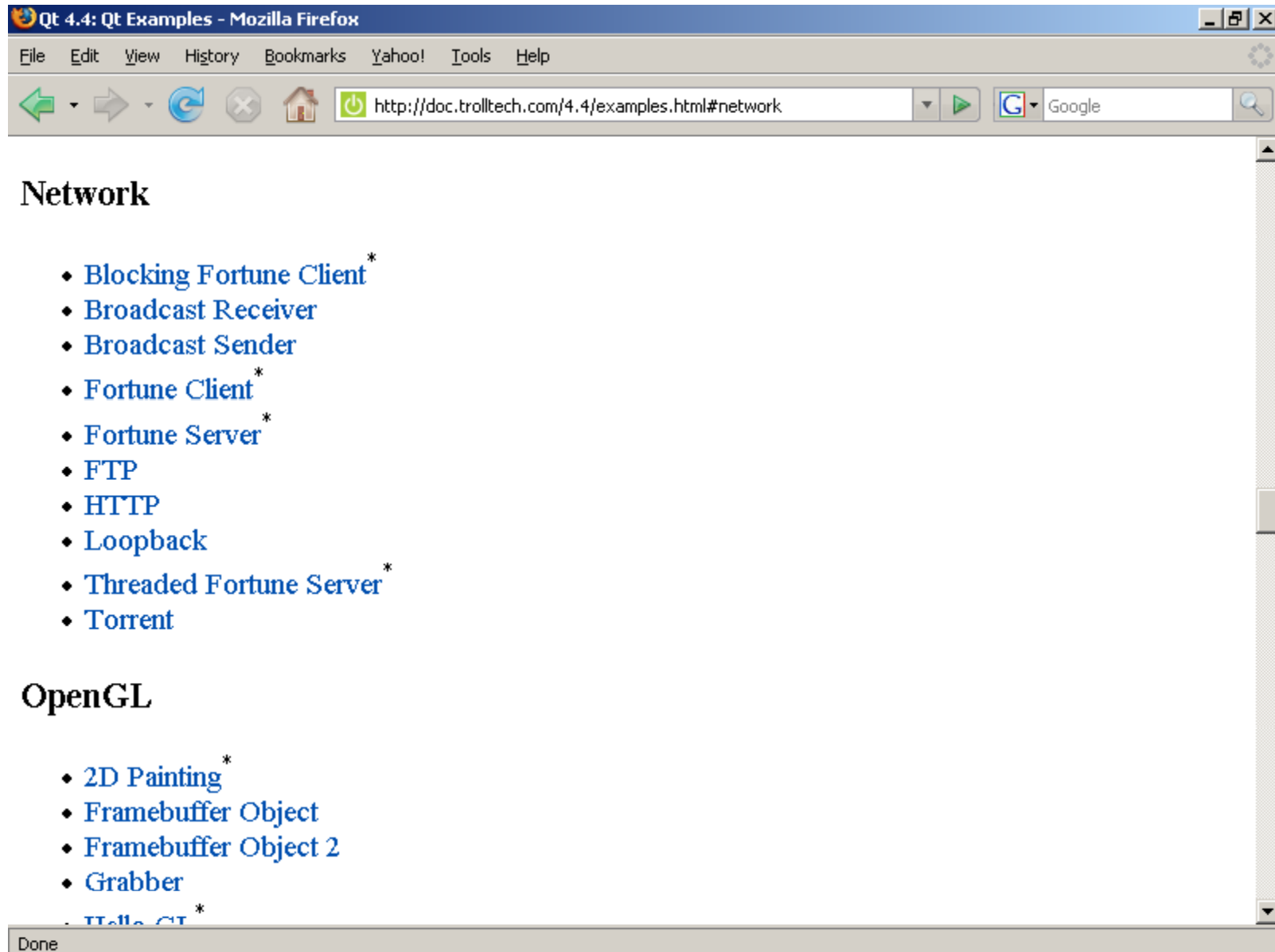
If you are new to Qt, you should probably start by going through the [Tutorials](#) before you have a look at the [Application](#) example.

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

One more valuable source for examples and explanations of Qt features is the archive of the [Qt Quarterly](#).

In the list below, examples marked with an asterisk (\*) are fully documented. Eventually, all the examples will be fully documented, but sometimes we include an example before we have time to write

Done



```
emacs@HELPSTABLET2
File Edit Options Buffers Tools Debug IDLWAVE Help
#####
#####
###
# Automatically generated by qmake (2.01a) Wed Jul 2 11:47:09 2008
#####
###

TEMPLATE = app
TARGET =
DEPENDPATH += .
INCLUDEPATH += .
QT += network
# Input
HEADERS += ClientSocket.h
SOURCES += ClientSocket.cc

-- (Unix) -- httpclient.pro (IDLWAVE Abbrev Fill) --L4--All-----
```

