

Using Java development tools

The following is presented here:

- How to install and use the Sun Java Development Kit (JDK) from the Windows command line;
- Installing and using Java integrated development environments such as BlueJ, NetBeans, and Eclipse;

Installing the Java Development Kit (JDK)

1. Go to www.java.sun.com. What you most likely need for your system is the Java SE Development Kit (JDK 6) for Windows. There are many choices, including the Java Runtime Environment. Your computer probably has the runtime environment (RE), so you don't need to download that. You want JDK, whose installer was called *jdk-6-windows-i586.exe* on 3/12/08. Download the installer to your computer.
2. Run this installer. To do so, you will have to choose a folder to contain the Java compiler and tools. The installer will suggest a set of folder names, such as *Java\jdk1.5.0\bin*. Remember this path (directory location) or write it down

Using the Java compiler from the command line

1. Choose *Start / Run / Sysedit*, then use Notepad to edit *c:\autoexec.bat* to include *path=%path%;* plus the path name you saved Java to. For example, you might type the following line into *autoexec.bat*:
path=%path%;c:\Java\jdk1.5.0\bin.
This means that the command prompt will look in the *bin* directory in *jdk1.5.0* in *Java* to find the compiler, *javac.exe*.*
 2. Start the DOS command environment. In this environment, you type commands, such as *notepad* or *cd* or *md* or *dir*, to run commands or programs.
 3. Make a directory for CS I Java files (e.g., *md csI*, or use Windows Explorer).
 4. Change to this directory within the command line environment (e.g., *cd csI*).
 5. Use Notepad to edit or create a Java file, e.g., *hello.java*. (See "Using Windows Notepad" below.)
 6. At the command line, using name of Java file, type e.g. *javac hello.java* to compile your Java file. This creates an executable file, *hello.class*.
 5. Run the program using the Java runtime environment: *java hello*.
 6. Use up-arrow to repeat previous command, use up-arrow to use second-previous command. You may edit the command as well at the keyboard.
 7. Use *help* to get DOS help.
 8. Use *javac* without parameters for command-line help.
 9. You may copy the text in the DOS window by expanding it (Alt-Enter) and pressing *PrtScr*. This is useful for printing or submitting test results.
 10. (Option) You may compile and test using a batch file, *test.bat*:
javac %1.java
java %1
-
- *It appears that at least some Windows systems don't allow editing of *C:\Autoexec.bat*. Here is a work-around.
- To use the 'java' and 'javac' commands from the command line, the operating system must know where to find the executable files that execute the commands. These will be in the directory where you installed your JDK.
- Where the handout says to edit 'autoexec.bat,' please edit a different file, which you can call 'javapath.bat', putting the "path=" command there.
- At the start of each homework session when you will use the java compiler, type at the command line, 'javapath'. This will set up the connection with the path where the compiler is.
- Prof. Jeff Gao writes, "Many computer OS are multi-user by default, even you might be the only user. That means that your computer has different settings, one for the whole system, one for each individual user. The *autoexec.bat* you tried might be the one for the whole system (you can not write to it). Try to find the autoexec.bat for you(as an individual

user), it might under some directory like users\yourName..."

Using Windows Notepad as a Java editor

Notepad is a text-file editor. Text files contain no formatting. Most files with the extension '.txt' are text files.

Notepad defaults to saving all files with the '.txt' extension. The Java compiler requires use of the '.java' extension. To save a file with extension '.java' in Notepad, use quotation marks in the *Save* dialog.

The Java compiler provides error messages with line-number references. To find a line by its number in Notepad, it is necessary to have the Notepad Status Bar turned on. To do this, make sure Word Wrap is turned off (choose *Format*, and if *Word Wrap* is checked, uncheck it. Then when Word Wrap is off, you may choose *View / Status Bar* to turn on the Status Bar. The Status Bar shows the line number at the bottom of the editing window.

Using Java integrated development environments (IDEs)

Java projects and software development

Work in Java IDEs is based on the idea that each program is kept in a directory called a *project*. If a program is a single Java source file, the project may have the same name as the Java file, and the Java file will be inside the project directory.

Installing Java IDEs such as BlueJ

To conveniently compile and test Java programs, you need the Java Development Kit (JDK), from Sun Microsystems and an application that lets you edit, compile, and run your Java code. Such an application is called an integrated development environment (IDE). Eclipse, BlueJ, and JavaBeans are three free IDEs. None of these is convenient to learn, although each has documentation and tutorials. The BlueJ tutorial is available at the BlueJ site.

If you wish to use BlueJ, download its installer program at www.bluej.org and JDK's at the Sun Microsystems site (www.java.sun.com); install JDK, then BlueJ.

To compile and run an example project

1. Run BlueJ.
2. Choose *Project / Open Project* and type the name of the project, e.g., "hello". You will see one or more reddish rectangles, one of which should have the name of the project.
3. Right-click on the rectangle with the name of the project, and choose "void main()...".
4. Press the OK button.
5. A terminal window will open where program output will appear and where input may be

entered. This record of a test session may be saved as a text file.

To modify, test, and save a project

1. Open a project as above.
2. Double-click on the reddish rectangle to open the source file in the editor.
3. Make the desired changes in the Java source code.
4. Press *Compile* to compile the source file into machine code.
5. Right-click and choose *void main()...* as above to test, returning to step 3 as many times as necessary to produce a successful test.
6. Choose *Project / Save* to save the project as modified.
7. To create a new project, choose *Project / Save As* and type the new project's name.
8. To create a new Java archive (.jar file), choose *Project / Create Jar File*.