ECE 462 C++ and Java

Lab Exercise 1
Netbeans

Yung-Hsiang Lu
yunglu@purdue.edu
Goals

- learn the Netbeans working environment
- write, compile, and execute Java programs
- set up a code repository for version control
- create a simple user interface with menu and buttons
http://www.netbeans.org/
Create a Project in Netbeans
Lab 01 Netbeans
Lab 01 Netbeans
Lab 01 Netbeans
Lab 01 Netbeans
```java
package javaapplication1;

public class Main {

    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```
package javaapplication1;

/**
 * @author yunglu
 */

public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
package javaapplication1;

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello Java");
    }
}

Finished building JavaApplication1 (run).
Sample Programs from Textbook
Programming with Objects

A Comparative Presentation of Object-Oriented Programming with C++ and Java

(A new book authored by Avi Kak and published by John-Wiley)
(The book was released by the publisher on March 21, 2003.)
The second printing came out in mid 2005. The second printing incorporates a large number of corrections.

ERRATA  ERRATA  ERRATA
errata updated August 11, 2006

NOTE FOR INSTRUCTORS
AND FOR THOSE ENGAGED IN SELF-STUDY

A solution manual for the homework problems is now available free of charge. If you'd like a copy, please send your request to objects@programmingwithobjects.com. Please include the following string in the subject line: "Requesting solution manual for PWO".

CODE EXAMPLES NOW AVAILABLE FOR DOWNLOAD
// This code example is from the following source:
//
// Book Title: Programming with Objects, A Comparative Presentation
// of Object Oriented Programming with C++ and Java
//
// Chapter: Chapter 3 ---- The NOTION Of A Class And Some Other Key Ideas
//
// Section: Section 3.2 -- Defining A Class In Java
//
// The links to the rest of the code in this book are at
//
//
// For further information regarding the book, please visit
//
// http://programming-with-objects.com

// User.java

class User { // |D|
    private String name;
    private int age;

    public User(String str, int yy) { name = str; age = yy; }
    public void print() {
        System.out.println("name: " + name + " age: " + age);
    }
}

class Test { // |E|
    public static void main(String[] args) {
        User u = new User("Zaphod", 23);
        u.print();
    }
}

Done
```java
// Class User

class User {
    private String name;
    private int age;

    public User(String str, int yy) { name = str; age = yy; }

    public void print() {
        System.out.println("name: " + name + " age: " + age);
    }
}

class Main {

    /*
     * Parse args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        User u = new User("Geeks", 23);
        u.print();
    }

    // Main entry point
}
```

Lab 01 Netbeans
Lab 01 Netbeans

YHL
```java
package javaapplication1;

/**
 * @author yunglu
 */

class User {

    private String name;
    private int age;

    public User(String str, int yy) {
        name = str;
        age = yy;
    }

    public void print() {
        System.out.println("name: " + name + " age: " + age);
    }

    public class Main {

        /**
         * @param args the command line arguments
         */
        public static void main(String[] args) {
            // TODO code application logic here
            User u = new User("Zaphod", 33);
            u.print();
        }
    }
}
```
```java
package javaapplication1;

class User {

    private String name;
    private int age;

    public User(String str, int yy) {
        name = str;
        age = yy;
    }

    public void print() {
        System.out.println("name: " + name + " age: " + age);
    }

    public class Main {
```

Output - JavaApplication1 (run)

```java
public class Main {
```
Create Graphical User Interface
Lab 01 Netbeans

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

public static void main(String[] args) {
    // TODO code application logic here
}
```
The Inspector window displays a tree hierarchy of components in the opened frame.
Add Menu
Lab 01 Netbeans
Lab 01 Netbeans
Run the Program
import javaapplication1;