after qmake -project  
before qmake



```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C Help
#include <QtGui>
#include <QtNetwork>
#include <QTcpServer>
#include <string>
using namespace std;

class ClientSocket : public QObject {
    Q_OBJECT
    string wwwName;
    QTcpSocket* socket;
    string constructHttpRequest();
public:
    ClientSocket( string name );
    ~ClientSocket();
public slots:
    void reportConnected();
    void reportHostFound();
    void getWebPage();
    void socketConnectionClosed();
};

#endif

-- (Unix)-- ClientSocket.h (C Abbrev)--L5--Bot-----
```



```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
ClientSocket::ClientSocket( string siteName )
{
    wwwName = siteName;
    socket = new QTcpSocket( );
    connect( socket, SIGNAL( connected() ),
            this, SLOT( reportConnected() ) );
    connect( socket, SIGNAL( hostFound() ),
            this, SLOT( reportHostFound() ) );
    connect( socket, SIGNAL( readyRead() ),
            this, SLOT( getWebPage() ) );
    connect( socket, SIGNAL( connectionClosed() ),
            this, SLOT( socketConnectionClosed() ) );
    QString qstr( wwwName.c_str() );
    socket->connectToHost( qstr, 80 );
}

ClientSocket::~~ClientSocket() {
    if (socket) {
        delete socket;
    }
}

string ClientSocket::constructHttpRequest( ) {
--(Unix)-- ClientSocket.cc (C++ Abbrev)--L11-- 6%-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
string ClientSocket::constructHttpRequest( ) {
    string hostHeader = "Host: " + wwwName;
    string urlString( hostHeader );
    string httpRequestString = "GET / HTTP/1.0\r\n" +
        urlString + "\r\n" + "\r\n";
    return httpRequestString;
}

void ClientSocket::reportHostFound() {
    // cout << "host found" << endl;
}

void ClientSocket::reportConnected() {
    // cout << "connection established" << endl;
    string httpRequest = constructHttpRequest();
    socket->write( httpRequest.c_str() );
}

void ClientSocket::getWebPage() {
    // cout << "socket ready to read" << endl;
    int howManyBytes = socket->bytesAvailable();
    // cout << "bytes available: " << howManyBytes << endl;
    char data[howManyBytes];
--(Unix)-- ClientSocket.cc (C++ Abbrev)--L33--40%-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
void ClientSocket::getWebPage() {
    // cout << "socket ready to read" << endl;
    int howManyBytes = socket->bytesAvailable();
    // cout << "bytes available: " << howManyBytes << endl;
    char data[howManyBytes];
    socket->read( data, howManyBytes );
    cout << data;
    cout.flush();
}

void ClientSocket::socketConnectionClosed() {
    // cout << "socketConnectionClosed" << endl;
    exit(0);
}

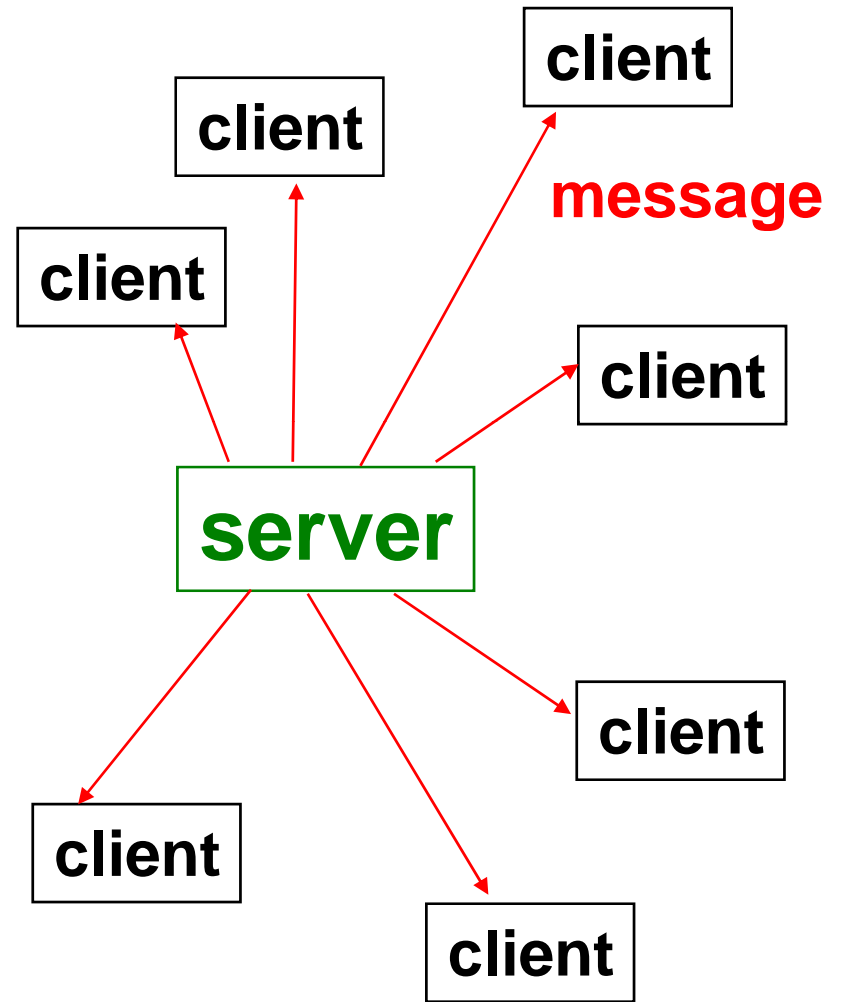
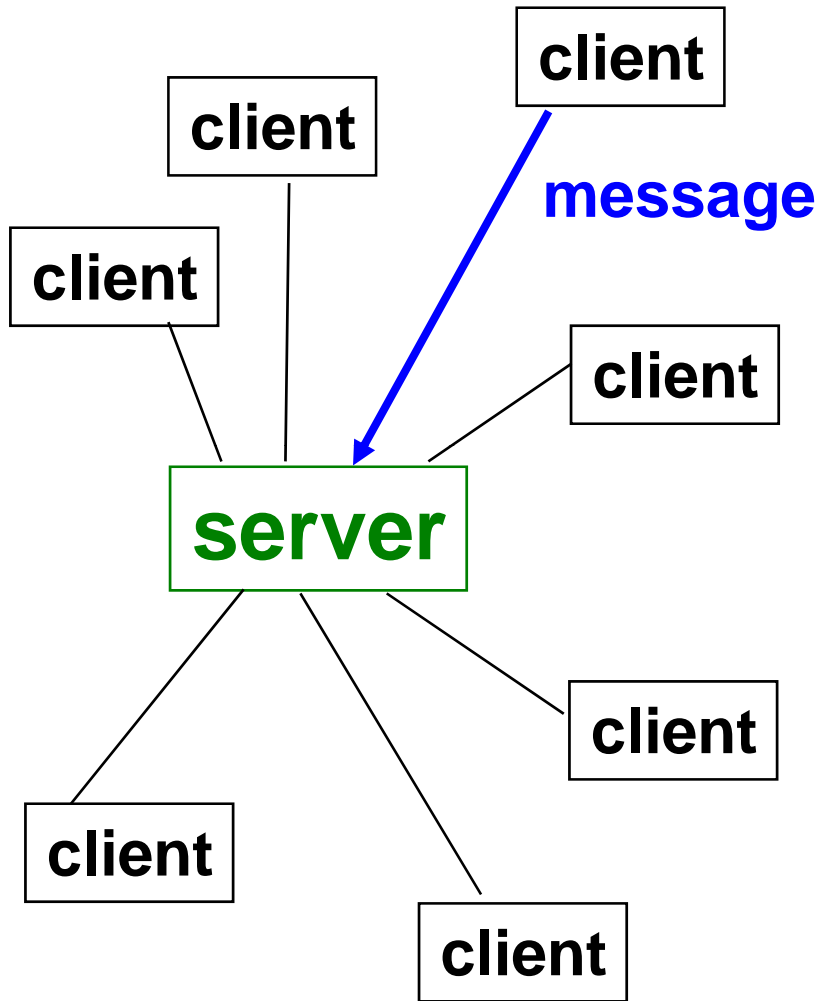
int main( int argc, char* argv[] )
{
    if (argc > 1)
    {
        QApplication app( argc, argv );
        ClientSocket sock( argv[1] );
        return app.exec();
    }
}
--(Unix)-- ClientSocket.cc (C++ Abbrev)--L59--68%-----
```

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

**Network Server**

Yung-Hsiang Lu  
yunglu@purdue.edu

# C++ Qt Chat Server



```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C Help
#include <QtNetwork>
#include <QString>
#include <QThread>
#include <QApplication>
#include <vector>
using namespace std;

class ChatServer;
class ClientHandler : public QObject
{
    Q_OBJECT
private:
    QString ch_name;
    QTcpSocket* ch_socket;
    QTextStream* ch_os;
public:
    ClientHandler( QTcpSocket* sock);
    virtual ~ClientHandler();
    static ChatServer * ch_server;
private slots:
    void readFromClient();
    friend class ChatServer;
};
--(Unix)-- chatserver.h (C Abbrev)--L10-- 8%
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C Help
void readFromClient();
friend class ChatServer;
};

class ChatServer: public QObject
{
    Q_OBJECT
private:
    QTcpServer * cs_server;
    QList<ClientHandler *> cs_clientList;
public:
    ChatServer( );
    virtual ~ChatServer();
    void broadcastClient(QString);
public slots:
    void connectNewClient( );
};
#endif

--(Unix)-- chatserver.h (C Abbrev)--L26--Bot-----
```



```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
//ChatServer.cc
#include "ChatServer.h"
#include <iostream>
using namespace std;

ChatServer * ClientHandler::ch_server = 0;
ChatServer::ChatServer( )
{
    cs_server = new QTcpServer();
    if (! cs_server->listen())
    {
        qWarning( "Failed to register the server port" );
        exit( 1 );
    }
    cout << "Server port " << cs_server->serverPort() << endl;
    connect(cs_server, SIGNAL(newConnection()),
           this, SLOT(connectNewClient()));
    ClientHandler::ch_server = this;
}

void ChatServer::connectNewClient()
{
    QTcpSocket* socket = cs_server->nextPendingConnection();
}
--(Unix)-- ChatServer.cc (C++ Abbrev)--L1--Top-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
void ChatServer::connectNewClient()
{
    QTcpSocket* socket = cs_server->nextPendingConnection();
    ClientHandler* clh = new ClientHandler( socket );
    cs_clientList.push_back(clh);
    cout << "A new client connected " << endl;
}

void ChatServer::broadcastClient(QString message)
{
    ClientHandler * handler;
    for (int index = 0; index < cs_clientList.size(); index ++)
    {
        handler = cs_clientList[index];
        *(handler-> ch_os) << message;
        (handler-> ch_os) -> flush();
    }

    // A chatter's terminal always shows name at beginning
    // of a new line.
    for (int index = 0; index < cs_clientList.size(); index ++)
    {
        ClientHandler * handler = cs_clientList[index];
--(Unix)-- ChatServer.cc (C++ Abbrev)--L24--18%-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
for (int index = 0; index < cs_clientList.size(); index ++ )
{
    ClientHandler * handler = cs_clientList[index];
    *(handler-> ch_os) << (handler->ch_name + ": ");
    (handler-> ch_os) -> flush();
}

ChatServer::~ChatServer()
{
    if (cs_server)
        { delete cs_server; }
    ClientHandler* clh = cs_clientList.takeFirst();
    while (clh != 0)
    {
        delete clh;
        clh = cs_clientList.takeFirst();
    }
}

ClientHandler::ClientHandler( QTcpSocket* socket)
: ch_name(""),
  ch_socket( socket )
--(Unix)-- ChatServer.cc (C++ Abbrev)--L53--39%-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
ClientHandler::ClientHandler( QTcpSocket* socket)
: ch_name(""),
  ch_socket( socket )
{
  ch_os = new QTextStream( ch_socket );

  (*ch_os) << "Welcome to a chat room powered by C++\n";
  (*ch_os) << ">>>>  Enter 'bye' to exit  <<<\n";
  (*ch_os) << "Enter chat name: ";
  ch_os -> flush();
  connect( ch_socket, SIGNAL( readyRead() ),
          this, SLOT( readFromClient() ) );
}
ClientHandler::~ClientHandler()
{
  if (ch_os)
    { delete ch_os; }
}

void ClientHandler::readFromClient() {
  QTcpSocket* sock = (QTcpSocket*) sender();
  while ( sock->canReadLine() ) {
--(Unix)-- ChatServer.cc (C++ Abbrev)--L75--54%-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
void ClientHandler::readFromClient() {
    QTcpSocket* sock = (QTcpSocket*) sender();
    while ( sock->canReadLine() ) {
        QString qstr = sock->readLine();
        qstr = qstr.trimmed(); // remove white space
        if ( ch_name == "" )
        {
            ch_name = qstr;
            QString outgoing = "\nMessage from chat server: " +
                ch_name + " signed in\n";
            ch_server -> broadcastClient(outgoing);
        }
        else if ( qstr == "bye" )
        {
            QString outgoing = "\nMessage from the chat server: " +
                ch_name + " signed off\n";
            ch_server -> broadcastClient(outgoing);
            ch_socket->close();
            ch_socket = 0;
        }
        else
        {
            QString outgoing = "\n" + ch_name + ": " + qstr + "\n";

