

Creating Java Programs and Maintaining Your Web Site

Introduction

This handout provides step-by-step instructions for creating and running a Java program. **Follow these instructions carefully, as there are many ways to make mistakes!** It is important to type commands *exactly* as given, and do **not** change the names of files or projects, etc. We will need several pieces of software to create Java programs and display applets for our labs this semester. All of these packages are available for your use in the Old Main 3041 computer lab*.

- The Java development kit (or JDK)
- An integrated development environment (or IDE) — *netbeans*
- A Java-enabled web browser — *firefox*
- The ACM Java Task Force library — *acm.zip* → *acmLibrary*

Throughout the semester, you will be asked to maintain a portfolio of your lab work, posting finished products — Java applets — on your EIU personal website. To do this, you will need to learn a little about the HTML language and understand where and how to place files on the EIU web server. This handout also provides instructions which detail how to do this.

The ACM Java Libraries — One Time Only

The Java programs described in our textbook use the ACM Java libraries to simplify the task of writing programs with Java. (Note: we have added a couple of locally written programs to the file we're making available on our web site.) To use these libraries, some initial steps are needed. Once you have prepared these libraries for use, *you won't need to do this again*.

1. Start **netbeans**: locate the bluish–silver cube on the dock at the bottom of the screen. When you mouse over it, **NetBeans 7.0** should display. Click on the cube.
You may be asked if you wish to participate in automatic user feedback — simply answer **NO**. If you are asked to register NetBeans, click on **NEVER REGISTER**. The first time you run NetBeans, a number of files will be created, so this may take a few moments.
2. Create a place to store the **acmLibrary** and some other useful files
 - (a) After the NetBeans window appears, click the **File** menu, then select **New Project**. (Hereafter, sequences like this will be shown as **File→New Project**.)
 - (b) Within the *Categories* panel, select **Java**.
 - (c) Within *Projects*, select **Java Class Library**.
 - (d) Click **Next** to proceed.
 - (e) For the project name, enter (exactly) the word: **acmLibrary**
In the project location, remove **NetBeansProjects** from the path name and replace it with **mat2170**, giving a result similar to:

Project Location:	/Users/jsbach/mat2170
Project Folder:	/Users/jsbach/mat2170/acmLibrary

then click on the **Finish** button.

*This software is available at no cost from several sources on the Internet. The Java JDK and the IDE known as NetBeans is made available by Sun Microsystems; the browser we use is Firefox and is developed by the Mozilla Foundation. The Java libraries are made available by the ACM, the Association for Computing Machinery.

3. Download the archive file `acm.zip`:
- Locate the `firefox` icon toward the left end of your dock and click on it to start the browser. In `firefox`, go to the our course website: `www.eiu.edu/~mathcs` and click on the `mat2170` link.
 - Find the link to `acm.zip` file in the `Labs` section.
 - Download by clicking the link and choosing `OPEN WITH ARCHIVE UTILITY`. This will place an `acm` folder in the `Downloads` folder of your home directory.
 - To move the `acm` folder to the `mat2170/acmLibrary/src` folder:
 - Open two Finder windows. In one, click on your home directory in the sidebar, then double-click `Downloads`. Locate the `acm` folder.
 - In the other window, click on your home directory, double-click `mat2170`, then on `acmLibrary`, then on `src`.
 - Drag the `acm` folder from the first window to the second window.
 - Close both windows by clicking on the red dot in the upper left corner.
4. Back in the `netbeans` window, select `Window` → `Output` → `Output` to open a display panel below the editor pane.
5. Within the projects panel, right-click on `acmLibrary` and select `Build`. Note the `Build Successful` message in the output window. Now, right-click `acmLibrary` again, and choose `Close`. (Do not exit `netbeans` yet.)

Note: the fewer projects you leave open, the less time it will take `netbeans` to complete scanning. If necessary you can always re-open any project you close.

At this point, you have created and compiled a project, `acmLibrary`, containing an assortment of classes which we'll use throughout the semester in various programming assignments.

