# ***Goals***

# To introduce you to programming in Java.

# To use IntelliJ editor for writing, compiling, and executing Java programs.

# To learn how to use CodePost for submitting your work electronically.

# ***Background***

First, you must set up our Java programming environment on your computer. Follow these step-by-step instructions carefully:

* [Mac OS X](https://lift.cs.princeton.edu/java/mac/)
* [Windows 10](https://lift.cs.princeton.edu/java/windows/)
* [Linux](https://lift.cs.princeton.edu/java/linux/)

Second, all assignments require that you download an IntelliJ project zip file. Use the project folder from that you downloaded after your installed IntelliJ.

Third, read Sections 1.1 and 1.2 of the textbook. If you don't understand something, post an Ed question or attend office hours for assistance. Don't be bashful about asking for help.

# ***Implementation Tasks***

## **Overall Requirements**

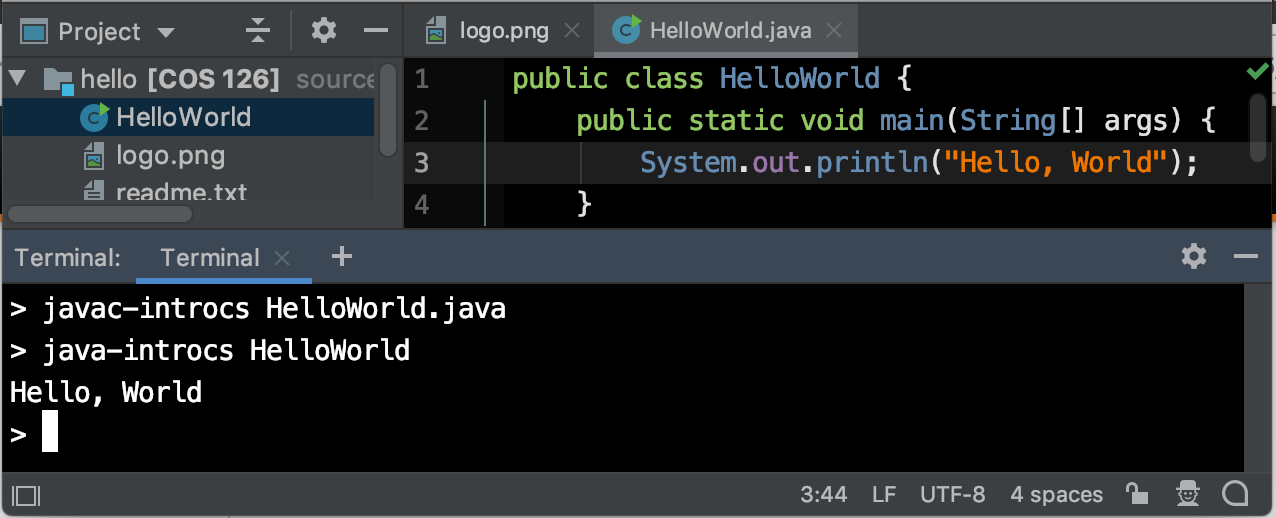
* You MUST implement five programs:
  + HelloWorld.java
  + HiFour.java
  + Ordered.java
  + GreatCircle.java
  + RGBtoCYMK.java
* You must submit a readme.txt

**HelloWorld.java**

Your first task is to write the HelloWorld program. In IntelliJ, select *File > New > Java Class*. Name the file HelloWorld then type in everyone's first program: [HelloWorld.java](https://www.cs.princeton.edu/courses/archive/fall20/cos126/precepts/p1-command-line/files/HelloWorld.java)

You will use the javac-introcs and java-introcs commands (instead of the default javac and java commands) to compile and execute your programs. These versions provide access to our course libraries.

In IntelliJ, select *LIFT > Open in Terminal.* Compile your program using the javac-introcs command and execute it using the java-introcs command:



| **FAQ**  ***I don't understand all of the jargon in HelloWorld.java. Should I drop the course?***  Don't worry—you'll discover the meaning of everything in the program over the course of the term. Java needs a lot of boilerplate code to get started. Do the readings and bring any lingering questions to precept. |
| --- |

## 

**HiFour.java**

This exercise demonstrates the use of the String data type and command-line arguments. Write a program HiFour.java that takes four first names as command-line arguments and prints a proper sentence with the names in the reverse of the order given.

Examples:

| > java-introcs HiFour Alice Bob Carol Dave  Hi Dave, Carol, Bob, and Alice.  > java-introcs HiFour Alejandro Bahati Chandra Deshi  Hi Deshi, Chandra, Bahati, and Alejandro. |
| --- |

| **FAQ**  ***When I run* HiFour*, I see the following message:***  **Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0 *What does this mean?*** Did you forget to type a command-line argument when you executed the program? |
| --- |

**Ordered.java**

This exercise demonstrates the use of the int and boolean data types. Write a program Ordered.java that takes three int command-line arguments, x, y, and z. Define a boolean variable whose value is true if the three values are either in strictly ascending order (x < y < z) or in strictly descending order (x > y > z), and false otherwise. You **must not** use any if statements.