```

```
--(Unix)-- ChatServer.cc (C++ Abbrev)--L95--71%-----
```

```
emacs@HELPSTABLET2
File Edit Options Buffers Tools C++ Help
}
else if ( qstr == "bye" )
{
    QString outgoing = "\nMessage from the chat server: " +
        ch_name + " signed off\n";
    ch_server -> broadcastClient(outgoing);
    ch_socket->close();
    ch_socket = 0;
}
else
{
    QString outgoing = "\n" + ch_name + ": " + qstr + "\n";
    ch_server -> broadcastClient(outgoing);
}
}
}
int main( int argc, char* argv[] )
{
    QApplication app( argc, argv );
    ChatServer server;
    return app.exec();
}
--(Unix)-- ChatServer.cc (C++ Abbrev)--L109--Bot-----
```

**QObject \* QObject::sender () const** [protected]

Returns a pointer to the object that sent the signal, if called in a slot activated by a signal; otherwise it returns 0. The pointer is valid only during the execution of the slot that calls this function.

The pointer returned by this function becomes invalid if the sender is destroyed, or if the slot is disconnected from the sender's signal.

**Warning:** This function violates the object-oriented principle of modularity. However, getting access to the sender might be useful when many signals are connected to a single slot.

See also [QSignalMapper](#).

**void QObject::setParent ( QObject \* parent )**

# **C++ Qt Server with Long Latency**



```
File Edit Options Buffers Tools C++ Help
bool IsPrime(int num)
{
    int factor = 2;
    // a very slow way to compute
    do
    {
        if ((num % factor) == 0)
            { return false; }
        factor ++;
    } while (factor < num);
    return true;
}
long TotalPrime(long maxValue)
{
    long sum = 0;
    if (maxValue <= 1) { return 0; } // no prime number
    if (maxValue == 2) { return 1; } // 2 is prime
    if (maxValue >= 3) { sum = 2; } // 2 and 3 are prime
    for (int curNum = 4; curNum <= maxValue; curNum ++)
    {
        if (IsPrime(curNum) == true)
            { sum ++; }
    }
    return sum;
}
--(Unix)-- PrimeNumber.cc (C++ Abbrev)--L12--All-----
```

```
File Edit Options Buffers Tools C++ Help
void ClientHandler::readFromClient() {
    QTcpSocket* sock = (QTcpSocket*) sender();
    while ( sock->canReadLine() ) {
        QString qstr = sock->readLine();
        qstr = qstr.trimmed();
        if ( qstr == "bye" )
        {
            (*ch_os) << "Bye\n\n";
            ch_os -> flush();
            ch_socket->close();
            ch_socket = 0;
        }
        else
        {
            long maxValue = qstr.toLong();
            long total = TotalPrime(maxValue);
            QString outgoing = "\n There are " +
                QString::number(total) +
                " prime numbers between 0 and " +
                QString::number(maxValue)
                + "\n";
            (*ch_os) << outgoing;
            (*ch_os) << "Give a number or say \"bye\": ";
            ch_os -> flush();
        }
    }
}
-- (Unix) -- PrimeServer.cc (C++ Abbrev) -- L78 -- 64%
```

# Server with Threads

```
File Edit Options Buffers Tools C Help
#ifndef PRIMESERVER_H
#define PRIMESERVER_H

#include <QtNetwork>
#include <QString>
#include <QThread>
#include <QApplication>
#include <vector>
using namespace std;

class PrimeThread: public QThread
{
public:
    PrimeThread(long v, QTextStream* s);
    void run();
    long total;
    QTextStream* os;
    long maxValue;
};

class PrimeServer;
class ClientHandler : public QObject
{
    Q_OBJECT
private:

```

-- (Unix) -- PrimeServer.h (C Abbrev) -- L17 -- Top -----

```
emils@HEPSTABLERZ
File Edit Options Buffers Tools C Help
class PrimeServer;
class ClientHandler : public QObject
{
    Q_OBJECT
private:
    QTcpSocket* ch_socket;
    QTextStream* ch_os;
public:
    ClientHandler( QTcpSocket* sock);
    virtual ~ClientHandler();
public slots:
    void answerReady();
private slots:
    void readFromClient();
    friend class PrimeServer;
};

class PrimeServer: public QObject
{
    Q_OBJECT
private:
    QTcpServer * cs_server;
    QList<ClientHandler *> cs_clientList;
public:
```

-- (Unix) -- PrimeServer.h (C Abbrev) --L21--38%-----

```
emilcs@HEEPSTABLERTZ
File Edit Options Buffers Tools C++ Help
PrimeThread::PrimeThread(long v, QTextStream* s)
{
    maxValue = v;
    os = s;
}

void PrimeThread::run()
{
    total = TotalPrime(maxValue);
}

PrimeServer::PrimeServer( )
{
    cs_server = new QTcpServer();
    if (! cs_server->listen())
    {
        qWarning( "Failed to register the server port" );
        exit( 1 );
    }

    cout << "Server port " << cs_server->serverPort() << endl;
    connect(cs_server, SIGNAL(newConnection()),
           this, SLOT(connectNewClient()));
}

void PrimeServer::connectNewClient()
```

