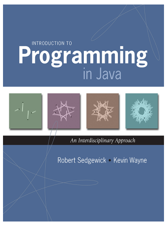
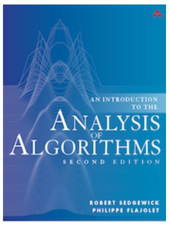
Algorithms
FOURTH EDITION
ROBERT SEDGEWICK | KEVIN WAYNE





- Web Resources
- FAQ
- Data
- Code
- Errata
- Cheatsheet
- References
- Online Course
- Lecture Slides
- Programming Assignments

Search ✕

Hello World in Java on Linux

Beta version of instructions (last updated 7/25/2015).

This document instructs you on how to setup our Java programming environment under Linux. It also provides a step-by-step guide for creating, compiling, and executing your first Java program using either DrJava or the command line. We assume some familiarity with the command line. All of the software is freely available on the web.

0. Install the Programming Environment

You will use the Java Platform, Standard Edition Development Kit (JDK 7) and DrJava.

- Log in to the user account in which you will be programming. Your account must have Administrator privileges and you must be connected to the Internet.
- Launch your shell. We'll assume that the command prompt looks like the following (though yours will likely differ):

  ```
  [username:~/]
  ```

  The `~/` is shorthand for your home directory.
- Create a directory `~/algs4` and a subdirectory `~/algs4/bin`.

  ```
  [username:~/] mkdir algs4
  [username:~/] cd algs4
  [username:~/algs4/] mkdir bin
  ```

- You will use either Oracle's implementation of the Java Platform, Standard Edition Development Kit (JDK 7) or the open-source implementation OpenJDK. Most Linux distributions provide their own mechanism for installing software. For example, on Ubuntu or Debian, type:

```
[username:~/introcs/] sudo apt-get update
[username:~/introcs/] sudo apt-get install openjdk-7-jdk
```

  If you use another distribution, use that distribution's package manager. (see the first Q+A under Troubleshooting). Feel free to use Java 8 instead of Java 7.

- Download DrJava and the wrapper script from drjava.jar and drjava to the directory `~/algs4/`.

```
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/drjava.jar
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/drjava
[username:~/algs4/] chmod 700 drjava
[username:~/algs4/] mv drjava bin
```

  The command `wget` downloads files from the web; if you don't have `wget`, try `curl -O` instead.

- Download the textbook libraries algs4.jar to `~/algs4`. Download the Java wrapper scripts from javac-algs4 and java-algs4 to `~/algs4/bin` and set the two files to be executable..

```
[username:~/algs4/] wget http://algs4.cs.princeton.edu/code/algs4.jar
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/javac-algs4
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/java-algs4
[username:~/algs4/] chmod 700 javac-algs4 java-algs4
[username:~/algs4/] mv javac-algs4 bin
[username:~/algs4/] mv java-algs4 bin
```

- Download Checkstyle 6.9 and Findbugs 3.0.1 from checkstyle.zip and findbugs.zip to `~/algs4`. Unzip `checkstyle.zip` and `findbugs.zip`. Downloads our checkstyle and findbugs configuration files from checkstyle.xml and findbugs.xml to `~/algs4`. Download the checkstyle and findbugs execution scripts from checkstyle-algs4 and findbugs-algs4 to `~/algs4/bin` and set the two files to be executable.

```
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/checkstyle.zip
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/findbugs.zip
[username:~/algs4/] unzip checkstyle.zip
[username:~/algs4/] unzip findbugs.zip
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/checkstyle.xml
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/findbugs.xml
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/checkstyle-algs4
[username:~/algs4/] wget http://algs4.cs.princeton.edu/linux/findbugs-algs4
[username:~/algs4/] chmod 700 checkstyle-algs4 findbugs-algs4
[username:~/algs4/] mv checkstyle-algs4 bin
[username:~/algs4/] mv findbugs-algs4 bin
[username:~/algs4/] mv checkstyle.xml checkstyle-6.9
[username:~/algs4/] mv findbugs.xml findbugs-3.0.1
```

- Add the directory `~/algs4/bin` to your PATH environment variable. This will depends significantly on your Linux distribution, shell, and user configuration. In the bash shell, this generally means adding the following line to either your `.bash_profile`, `.profile`, or `.bashrc` file.

```
# Add ~/algs4/bin to the PATH
export PATH=$PATH:$HOME/algs4/bin
```

This is a critical step. You may need to logout and log back in for it to take effect.