Your First Java Program

Now that the ACM Java libraries have been prepared, we are ready to create our first Java program. As suggested by our textbook, a good first program to try out is the traditional “hello world” program. Here are the steps needed:

Turn off a distracting feature, create a new project

1. Autocompletion can be a very useful feature, but at this juncture it is too distracting. Within the NetBeans IDE, select `Netbeans`→`Preferences`→`Editor`, then the `Code Completion` tab. Uncheck the “Auto Popup Completion Window” and “Auto Popup Documentation Window.” Click `OK` to continue.
2. Select `File`→`New Project...`
Within the `Categories` panel, select `Java`.
Within `Projects`, select `Java Class Library`.
Click `Next` to proceed.

3. You are now able to provide a project name and assign it a location. For this example, we want to put the project in the `lab1` directory and the name of the project is to be `HelloProgram`.
- For the project name, use `HelloProgram`
 - In the project location, add `/lab1` to the path name, giving a result similar to:

Project Location: `/Users/jsbach/mat2170/lab1`

Project Folder: `/Users/jsbach/mat2170/lab1/HelloProgram`

Click **Finish** to advance.

Make the `acmLibrary` available to this project; Create an empty java file

1. Right-click on **Libraries** in this new project, and select **Add Project**. In the **Add Project** window, click on the button to take you up one level (folder with an up-arrow on it), choose `acmLibrary`, then click on **Add Project JAR files** to finish.
2. Right-click on **Source Packages**, then select **New→Other...**. When you are given the choice, select **Java** from the categories provided. The file type to select is **Empty Java File**. Click **Next**, then replace the suggested class name `NewEmpty` with `HelloProgram`, since *the file name must match the class name*. Click the **Finish** button.

Create the Java program

1. OK! Most of the administrative details are now out of the way. On the right-hand panel appears a “blank slate” which will ultimately contain the source code for your program.
- In **firefox**, on our course webpage, locate the `HelloProgram` link and click on it. The Java code should now appear in the browser window. Select the entire text and copy it to the editor panel in **netbeans**. Modify the program by replacing the author and date, then use **RIGHT-CLICK** and **FORMAT** in the **netbeans** editor window to auto-format the code to improve readability. Always format your programs in this way before submitting them electronically.

Very important: All programs you write for this class should begin with a header comment section which includes: your name, class section, lab and exercise number, file name, and the purpose of the program (a brief summary of what the program does).

2. Click on the hammer on the tool bar to build the program. If all goes well, this program will compile with no errors. (You will see *Build Successful* in the bottom **Output** panel.) Otherwise, the computer will complain about *syntax errors*. If your program has errors, carefully compare it with the code shown in Figure 1. Make corrections to the program, then click on the hammer again. Repeat this process until the program has no more errors. Ask the instructor or proctor for help if you need it.
3. Once the program is error-free, click in the editor window (with the `HelloProgram` java code) to place the focus there if it isn’t already. Select **Run→Run File** from the menus. This causes the Java applet viewer to start, and the results of your program appear in that window. After you have seen the results, close the applet viewer window (click on red dot in the top, left-hand corner).
4. *Modify* the “hello world” Java program so it displays your name instead of `hello, world`.
5. *Modify* the program again to move your name closer to the top of the applet viewer. Build the project again, then run the program to check that it is behaving properly.

```
/*
 * Author: Johann Bach
 *         MAT 2170, Section 2
 * Exercise: Lab 1, #1
 * Date: January 6, 2012
 * File: HelloProgram.java
 * Purpose: This program displays "hello, world"
 * on the screen. It is inspired by the first
 * program in Brian Kernighan and Dennis Ritchie's
 * classic book, The C Programming Language.
 *
 */

import acm.graphics.*;
import acm.program.*;

public class HelloProgram extends GraphicsProgram
{
    public void run()
    {
        // Create a phrase and display to the user
        add(new GLabel("hello, world", 100, 75));
    }
}
```

Figure 1: A first Java program, stored in the file `HelloProgram.java` by *netbeans*.