-- (Unix) -- PrimeServer.cc (C++ Abbrev) -- L17 -- 5% -----

```
emils@HEEPSTABLERZ
File Edit Options Buffers Tools C++ Help
void ClientHandler::readFromClient() {
    QTcpSocket* sock = (QTcpSocket*) sender();
    while ( sock->canReadLine() ) {
        QString qstr = sock->readLine();
        qstr = qstr.trimmed();
        if ( qstr == "bye" )
        {
            (*ch_os) << "Bye\n\n";
            ch_os -> flush();
            ch_socket->close();
            ch_socket = 0;
        }
        else
        {
            long maxValue = qstr.toLong();
            PrimeThread * pt = new PrimeThread(maxValue, ch_os);
            pt -> start();
            connect(pt, SIGNAL( finished() ),
                    this, SLOT( answerReady() ) );
        }
    }
}

void ClientHandler::answerReady()
{
-- (Unix)-- PrimeServer.cc (C++ Abbrev)--L71--60%-----
```

```
emals@HEPSTABLERZ
File Edit Options Buffers Tools C++ Help

else
{
    long maxValue = qstr.toLong();
    PrimeThread * pt = new PrimeThread(maxValue, ch_os);
    pt -> start();
    connect(pt, SIGNAL( finished() ),
            this, SLOT( answerReady() ) );
}
}

void ClientHandler::answerReady()
{
    PrimeThread* pt = (PrimeThread*) sender();
    QString outgoing = "\n There are " +
        QString::number(pt -> total) +
        " prime numbers between 0 and " +
        QString::number(pt -> maxValue)
        + "\n";
    (* (pt -> os)) << outgoing;
    (* (pt -> os)) << "Give a number or say \"bye\": ";
    (pt -> os) -> flush();
    delete pt;
}

--(Unix)-- PrimeServer.cc (C++ Abbrev)--L99--72%-----
```



# Further Performance Improvement

- implement an efficient algorithm to count prime numbers
- create a thread only when the input number is large
- reuse threads