1. Create the Program in DrJava

Now you are ready to write your first Java program. You will develop your Java programs in an application called *DrJava*. DrJava features many specialized programming tools including syntax highlighting, bracket matching, auto indenting, and line numbering.

- If you use a file manager such as Konqueror or Nautilus, you can launch DrJava by double-clicking the `drjava.jar` file. Otherwise, launch DrJava from the command line by typing:

  ```
  [username:~/algs4/] drjava
  ```

- Make the following customizations.

  - Display line numbers by selecting *Edit -> Preferences -> Display Options -> Show All Line Numbers*.
  - Set the indentation level to 4 by selecting *Edit -> Preferences -> Miscellaneous -> Indent Level -> 4*.
  - Set the Java classpath by selecting *Edit -> Preferences -> Resources -> Extra Classpath -> Add* and add the following entry:

    ```
    ~/algs4/algs4.jar
    ```

- In the main DrJava window, type the Java program HelloWorld.java exactly as it appears below. If you omit even a semicolon, the program won't work.

  ```
  public class HelloWorld {
      public static void main(String[] args) {
          System.out.println("Hello, World");
      }
  }
  ```

  As you type, DrJava does the indenting for you.

- Finally, click the *Save* button to save the file. Use DrJava to create the directory `~/algs4/hello` and name the file `HelloWorld.java`. The file name is case sensitive and must exactly match the name of the class in the Java program.

2. Compile the Program from DrJava

It is now time to convert your Java program into a form more amenable for execution on a computer. To do this, click the *Compile* button. If all goes well, you should see the following message in the *Compiler Output* pane at the bottom:

```
Compilation completed.
```

If DrJava complains in some way, you mistyped something. Check your program carefully, using the error messages in the Compiler Output pane as a guide.

3. Execute the Program from DrJava

Now it is time to run your program. This is the fun part.

- Type the following in the *Interactions* pane at the bottom. By convention, we highlight the text you type in bold.

  ```
  > java HelloWorld
  ```

  If all goes well, you should see the following message:

  ```
  Welcome to DrJava.  Working directory is /Users/username/algs4/hello
  > java HelloWorld
  Hello, World
  ```

- You may need to repeat this *edit-compile-execute* cycle a few times before it works.

4. Command-Line Interface

The command-line provides capabilities beyond those available in DrJava, including redirection and piping. You will type commands in an application called the *shell*.

If you plan to take COS 217, you might want to buy the required book *Programming with GNU Software* by Loukides and Oram. It contains an overview of Unix from the user's point of view. It also describes shell fundamentals, with reference to the Bourne-Again shell (bash), Bourne shell (sh), and C shell (csh).

- To confirm that the Java compiler is installed, type the command in boldface below and check that the results match:

  ```
  [username:~/] javac-algs4 -version
  javac 1.7.0_85
  ```

  It's important that you see the number `1.7` (or `1.6` or `1.8`) for the Java version number, but the rest is not critical.

- To confirm that the Java interpreter is installed, type the command in boldface below and check that the results match:

  ```
  [username:~/] java-algs4 -version
  java version "1.7.0_85"
  OpenJDK Runtime Environment (rhel-2.6.1.2.el7_1-x86_64 u85-b01)
  OpenJDK 64-Bit Server VM (build 24.85-b03, mixed mode)
  ```

  Again, it's important that you see the number `1.7` (or `1.6` or `1.8`) for the Java version number, but the rest is not critical.

5. Compile the Program from the Shell

You will use the `javac` command to convert your Java program into a form more amenable for execution on a computer.