6. Insert a second `add(new GLabel())` statement after the one already in the program. Have the displayed text show your hometown and major at a new location beneath (and not overwriting) your name. Build and run the program to ensure it is correct.
7. Modify the header comments to reflect the changes in your program. Make sure it still builds and executes, then re-format one last time.
8. Select **File**→**Close Project "HelloProgram"**, to close the project.
9. It is helpful to realize where a few important files got placed when you built your main project. Figure 2 shows some of the most important files:
 - `HelloProgram.java` — the program source file – what you copied into the *netbeans* editor
 - `HelloProgram.class` — the compiled Java bytecode
 - `HelloProgram.jar` — the Java archive file for your program

Creating Your EIU Web Site

As a side benefit of the build (compilation) step, a Java “applet” was created for this program. This applet can be placed on your personal website and you or others can run this applet from anywhere else in the world, provided you have Internet access and a web browser with a Java plug-in, something that is quite common. By following the instructions in this section, you will

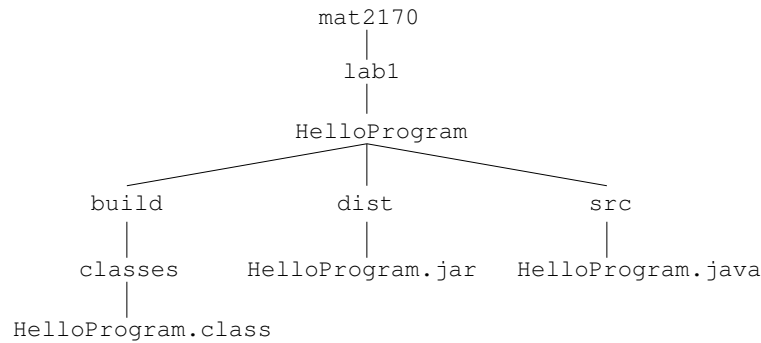


Figure 2: A few important Java-related files.

create the files needed to publish your program to your website[†]. We are hoping to have the upload script finished by next week's lab.

It is important to understand that you will maintain two copies of your entire website: one in your account in the `mathcs` lab, and one on `pen.eiu.edu`. The version in your lab account is useful for testing: you (and only you) have access to these files and can verify that your Java applets run the way you intended. The copy on `pen.eiu.edu` will be a *mirror image* of your local (`mathcs lab`) files and will be available for viewing by anyone.

When you want to make a change to your website, you will do this first in your `mathcs` lab account, testing it to be sure it has the desired appearance and that the Java applets work. When everything is correct — and *only* then — you will upload the files from your `mathcs` lab account to the EIU web server, at which point others can interact with your Java applets through your personal web page.

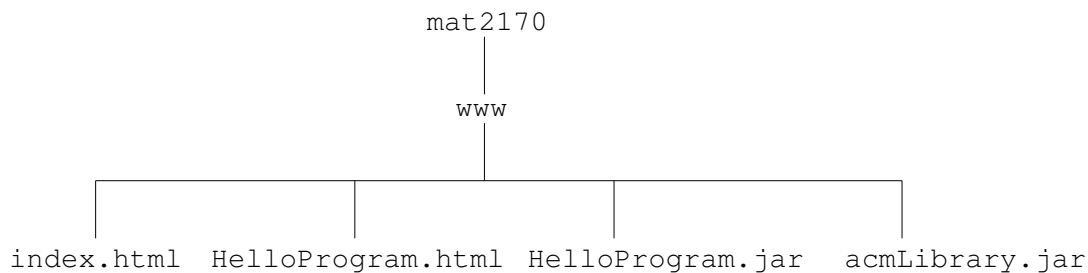


Figure 3: A few files needed for your website.

We need to create or copy four files in a directory named `www`, as shown in Figure 3. Two of these files, `HelloProgram.jar` and `acmLibrary.jar`, were created when you built the two projects. The other two `html` files you will need to create. The content of these two other files is shown in Figures 4 and 5. Be sure to personalize `index.html` by using your own name and e-mail address. Here are the directions:

[†]*Caution:* these instructions will potentially replace a website you might already have. If you are currently maintaining a website on `pen.eiu.edu`, you probably know enough about how to avoid this. If not, consult your instructor.