author yunglu

class Main {

    public static void main(String[] args) {
        // TODO code application logic here
        NewFrame njf = new NewFrame();
        njf.setVisible(true);
    }
}
Version Control
```bash
$ set prompt="[\(\%m\) \(\%~\) ] "
[ee462b30@msee190pc3 ~]$ mkdir projects
[ee462b30@msee190pc3 ~]$ cd projects/
[ee462b30@msee190pc3 ~/projects/] $ mkdir CVSROOT
[ee462b30@msee190pc3 ~/projects/] $ 
```
more .cshrc
source /home/shay/a/sfwtools/public/settings
[msee190pc3] ~ ] which java
/home/shay/a/sfwtools/public/jdk1.6.0_02/bin/java
[msee190pc3] ~ ] java -version
java version "1.6.0_02"
Java(TM) SE Runtime Environment (build 1.6.0_02-b05)
Java HotSpot(TM) Server VM (build 1.6.0_02-b05, mixed mode)
[msee190pc3] ~ ] which netbeans
/home/shay/a/sfwtools/public/netbeans/bin/netbeans
[msee190pc3] ~ ]
Lab 01 Netbeans
Lab 01 Netbeans

Steps

1. CVS Root
2. Folder to Import

CVS Root

Specify location of CVS repository defined by CVS root.

CVS Root: [Enter CVS Repository]

IDE directly supports following CVS root types:

:pserver:username@hostname:/repository_path
   for connecting to CVS pservers repositories
:ext:username@hostname:/repository_path
   for connecting CVS repository using SSH, etc.

Following types require external cvs program:

:fork:/repository_path
   for protocol access to local CVS repository
:local:/repository_path
   for direct access to local CVS repository
Lab 01 Netbeans
Lab 01 Netbeans

```java
package javaapplication1;

public class Main {

  /**
   * @param args the command line arguments
   */
  public static void main(String[] args) {
    // TODO code application logic here
    JFrame nbf = new NewFrame();
    nbf.setVisible(true);
  }

  Log
}
```
always update before commit
Commit Message:

add change log

Files to Commit:

<table>
<thead>
<tr>
<th>File</th>
<th>Status</th>
<th>Commit</th>
<th>Repository path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main.java</td>
<td>Locally Mod...</td>
<td>Commit</td>
<td>.../JavaApplication1/src/javaapplication1</td>
</tr>
</tbody>
</table>
package javaapplication1;

public class Main {

    public static void main(String[] args) {
        // TODO code application logic here
        NewJFrame njf = new NewJFrame();
        njf.setVisible(true);
    }
}
Lab 01 Netbeans
Add Button
Lab 01 Netbeans
Lab 01 Netbeans
### Lab 01 Netbeans

#### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>null</td>
</tr>
<tr>
<td>background</td>
<td>[236, 233, 216]</td>
</tr>
<tr>
<td>componentPopupMenu</td>
<td>&lt;none&gt;</td>
</tr>
<tr>
<td>font</td>
<td>Tahoma 11 Plan</td>
</tr>
<tr>
<td>foreground</td>
<td>[0, 0, 0]</td>
</tr>
<tr>
<td>icon</td>
<td></td>
</tr>
<tr>
<td>mnemonic</td>
<td>0, 0, 0</td>
</tr>
<tr>
<td>text</td>
<td>Green</td>
</tr>
</tbody>
</table>

#### Other Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UClassID</td>
<td>ButtonUI</td>
</tr>
<tr>
<td>actionCommand</td>
<td>jButton1</td>
</tr>
<tr>
<td>alignmentX</td>
<td>0.0</td>
</tr>
<tr>
<td>alignmentY</td>
<td>0.5</td>
</tr>
<tr>
<td>autoscrolls</td>
<td></td>
</tr>
<tr>
<td>border</td>
<td>[XPEmptyBorder]</td>
</tr>
<tr>
<td>borderPointed</td>
<td></td>
</tr>
<tr>
<td>buttonGroup</td>
<td>&lt;none&gt;</td>
</tr>
<tr>
<td>contentAreaFilled</td>
<td></td>
</tr>
<tr>
<td>debugGraphicsOptions</td>
<td>NO_CHANGE5</td>
</tr>
<tr>
<td>defaultCapable</td>
<td></td>
</tr>
</tbody>
</table>

