

Assignment Mechanics

From ACM JTF Tutorial

September 2008

1 Getting started

In the classic book *The C Programming Language*, Brian Kernighan and Dennis Ritchie offered the following observation at the beginning of Chapter 1:

The only way to learn a new programming language is by witting programs in it. The first program to write is the same for all languages:

Print the words

```
hello, world
```

This is the big hurdle; to leap over it you have to be able to create the program text somewhere, compile it successfully, load it, run it, and find out where the output went. With these mechanical details mastered everything else is comparatively easy.

Downloading the “hello, world” programs

Taking our inspiration from Kernighan and Ritchie, our first example will be a simple program that displays the “hello, world” message. You can download the code for this program—along with a copy of the `acm.jar` library and a couple of more advanced versions of the program—from the following web address:

```
http://jtf.acm.org/downloads/Hello.zip
```

save the file and unzip it on your computer. When you have done so, you should see a directory named `Hello` that contains the following four files: `HelloConsole.java`, `HelloDialog.java`, `HelloGraphics.java`, and `acm.jar`. The first three are Java program files, the last one is the library file that contains the compiled code for the JTF packages.

Once you have successfully downloaded these files, your next step is to compile and run the `HelloGraphics` program.

```
/*
 * File: HelloGraphics.java
 * -----
 * 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 HelloGraphics extends GraphicsProgram {
    public void run() {
        add(new GLabel("hello, world", 100, 75));
    }

    /* Standard Java entry point */
    /* This method can be eliminated in most Java environments */
    public static void main(String[] args) {
        new HelloGraphics().start(args);
    }
}
```

The code for this version of the program can be found in the textbook page 25. We use Sun's Java Development Kit (JDK). If you plan to work on your own computer, you need to download JDK version 6 update 7. Once you have JDK on the system, you can compile Java programs. In Windows, you start a command window and change the working directory to this `Hello` directory. The compilation process requires two steps. The first step is to compile the `HelloGraphics.java` program by issuing the command

```
javac -classpath acm.jar HelloGraphics.java
```

After the compilation, a file `HelloGraphics.class` is created. You can then run the program by invoking the command

```
java -cp .;acm.jar HelloGraphics
```

Note that `acm.jar` is in the same directory where the programs are. So, it is easier to put `acm.jar` in the same directory where you work on assignments. On Unix or Linux platforms, the semicolon in the classpath must be replaced with a colon (:). Note that the `acm.jar` file must be specified as part of both the compilation and execution steps.

If everything is working, the computer should pop up a graphic window that looks like the figure on page 28.

If you are using one of many Integrated Development Environments (IDEs) available for Java—such as the open-source Eclipse system or NetBeans—you will need to ensure that the `acm.jar` file is included as part of the classpath, which is the list of directories and JAR files that Java searches to find class definitions that are not part of the source files. The procedure for doing so varies from system to system. Please check the documentation for your own IDE to see how one goes about specifying the classpath.

2 Eliminating the static main method

As soon as you have the `HelloGraphics` program working, it is useful to try one additional experiment. If you look at the code above, you will see that there is a `main` method at the end of the class definition. As the comment indicates, it is possible to eliminate this method in many Java environments, but not all. Open the `HelloGraphics.java` file in an editor, delete the `main` method and its associated comments, and then see if you can still compile the run the program. If so, you will be able to write shorter programs that will be much easier for novices to understand. If not, you will need to include a standardized `main` method in your programs that always looks like

```
public static void main(String[] args) {
    new MainClass().start(args);
}
```

where `MainClass` is the name of the main class.

The examples available on the JTF web site include a static `main` method to ensure that these programs will run in as many environments as possible. For clarity of presentation, however, the programs in the book eliminate the `main` method to focus attention on the substantive parts of these examples.

3 Programming style

In reality, your programs are to be read, used, and revised by others. A well documented and structured program is essential. One of the purposes of this course is to develop disciplined programming skills from the beginning. To encourage you to write good style programs, for each assignment, style counts for 50% of the total marks.

4 Marking

The class is divided into four groups, with last names start:

A–D

E–K

L–Q

R–Z

For each assignment, a TA is assigned to each group for tutoring and marking assignments.

5 Help

You can get help from:

- TAs (send an email asking questions or making an appointment). Please send them emails through WebCT, so they know the emails are from the students in the class they are TAing.
- Drop-in Centre (ITB/101)
- Instructor (during office hours or send an email making an appointment)

6 Submission

We use WebCT to submit assignments. The McMaster WebCT Login page is located at:

<http://webct.mcmaster.ca>

WebCT ID: Your MAC ID user name

Password: Your MAV ID password

After uploading your file, you must click the **submit** button. You should get an acknowledgement for the submission. We do not accept late submissions.