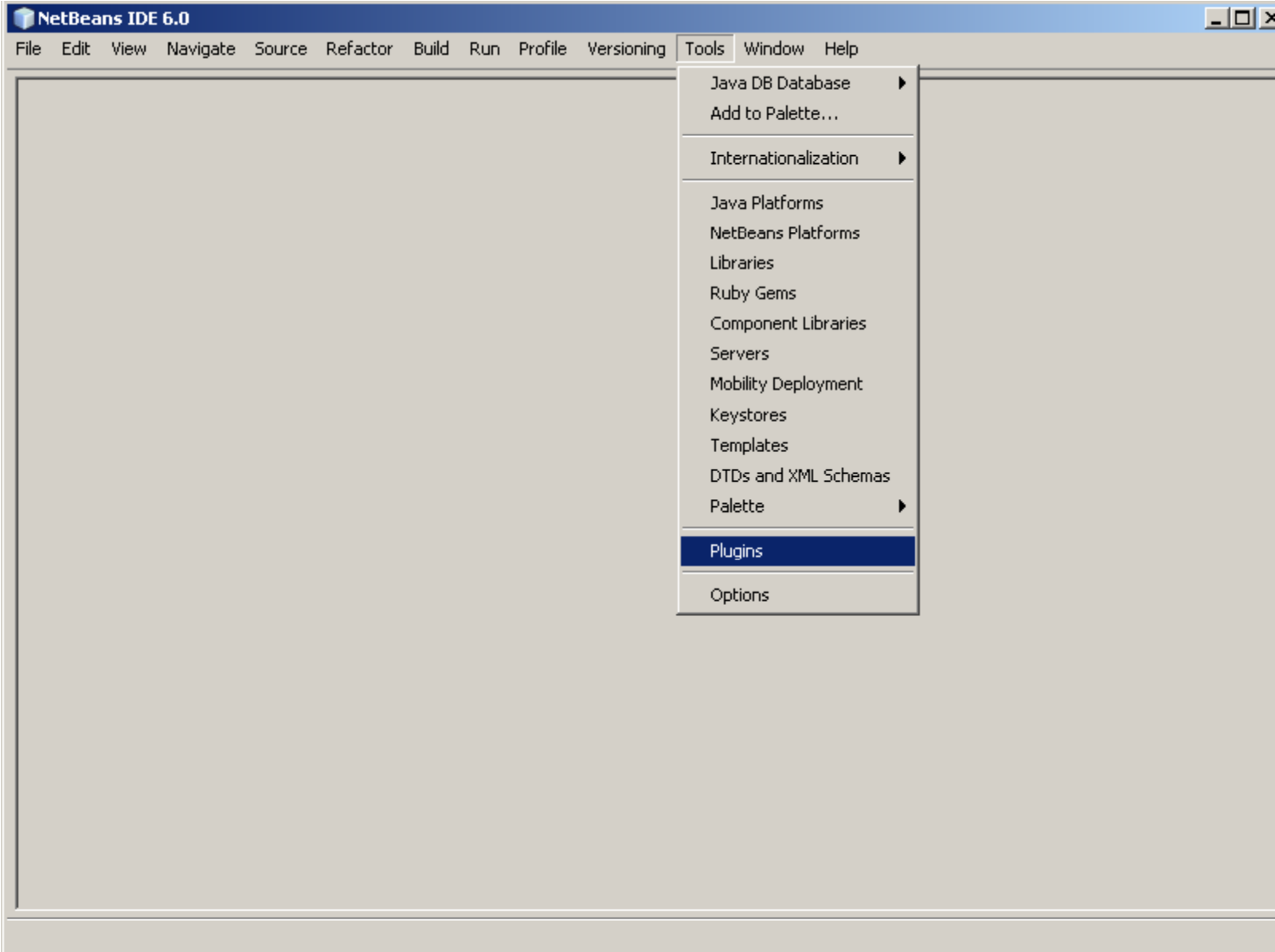
# **ECE 462**

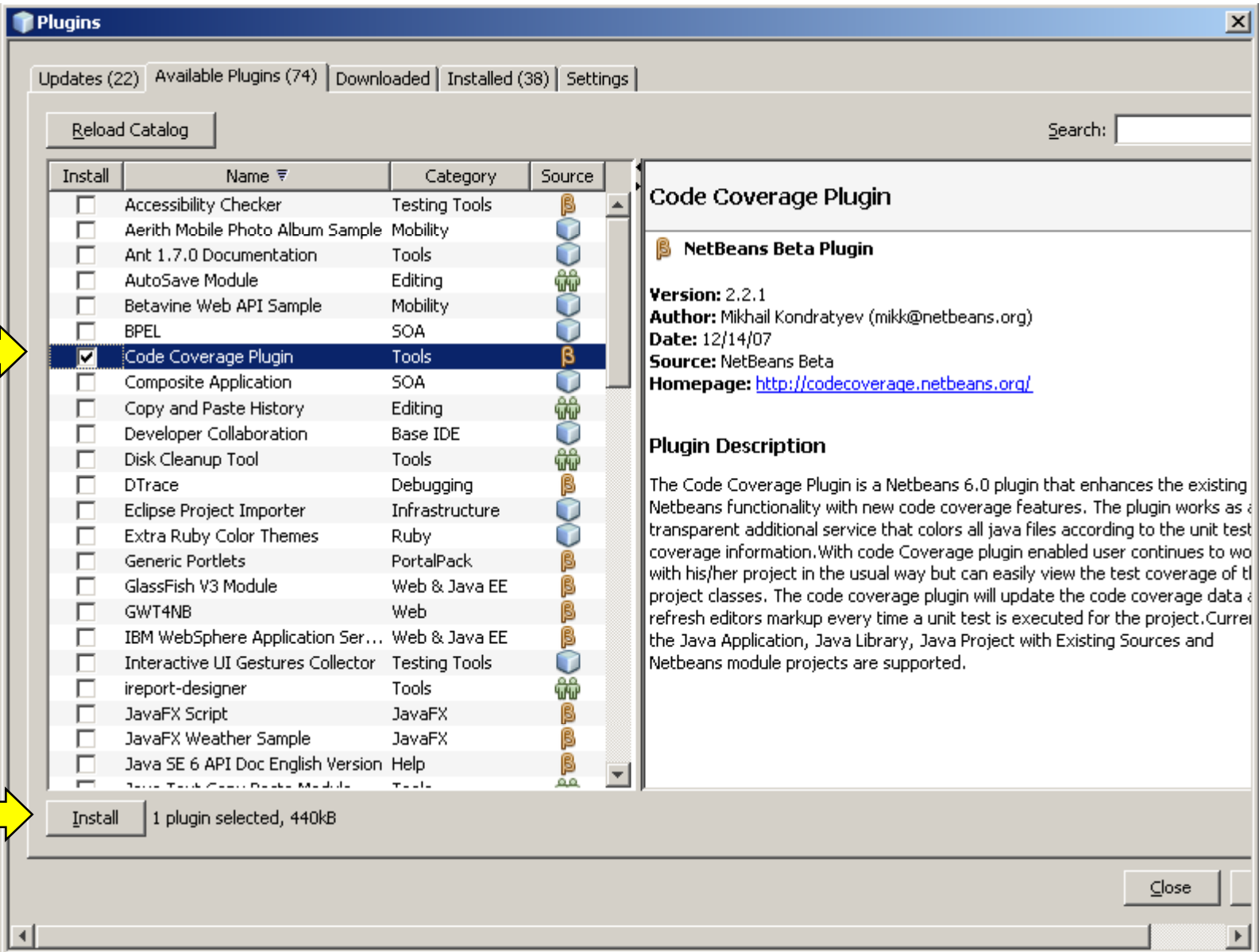
# **Object-Oriented Programming**

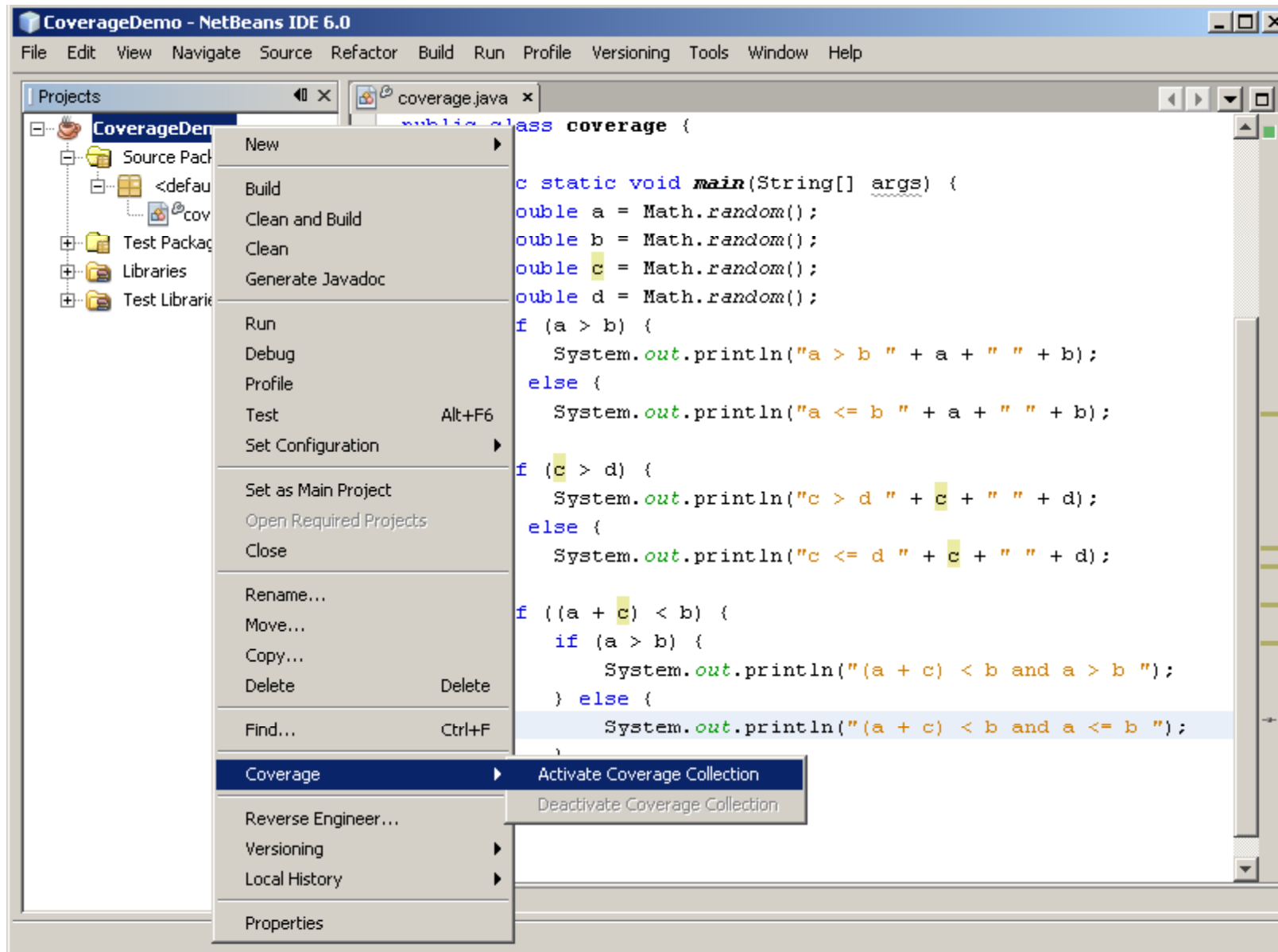
# **using C++ and Java**

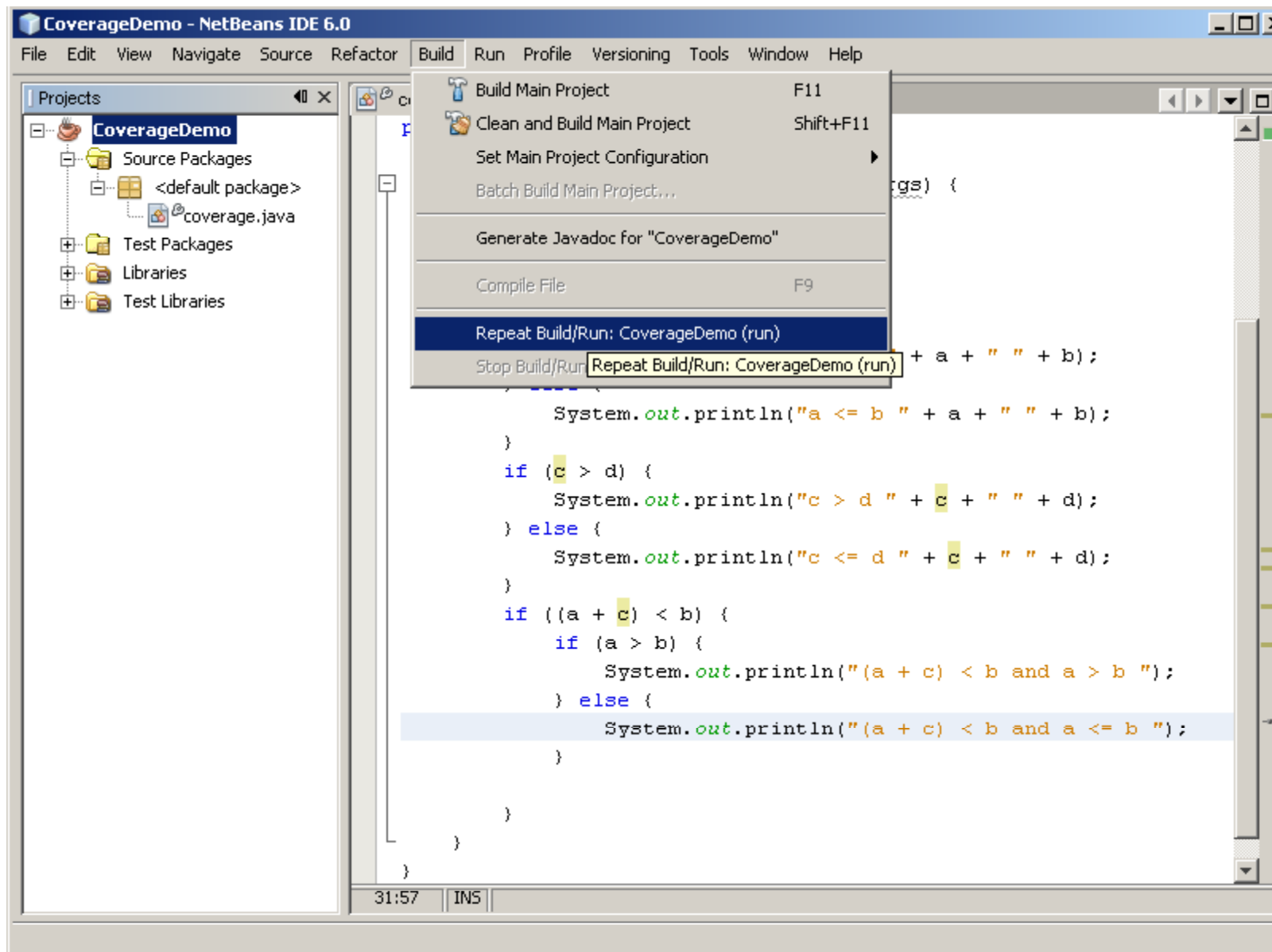