Getting set up — one time only

1. Create a directory named `www` in your `mat2170` directory within your home directory.
 - Open a Finder window. Locate the `mat2170` folder and drag it onto the sidebar under `PLACES` to give you quick access to it in the future.
 - Click on the new link, going to your `mat2170` folder. You should see the `acmLibrary` and `lab1` folders you created in `netbeans`.
 - Select **File** → **New Folder** and name it `www`.

This only needs to be done once. In the next set of instructions, you will place the four files shown in Figure 3 in this directory.

Create the html files

The `html` files are independent of any particular project, so they should be created separately using the `favorites` window in `netbeans`.

1. Go back to NetBeans.
2. Choose **Window**→**Favorites** to add a “Favorites” panel to the IDE. Toggle (click) the small `▷` icon next to your home directory to see its contents. Double-click on the `mat2170` directory. Find and highlight the `www` directory. Right-click on it, then select **New**→**Empty File...** Enter `index.html` for the file name. As you did before, go to the course web page in `firefox`, click on the `index.html` file, and copy the contents to your `index.html` file in the `netbeans` editor. Replace “Johann Bach” and “jsbach” with your name and userid wherever appropriate. When you are finished, choose **File**→**Save**.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>
      Johann Bach -- Home Page
    </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body>
    <center>
      <h1>Johann Bach</h1>
    </center>
    <h2>My Math 2170 Java applets</h2>
    Lab 1: <a href="HelloProgram.html" target="_blank">Hello, world!</a> <br>
    <hr>
    <address>
      Johann Bach, <a href="mailto:jsbach@iu.edu">jsbach@iu.edu</a>
    </address>
    <hr>
  </body>
</html>
```

Figure 4: A bare-bones web page, stored in the file `index.html`.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>
      HelloProgram applet
    </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body bgcolor="#0066FF">
    <!--[if !IE]> Firefox and others will use outer object -->
    <object classid="java:HelloProgram.class"
      type="application/x-java-applet;jpi-version=1.6.0"
      archive="HelloProgram.jar,acmLibrary.jar"
      height="200" width="300" >
    <!--![endif]-->
    <!-- MSIE Microsoft Internet Explorer will use inner object -->
    <object classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
      codebase="http://java.sun.com/update/1.6.0/jinstall-6-windows-i586.cab"
      height="200" width="300" >
      <param name="code" value="HelloProgram">
      <param name="archive" value="HelloProgram.jar,acmLibrary.jar">

      <strong>This browser does not have a Java Plug-in.<br >
        <a href="http://java.sun.com/products/plugin/downloads/index.html">Get
          the latest Java Plug-in here.</a>
      </strong>
    </object>
    <!--[if !IE]> close outer object -->
  </object>
  <!--![endif]-->
</body>
</html>

```

Figure 5: A web page for your Java applet, stored in the file `HelloProgram.html`.

-
3. In a similar manner, create the file `HelloProgram.html`, copying the contents of Figure 5 from the `HelloProgram.html` file link on our web page. Choose **File**→**Save** when you've finished.
 4. Please note the occurrence of:

```
height="200" width="300" >
```

on two lines in this file. These are what control the size of the java window when the program executes. Often, we will need larger windows. For a medium-sized window, set both height and width to 500. Some of our graphics windows will require a height and width of 700. About the largest graphics window ever used in `mat2170` is 800×800 .

Copy jar files into www

-
1. Create a copy of the ACM library Java archive file, which need only be done **once**: In the **favorites** pane, copy `acmLibrary.jar` from `mat2170/acmLibrary/dist` into `mat2170/www`.

2. Make a copy of the program jar as well, copying `HelloProgram.jar` from `mat2170/lab1/HelloProgram/dist` into `mat2170/www`.

Testing

1. Now try it out. In netbeans, click on the `index.html` editor tab to display the file.
2. Right-click in the `index.html` editor window and choose **View**. In response, you should see a bare-bones website with your name, a link to your first Java applet and a few other elements. If you click on the applet link, a new window/tab should open and the “hello world” program should run.

If there is a problem, compare your files to the text given in the figures. If you cannot locate the error, ask the instructor or proctor for help.

