Compiling on the Command Line Workshop Activity

Getting Started

When you want to run a Java program, before it can be run it must be compiled. Often compilation is handled through an IDE, however it can be done on the command line as well.

Learning how to compile Java programs on the command line will help you to understand how programs are converted from text into executable programs.

It will also make you better equipped to be able to work with other languages that do not have good IDEs or are just easier to work with on the command line, for example C, C++, and Python.

For this workshop we will be working on the CS servers, so be sure to install an FTP client such as Filezilla. Also if you are using Windows, be sure to install PuTTY so that you can SSH into the servers.

Compiling a Single File

Let's create a simple source file that just contains a hello world program. We will name the file "Main.java". You can create this file in a text editor like Notepad.

Main.java

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

Uploading Files by FTP

Now let's send the file up to the server using Filezilla. Run Filezilla and connect to Moxie using the following settings.

> **Host:** moxie.cs.oswego.edu
> **Username:** YOUR USERNAME
> **Password:** YOUR LAB PASSWORD

Once you have connected to the server, create a new directory on the server called "compiling".

Then find where you put the "Main.java" file on your computer and drag it into the new fold your created on the server.

SSHing Into the Server

Mac + Linux

Now you will need to SSH onto the server. If you are using Mac or Linux you can open up the terminal and run the following command.

```
ssh YOUR_USERNAME@pi.cs.oswego.edu
```

For example if your username is "jdoe42" then you would run the following command.

```
ssh jdoe42@pi.cs.oswego.edu
```

Windows

If you are using Windows, run PuTTY. Put in the following settings and click "Open" at the bottom of the window.

> **Host Name:** YOUR_USERNAME@pi.cs.oswego.edu

* If you are prompted about an RSA key type 'yes'

Compile + Run

Now that you have SSHed into the server you should enter the directory where you put the source file. You can enter a directory by using the "cd" command (change directory).

```
cd compiling
```

You can take a look at the files in the directory using the “ls” command (list files). You can also print out the contents of the source file to the screen using the “cat” command (concatenate).

```
ls
cat Main.java
```

Now to compile the program you will need to use the “javac” command (Java compile). To use “javac” you need to specify the file(s) that you want to compile, separated by spaces.

So to compile the “Main.java” file, run the following command.

```
javac Main.java
```

Now that you have compiled the source file, if you run “ls” again you will see that there is now a compiled “Main.class” file.

You can run the “Main.class” file using the “java” command and specifying the name of the class file to run, without the “.class” extension.

```
java Main
```

The program should then print out “Hello, World!”. If this is the case, then that means that it correctly compiled and ran.

Compiling a Multi-File Program

Now we will try to compile and run a Java program that contains multiple files.

First edit your “Main.java” to be the following:

Main.java

```
public class Main {
    public static void main(String[] argv) {
        Cube c = new Cube("blue");
        System.out.println(c);
    }
}
```

Then create a new file called “Cube.java” and put the following in:

Cube.java

```
public class Cube {
    private String color;

    public Cube(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return "Cube(" + this.color + ")";
    }
}
```

Now send those two files up to the server using Filezilla into the “compile” directory you created earlier. Feel free to overwrite the previous “Main.java” file.

Now in your SSH window you can compile the source code files by running “javac” and giving the names of both of the source files.

```
javac Main.java Cube.java
```

Now you can run the program by using the “java” command and giving the name of the Main class.

```
java Main
```

If the program prints out “Cube(blue)” then that means that it compiled and ran successfully.