## **Test Coverage**

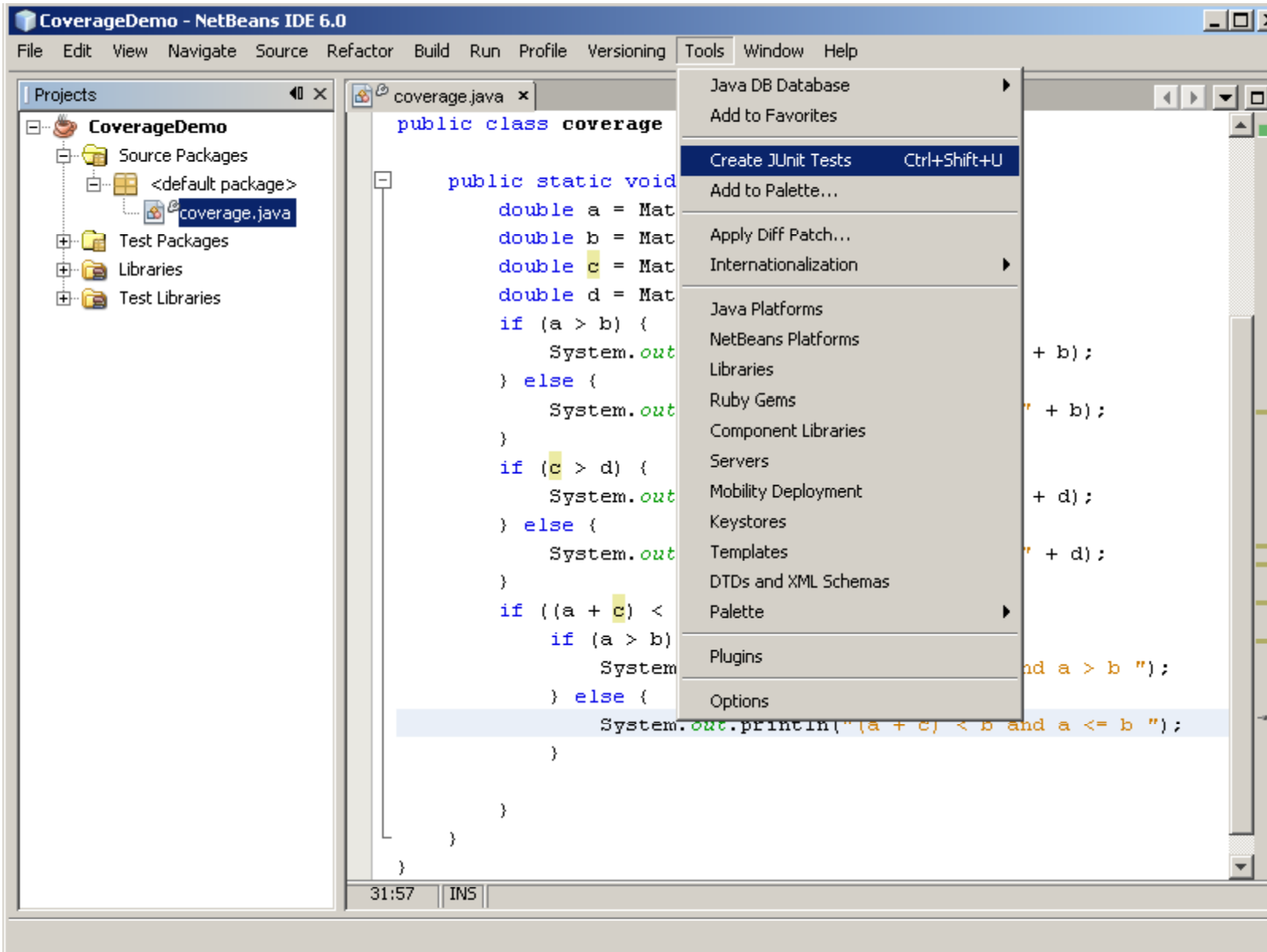
Yung-Hsiang Lu  
yunglu@purdue.edu

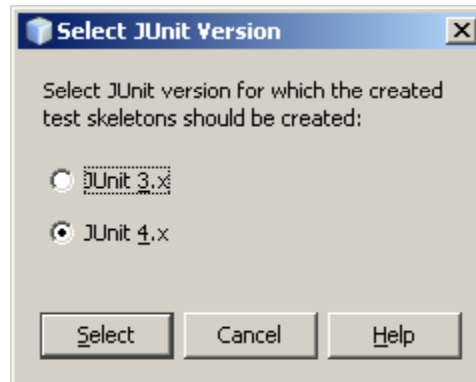




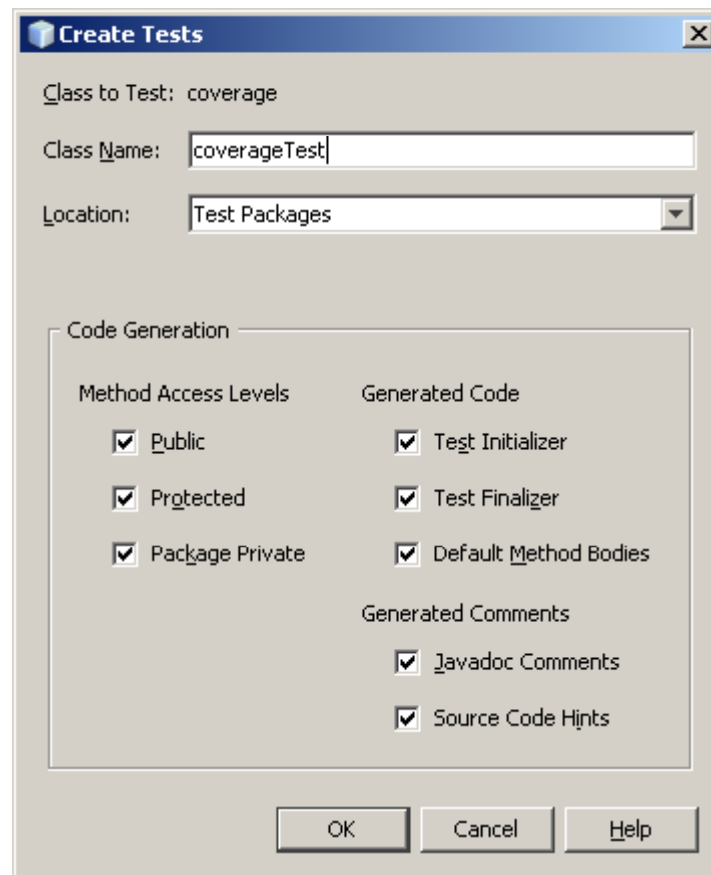