Your Second Java Program

You have now gone through all of the mechanical aspects of creating a Java program and the html files to display the resulting applet in a browser. To reinforce these steps, you will now go through this process once more. For this second Java program, we will create a program which adds two integers, as shown below in Figure 6.

Creating a new java program

1. If NetBeans isn't already running, click on the netbeans cube to start it up.
2. Select **File**→**New Project...** Within the *Categories* panel, select **Java**. Within *Projects*, select **Java Class Library**. Click **Next** to proceed.
3. Name the project `Add2Integers` and save it in the `lab1` directory. Click **Finish** to go on.
4. Right-click on **Libraries** under this project, select **Add Project**, choose `acmLibrary`, then click on **Add Project/JAR files**.
5. Right-click on **Source Packages**, select **New**→**Empty Java File**. Replace the suggested class name with `Add2Integers`. Click the **Finish** button.
6. As before, copy the program code (as it appears in Figure 6) from the `Add2Integers` link on the course web site. Be sure to modify it to include your name and course information within the comment section, as previously discussed.
7. Click in the editor window to put the focus there, then click the task bar hammer to build the `Add2Integers` Project.
8. Click in the editor panel, then select **Run**→**Run File**. This causes the Java applet viewer to start, and the results of your program to appear there. You will need to enter two integers, then the final window displays their sum. Run it several times with various inputs. Will it accept negative numbers? Alphabetic characters? Close the applet viewer window when finished with each execution.
9. Replace `ConsoleProgram` (near the top of the program) with `DialogProgram`, then rebuild and run the program again. What changed? Are the same results generated? Repair any errors in the program, building and executing again to make sure all is well. Check that your name and other information are included in the file. When finished, **re-format** the code.

```

/*
 * Author: Johann Bach
 *         MAT 2170, Section 1
 * Exercise: Lab 1, #2
 * Date: August 28, 2007
 * File: Add2Integers.java
 * Purpose: This program requests two integers
 * from the user, adds them together, and displays
 * the sum.
 */

import acm.program.*;

public class Add2Integers extends ConsoleProgram
{
    public void run()
    { // prompt for and obtain two integers from the user,
      // then find and display their sum
      println("This program adds two integers.");
      int n1 = readInt("Enter first integer : ");
      int n2 = readInt("Enter second integer : ");
      int total = n1 + n2;
      println("The total is " + total + ".");
    }
}

```

Figure 6: The second Java program, stored in the file `Add2Integers.java` by *netbeans*.

Updating your web site

After you have built and tested the `Add2Integers` project, two files must be added to your `www` directory: `Add2Integers.html` and `Add2Integers.jar`; and the file `index.html` must be modified.

Create and update html files

- 1. The file `Add2Integers.html` will be very similar to the HTML file you created for your first program. Use Netbeans to create this file, using `Add2Integers` in place of `HelloProgram` everywhere it occurs in the figure 5. Save this file (in the `www` directory). It should be listed in favorites, and not be part of any project.
- 2. You will need to edit the `index.html` file to add the line:

```
<a href="Add2Integers.html" target="_blank">Add 2 Integers</a> <br>
```

just after the `Lab 1` line, and before the `<hr>` line. Be sure to save this file as well.

Copy jar file into www

- 1. Next, copy the `jar` file from `mat2170/lab1/Add2Integers/dist` into the `mat2170/www` directory. Use the favorites pane in netbeans and simply copy and paste.

Testing

- 1. Now try it out — open `index.html` with the Firefox browser: right-click `index.html` in

netbeans and select **View**. This will open the index in firefox.

You should see a new link to the **Add2Integers** applet. If you click on this link, a new window/tab should open and the program should run.

If there is a problem, compare your files to the text given in the figures. If you cannot locate the error, ask the instructor or proctor for help.

Close programs and logoff

- In netbeans, select **File** → **Exit**. Close the firefox browser window with **Firefox** → **Quit Firefox**.
- When all programs have been shut down, logoff: At the top left of the screen, select the **Apple icon** → **Log Out** *userid...* Before lab 2, complete the Lab 1 Postlab. You may find that you have sufficient time to complete everything today.

Be sure to **always log out of your account** before you leave lab. Failure to do so may result in catastrophic loss of files, or worse.