Examples:

| > java-introcs Ordered 10 17 49  true  > java-introcs Ordered 49 17 10  true  > java-introcs Ordered 10 49 17  false |
| --- |

| **FAQ**  ***How do I get Java to print* true *or* false *without an* if-else *statement?*** If b is a boolean variable, then System.out.println(b) will print true or false, according to its value. |
| --- |

| ***Advanced Programming (Optional)***   * Implement Ordered.java without using the comparison operators (**<, <=, >, and >=**) and **if** statements. |
| --- |

**GreatCircle.java**

This exercise demonstrates the use of the double data type and Java’s [Math](https://introcs.cs.princeton.edu/java/11cheatsheet/#Math) library. The great circle distance is the shortest distance between two points on the surface of a sphere if you are constrained to travel along the surface. Write a program GreatCircle.java that takes four double command-line arguments x1, y1, x2, and y2 (the latitude and longitude, in ***degrees***, of two points on the surface of the earth) and prints the great circle distance (in nautical miles) between them. Use the following formula, which is derived from the spherical law of cosines:

This formula uses degrees, whereas Java's trigonometric functions use radians. Use Math.toRadians() and Math.toDegrees() to convert between the two. For reference, a nautical mile is 1/60 of a degree of an arc along a meridian of the Earth (which is approximately 1.151 miles).

Examples:

| > java-introcs GreatCircle 40.35 74.65 48.87 -2.33 // Princeton to Paris  3185.1779271158425 nautical miles  > java-introcs GreatCircle 48.87 -2.33 40.35 74.65 // Paris to Princeton  3185.1779271158425 nautical miles |
| --- |

| **FAQ**  ***My output for GreatCircle matches the sample output in the assignment specification, except in the very last digit or two. Why is there this tiny discrepancy? Is my answer wrong?*** Did you multiply by 60 (to convert the arc to nautical miles) before or after you converted the angle from radians to degrees? Computers work with limited precision, so algebraically equivalent formulas can produce slightly different answers. We typically ignore such tiny discrepancies when grading. |
| --- |

**RGBtoCMYK.java**

This exercise demonstrates the use of type conversions. Several different formats are used to represent color. For example, the primary format for LCD displays, digital cameras, and web pages — known as the *RGB format* — specifies the level of red (R), green (G), and blue (B) on an integer scale from 0 to 255. The primary format for publishing books and magazines — known as the *CMYK format* — specifies the level of cyan (C), magenta (M), yellow (Y), and black (K) on a real scale from 0.0 to 1.0.

Write a program RGBtoCMYK.java that converts from RGB format to CMYK format. Your program must take three integer command-line arguments red, green, and blue; print the RGB values; then print the equivalent CMYK values using these mathematical formulas.

You **may not** use if statements on this assignment, but you may assume that the command-line arguments are not all simultaneously zero.

Recall that Math.max(x, y) returns the maximum of x and y.

For full credit, your programs must not only work correctly for all valid inputs, but they should be easy to read. You are expected to follow these [style guidelines](https://www.cs.princeton.edu/courses/archive/fall20/cos126/assignments/faq/index.html#style) in this course.

Examples:

| | > java-introcs RGBtoCMYK 75 0 130  red = 75  green = 0  blue = 130  cyan = 0.423076923076923  magenta = 1.0  yellow = 0.0  black = 0.4901960784313726 | >java-introcs RGBtoCMYK 255 143 0  red = 255  green = 143  blue = 0  cyan = 0.0  magenta = 0.4392156862745098  yellow = 1.0  black = 0.0 | | --- | --- | | indigo | Princeton orange | |
| --- | --- | --- | --- | --- |

| ***Advanced Programming (Optional)***   * Implement RGBtoCMYK.java without using Math.max() and **if** statements. |
| --- |

**readme.txt**

Edit the text file named readme.txt that is a narrative description of your work. Each week, we provide a readme.txt file for you to download and use as a template, answering all questions in the space provided. Submit this file with your .java files.

# ***Submitting the assignment***

Submit HelloWorld.java, HiFour.java, Ordered.java, GreatCircle.java, RGBtoCMYK.java, and readme.txt via the Web submission system called codePost. Join codePost to submit your programming projects - <https://codepost.io/signup/join?code=F8CL1DHNIC>

# ***Enrichment***

## **Hello World in 200 Languages**

Here is [Hello World](http://helloworldcollection.de) in over 200 different programming languages.