### text

(java.lang.String) The button's text.
### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>null</td>
</tr>
<tr>
<td>background</td>
<td>[236,233,216]</td>
</tr>
<tr>
<td>componentPopupMenu</td>
<td>&lt;none&gt;</td>
</tr>
<tr>
<td>font</td>
<td>Tahoma 11 Plan</td>
</tr>
<tr>
<td>foreground</td>
<td>[0,0,0]</td>
</tr>
<tr>
<td>icon</td>
<td>Green</td>
</tr>
<tr>
<td>mnemonic</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td></td>
</tr>
<tr>
<td>tooltipText</td>
<td>null</td>
</tr>
<tr>
<td>otherProperties</td>
<td>null</td>
</tr>
<tr>
<td>UIClassID</td>
<td>ButtonUI</td>
</tr>
<tr>
<td>actionCommand</td>
<td>Green</td>
</tr>
<tr>
<td>alignmentX</td>
<td>0.0</td>
</tr>
<tr>
<td>alignmentY</td>
<td>0.5</td>
</tr>
<tr>
<td>autoscrolls</td>
<td></td>
</tr>
<tr>
<td>border</td>
<td>XPEmptyBorder</td>
</tr>
<tr>
<td>borderSelected</td>
<td></td>
</tr>
<tr>
<td>buttonGroup</td>
<td>&lt;none&gt;</td>
</tr>
<tr>
<td>contentAreaFilled</td>
<td></td>
</tr>
<tr>
<td>debugGraphicsOptions</td>
<td>NO_CHANGES</td>
</tr>
<tr>
<td>defaultCapable</td>
<td></td>
</tr>
</tbody>
</table>

**foreground**

*(java.awt.Color)* The foreground color of the component.
Lab 01 Netbeans
Commit Message:
add four buttons and a panel

Files to Commit:

<table>
<thead>
<tr>
<th>File</th>
<th>Status</th>
<th>Commit</th>
<th>Repository path</th>
</tr>
</thead>
<tbody>
<tr>
<td>New.JFrame.java</td>
<td>Locally Mod.</td>
<td>Commit</td>
<td>../javaApplication1/src/javaapplication1</td>
</tr>
<tr>
<td>New.JFrame.form</td>
<td>Locally Mod.</td>
<td>Commit</td>
<td>../javaApplication1/src/javaapplication1</td>
</tr>
</tbody>
</table>
Create a Derived Class and Override paintComponent
```
package javaapplication1;

import java.awt.Graphics;

/**
 * @author yunlu
 */

class ShapePanel extends javax.swing.JPanel {
    public ShapePanel () {
        setSize(300, 300);
    }
}

@Override
public void paintComponent(Graphics gfx) {
    gfx.setColor(java.awt.Color.BLUE);
    gfx.fillOval(100, 100, 50, 50);
}

class NewJFrame extends javax.swing.JFrame {

    /** Creates new form NewJFrame */
    public NewJFrame () {
        initComponents();
        sp = new ShapePanel();
        add(sp);
    }

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

    Generated Code
```

```java
/**
 * This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * 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 NewFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
// End of variables declaration

ShapePanel sp;
```
Lab 01 Netbeans
```java
import java.awt.Color;
import javax.swing.JPanel;

public class ShapePanel extends JPanel {

    private Color spColor;
    private boolean spCircle;

    public ShapePanel() {
        setSize(300, 300);
        spColor = Color.blue;
        spCircle = true;
    }

    public void setShape(boolean isCircle) {
        spCircle = isCircle;
        repaint();
    }

    public void setColor(Color c) {
        spColor = c;
        repaint();
    }

    @Override
    public void paintComponent(Graphics g) {
        g.setColor(spColor);
        if (spCircle) {
            g.fillOval(100, 100, 50, 50);
        } else {
            g.fillRect(100, 100, 50, 50);
        }
    }
}
```
Handle Button Action
Lab 01 Netbeans
/** Creates new form NewJFrame */
public NewJFrame()
{
    initComponents();
    sp = new ShapePanel();
    add(sp);
}