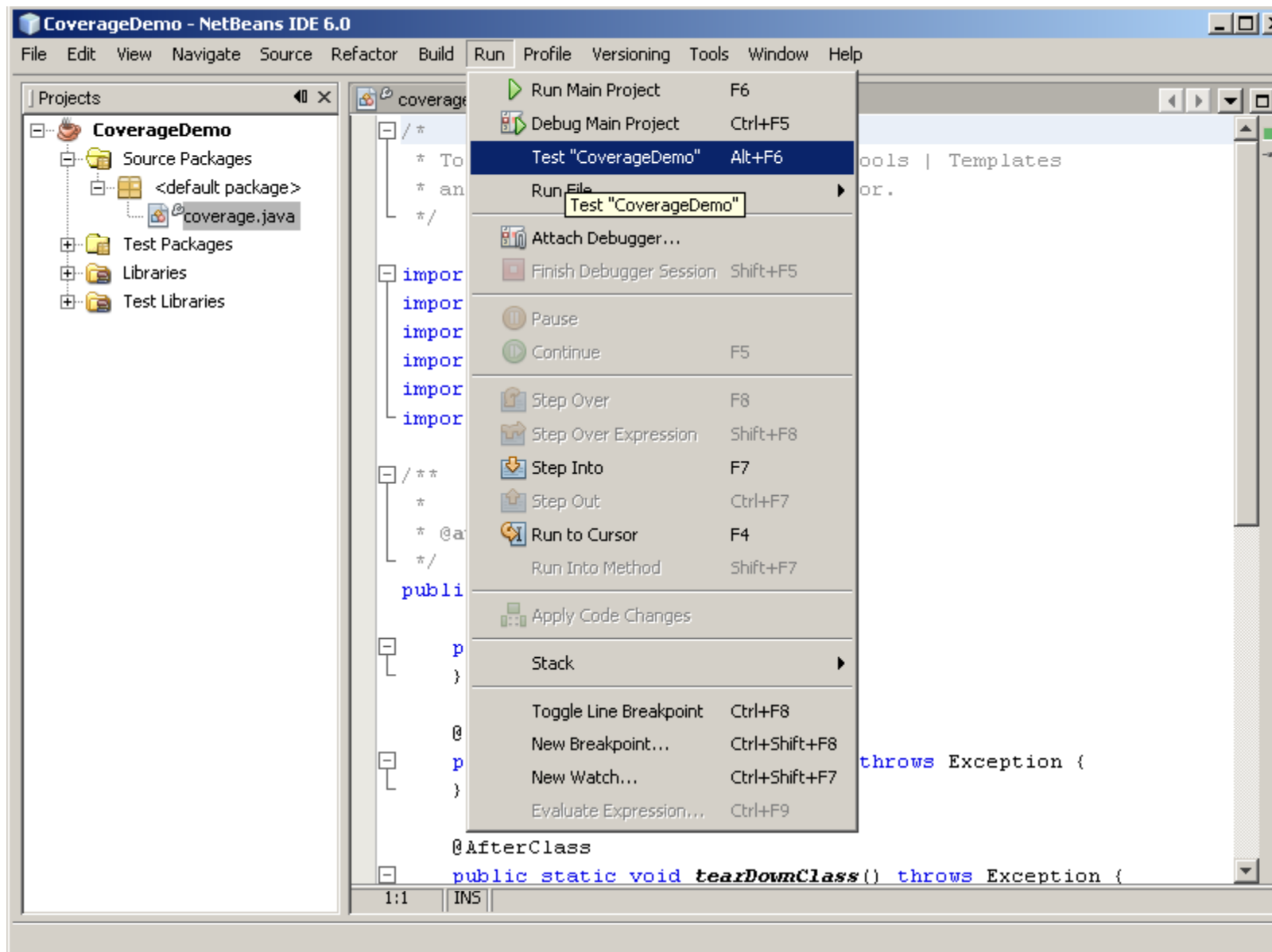


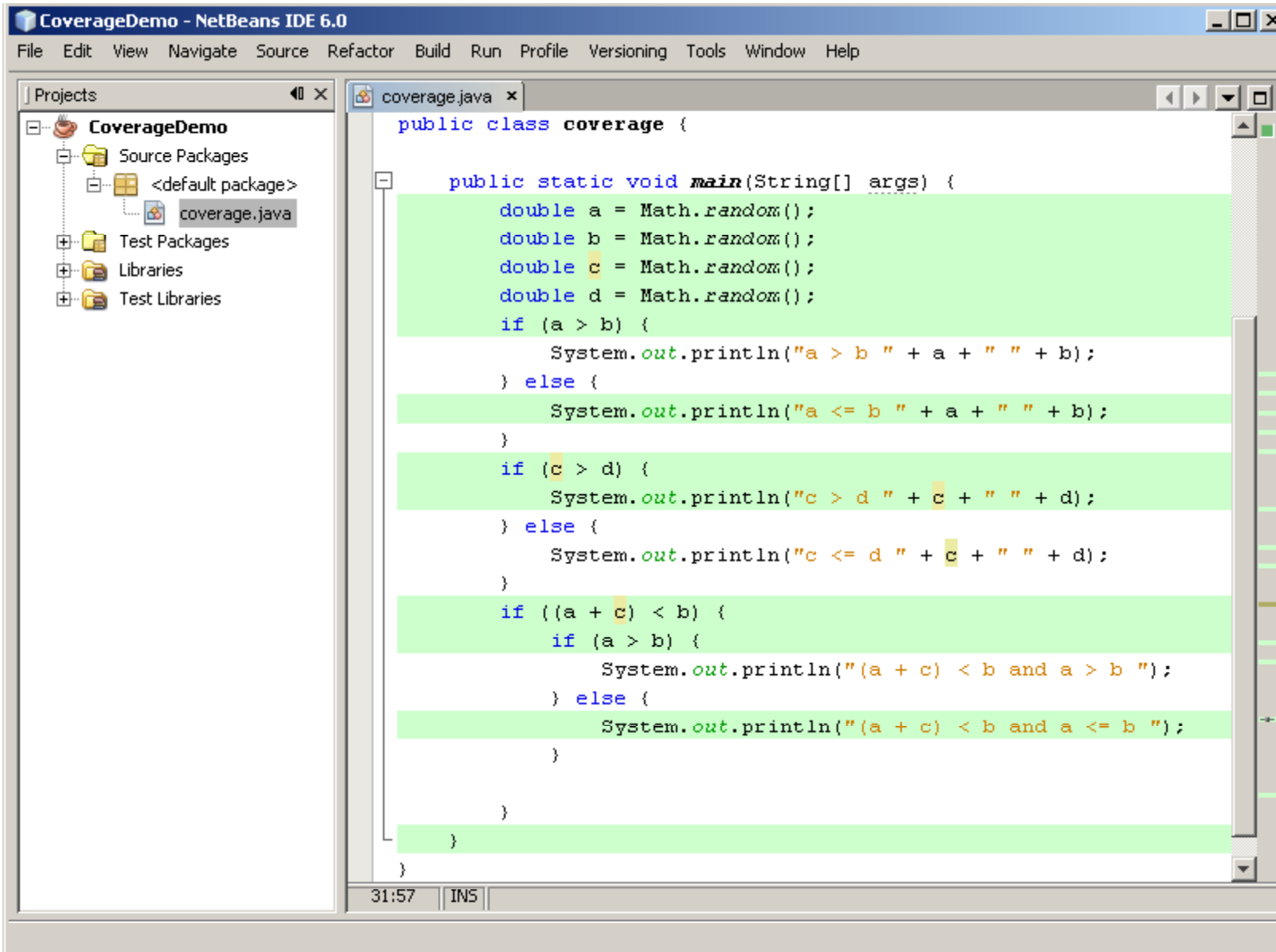


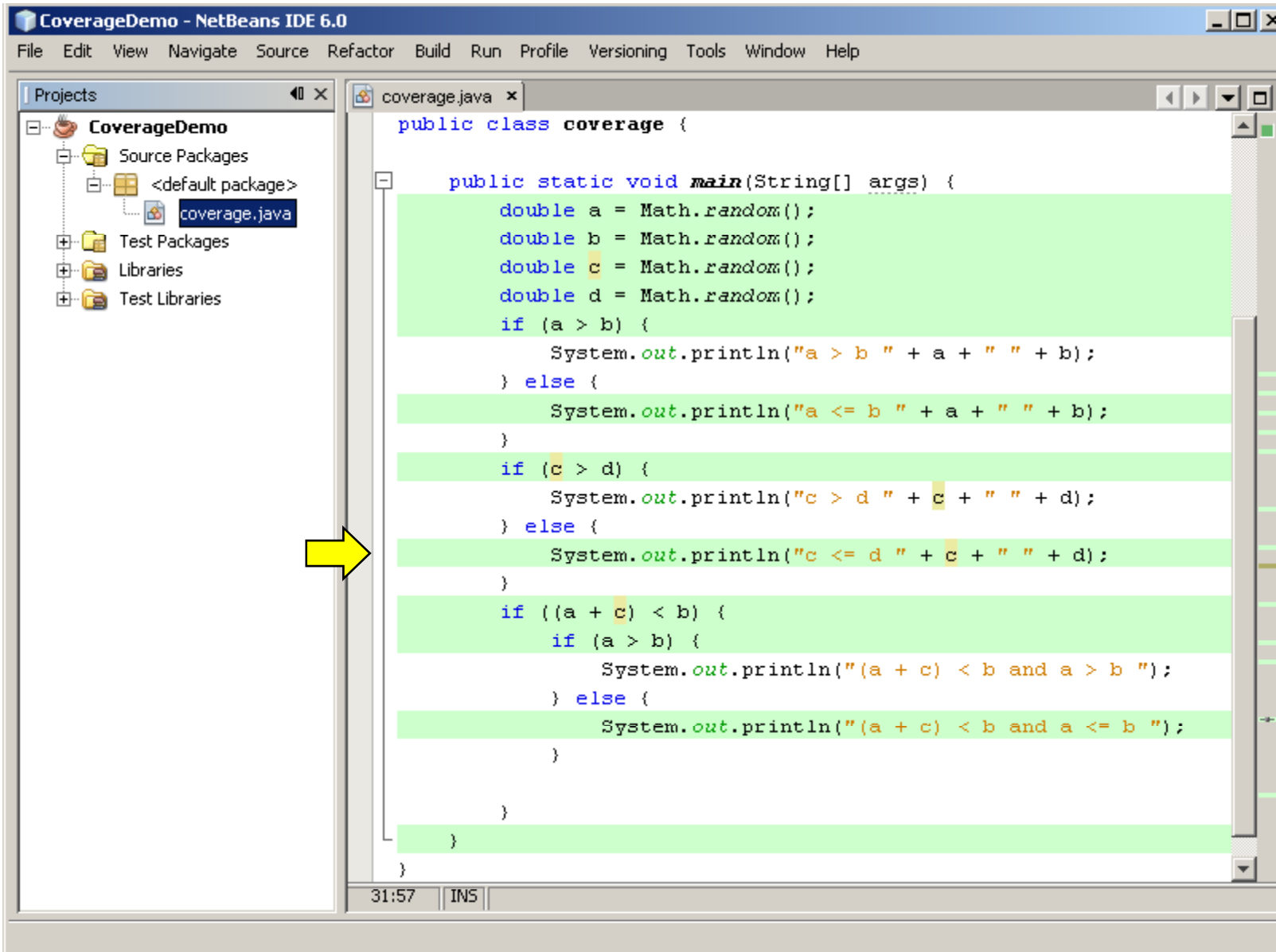


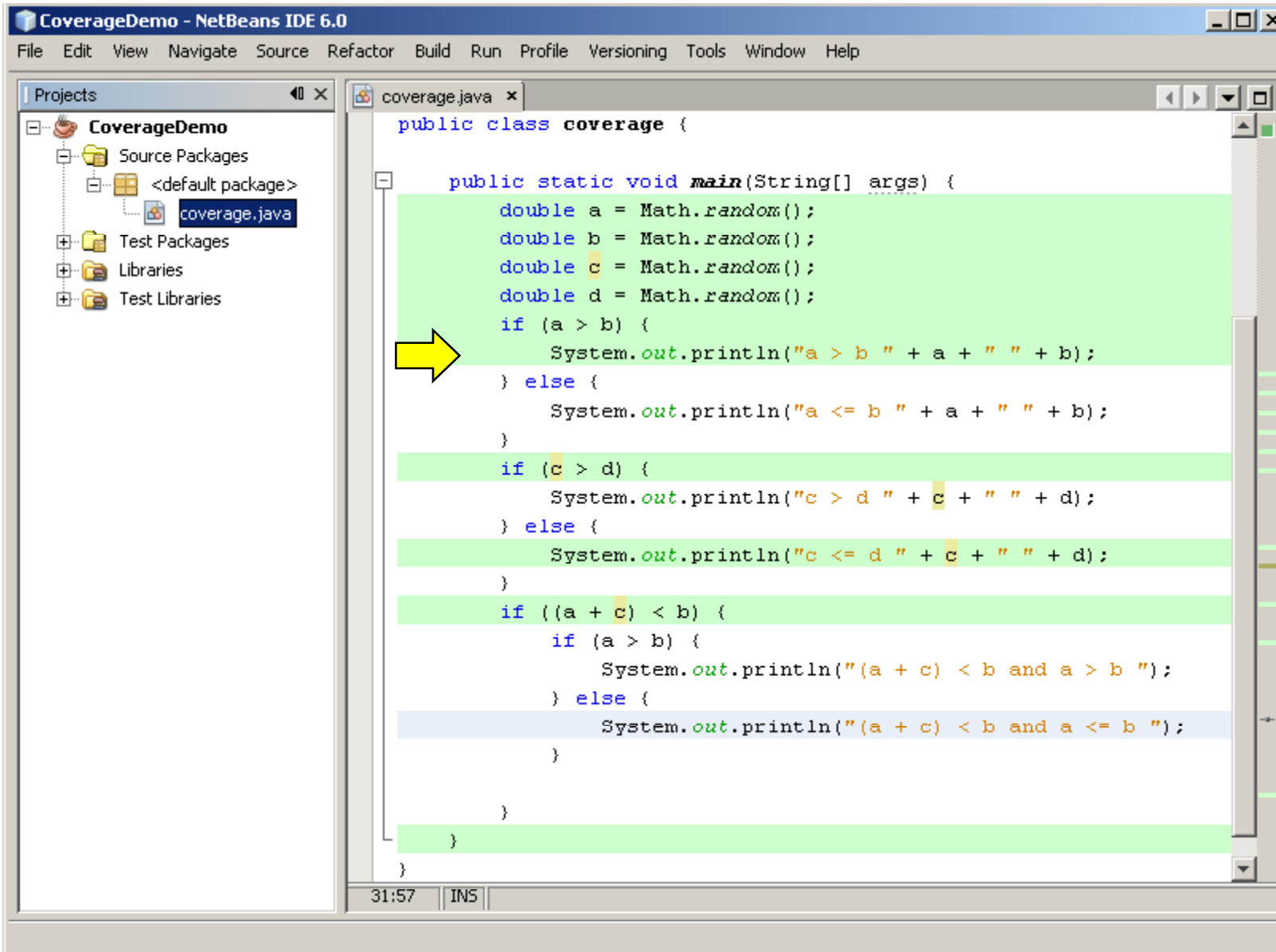












# Unreachable Code

If a, b, and c are zeros or positive numbers

$(a + c) < b \Rightarrow a > b$  is impossible

$\Rightarrow$  problem in the logic?

```
if ((x <= 0) && (x >= width)) // hit left or right wall
{
    // width > 0
    vx = -vx; // change direction
}
```