- From the shell, navigate to the directory containing `HelloWorld.java`, say `~/algs4/hello`, by typing the `cd` (*change directory*) commands below:

  ```
  [username:~/] cd algs4
  [username:~/algs4/] cd hello
  [username:~/algs4/hello/]
  ```

- Compile it by typing the `javac` command below:

  ```
  [username:~/algs4/hello/] javac HelloWorld.java
  [username:~/algs4/hello/]
  ```

  Assuming the file `HelloWorld.java` is in the current working directory, you should see no error messages.

- If you want to classpath in our textbook libraries, use the command `javac-algs4` instead.

6. Execute the Program from the Shell

You will use the `java` command to execute your program.

- From the shell, type the `java` command below.

  ```
  [username:~/algs4/hello/] java HelloWorld
  Hello, World
  ```

  You should see the output of the program.

- If you want to classpath in our textbook libraries, use the command `java-algs4` instead. For example, to test standard draw and standard audio type the following two commands:

  ```
  machine:~/algs4/hello username$ java-algs4 edu.princeton.cs.algs4.StdDraw
  [ displays a graphics window with some geometric shapes and text ]

  machine:~/algs4/hello username$ java-algs4 edu.princeton.cs.algs4.StdAudio
  [ plays an A major scale ]
  ```

7. Checkstyle and Findbugs

You can use Checkstyle and Findbugs to check the style of your programs and identify common bugs.

- To run Checkstyle, type the following command in the Terminal:

  ```
  machine:~/algs4/hello username$ checkstyle-algs4 HelloWorld.java
  Running checkstyle on HelloWorld.java:
  Starting audit...
  Audit done.
  ```

  Here is a list of available checks. You can customize the settings by editing the file `/Users/username/algs4/checkstyle-6.9/checkstyle.xml`.

- To run Findbugs, type the following command in the Terminal:

  ```
  machine:~/algs4/hello username$ findbugs-algs4 HelloWorld.class
  Running findbugs on HelloWorld.class:
  ```

  Here is a list of bug descriptions. You can customize the settings by editing the file `/Users/username/algs4/findbugs-3.0.1/findbugs.xml`.

Troubleshooting

My distribution of Linux is { Gentoo, Debian, Ubuntu, Fedora, Red Hat, SuSE, Mandriva, or Slackware }. How should I modify the instructions? We haven't tested out these instructions on all flavors of Linux, but the instructions should be identical except for installing Java. We recommend using your distribution's package manager (such as portage, apt, emerge, or yum) to install Java. Here are some instructions for installing OpenJDK.

Can I use a different version of Java? Yes, any version of Java 7 or Java 8 should work fine.

I had to manually enter the location of tools.jar in DrJava, but it doesn't seem to have any effect. Any suggestions? This setting doesn't take effect until you restart DrJava.

Can I use a different IDE? Yes, feel free to use another IDE (such as Eclipse) but you will have to configure the IDE properties yourself (such as the classpath).

How do I determine which shell I'm running? Type the following command:

```
[username:~/] echo $SHELL
bash
```

You shell will likey be `bash`, `tcsh`, `sh`, `ksh` or `zsh`.

When I compile or execute a program from the shell that uses one of the textbook libraries, I get an error. How can I fix this? First, make sure that you are using the `javac-algs4` and `java-algs4` wrapper scripts. Next, verify that you have the file `~/algs4/algs4.jar`. If so, it is probably an issue with the `PATH` environment variable. From the shell, type the following command to display it:

```
[username:~/] echo $PATH
```

The `PATH` environment variable should include an entry for `~/algs4/bin`.

How do I break out of an infinite loop? From DrJava, click the *Reset* button in the menubar or select the menu option *Tools -> Reset Interactions*; from the shell, type `Ctrl-c`.

When using standard input, how do I signify that there is no more data? If you are entering input from the keyboard, type `Ctrl-d` for EOF (end of file) from either DrJava or the shell.

Last modified on August 26, 2016.