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    sp.setColor(Color.green);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    sp.setColor(Color.blue);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[])
{
    java.awt.EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            new NewJFrame().setVisible(true);
        }
    });// Embassy of SE2022

    YHL
    Lab 01 Netbeans
```java
/**
 * This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sp.setColor(Color.green);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sp.setColor(Color.blue);
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sp.setShape(true);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sp.setShape(false);
}

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

```
Please remember to commit all changes in Netbeans before creating a zip file of the repository.

```
[(msee190pc3) ~/projects/ ] ls
CVSROOT/ JavaApplication1/
[(msee190pc3) ~/projects/ ] zip -r lab01.zip JavaApplication1/
  adding: JavaApplication1/ (stored 0%)
  adding: JavaApplication1/manifest.mf, v (deflated 44%)
  adding: JavaApplication1/build.xml,v (deflated 65%)
  adding: JavaApplication1/.cvsignore, v (deflated 47%)
  adding: JavaApplication1/test/ (stored 0%)
  adding: JavaApplication1/src/ (stored 0%)
  adding: JavaApplication1/src/javaapplication1/ (stored 0%)
  adding: JavaApplication1/src/javaapplication1/NewJFrame.java,v (deflated 76%)
  adding: JavaApplication1/src/javaapplication1/NewJFrame.form,v (deflated 82%)
  adding: JavaApplication1/src/javaapplication1/Main.java,v (deflated 48%)
  adding: JavaApplication1/nbproject/ (stored 0%)
  adding: JavaApplication1/nbproject/project.xml,v (deflated 55%)
  adding: JavaApplication1/nbproject/project.properties,v (deflated 59%)
  adding: JavaApplication1/nbproject/gengenfiles.properties,v (deflated 50%)
  adding: JavaApplication1/nbproject/build-impl.xml,v (deflated 85%)
  adding: JavaApplication1/nbproject/.cvsignore,v (deflated 47%)
[(msee190pc3) ~/projects/ ]
```
Submission: A zip file of the CVS repository
Submission in Blackboard

You **MUST** submit through this procedure. Email submission is **not** accepted. Late submission is not accepted.
Lab 01 Netbeans

To view a list of available courses or to create an account, visit the Entry Page. Forgot your password? Receive your user name and a new password via e-mail.

Browser Check
Before logging in, it is highly recommended that you perform a browser check to see if your computer is properly configured to use the Blackboard Learning System.

Check Browser

Pop-up Blockers
The Blackboard Learning System sometimes uses small browser windows to deliver requested content. Some pop-up blockers incorrectly interpret these small windows as pop-ups and block them. If you have installed a pop-up blocker, turn it off while using the Learning System.

Java Security Certificate
After logging in to the Blackboard Learning System, you may receive a security certificate for a Java applet. You can trust this certificate and accept it.
Click
Lab 01 Netbeans

Click
Due Date: February 28, 2008 12:00 PM
Type: Work individually
Grading Criteria: Alphanumeric

Instructions:

Attachments:

Submission:
Lab 01 Netbeans
Select A File From Your Computer
Lab 01 Netbeans
Lab 01 Netbeans

Click Submit

Check File Name

Add Comment:
Confirmation

- Demo Student: You have submitted your assignment (Demo Student).
  Submitted Time: February 28, 2008 11:49 AM

OK
Lab 01 Netbeans

Submitted

Take Back for Resubmission
You have taken back your submission and can edit it now.

Edit Submission: Lab01 (Attempt 1)

Due Date: February 28, 2008 12:00 PM
Type: Work individually
Grading Criteria: Alphanumeric

Instructions:

Attachments:

Submission:

Enable HTML Creator
Lab 01 Netbeans

Due Date: February 28, 2008 12:00 PM
Type: Work individually
Grading Criteria: Alphanumeric

Status: In Progress (Attempt 1)
You are **strongly** encouraged to submit your partial assignment early.

You can submit improvements before the deadline.

Please remember that late submission is not accepted for any reason.