Objectives.

1. Prepare for the course: sign up for Top Hat, Piazza, and Gradescope; apply for a CS account; setup the programming environment and get familiar with it.
2. Write simple straight-line Java programs.

Problem 1. (*Preparing for the Course*) Take care of the following action items (see course website for details):

- Sign up for Top Hat using your UMass Boston email.
- Sign up for Piazza using your UMass Boston email.
- Sign up for Gradescope using your UMass Boston email.
- Apply for a CS account.
- Setup the programming environment and familiarize yourself with it.

Edit the Java program `CoursePrep.java` by replacing the placeholders `[Name]`, `[UMass Boston Email]`, and `[CS Account Username]` with relevant information and test the program by running the following command on the terminal:

```
$ javac CoursePrep.java
$ java CoursePrep
I acknowledge that I have fully read and understood the contents of the course
website. I understand that if I have any questions or concerns about the course
mechanics, it is my responsibility to discuss them with the instructor.

Jane Doe
jane.doe@.umb.edu
jdoe
```

Problem 2. (*Wind Chill*) Given the temperature t (in Fahrenheit) and the wind speed v (in miles per hour), the National Weather Service defines the effective temperature (the wind chill) to be

$$w = 35.74 + 0.6215t + (0.4275t - 35.75)v^{0.16}.$$

Write a program `WindChill.java` that takes two doubles t and v as command-line arguments and writes the wind chill.

```
$ javac WindChill.java
$ java WindChill 32 15
21.588988890532022
```

Problem 3. (*Day of the Week*) Write a program `DayOfWeek.java` that takes three integers m (for month), d (for day), and y (for year) as command-line arguments and writes the day of the week (0 for Sunday, 1 for Monday, and so on) $\mathcal{D}$, calculated as follows:

$$\begin{aligned} y_0 &= y - (14 - m)/12 \\ x_0 &= y_0 + y_0/4 - y_0/100 + y_0/400 \\ m_0 &= m + 12 \times ((14 - m)/12) - 2 \\ \mathcal{D} &= (d + x_0 + 31 \times m_0/12) \bmod 7 \end{aligned}$$

```
$ javac DayOfWeek.java
$ java DayOfWeek 3 14 1879
5
```

Problem 4. (*Great Circle*) Write a program `GreatCircle.java` that takes four doubles x_1, y_1, x_2, and y_2 representing the latitude and longitude in degrees of two points on earth as command-line arguments and writes the great-circle distance (in km) between them, given by the equation:

$$d = 111 \arccos(\sin(x_1)\sin(x_2) + \cos(x_1)\cos(x_2)\cos(y_1 - y_2)).$$

Note that this equation uses degrees, whereas Java's trigonometric functions use radians. Use `Math.toRadians()` and `Math.toDegrees()` to convert between the two. Use your program to compute the great-circle distance between Paris (48.87° N and 2.33° W) and San Francisco (37.8° N and 122.4° W).

```
$ javac GreatCircle.java
$ java GreatCircle 48.87 -2.33 37.8 -122.4
8701.389543238289
```

Problem 5. (*Three Sort*) Write a program `ThreeSort.java` that takes three integers as command-line arguments and writes them in ascending order, separated by spaces. Use `Math.min()` and `Math.max()`.

```
$ javac ThreeSort.java
$ java ThreeSort 1 2 3
1 2 3
$ java ThreeSort 1 3 2
1 2 3
$ java ThreeSort 2 1 3
1 2 3
$ java ThreeSort 2 3 1
1 2 3
$ java ThreeSort 3 1 2
1 2 3
$ java ThreeSort 3 2 1
1 2 3
```

Problem 6. (*Three Dice*) Write a program `ThreeDice.java` that writes the sum of three random integers between 1 and 6, such as you might get when rolling three dice.

```
$ javac ThreeDice.java
$ java ThreeDice
5
```

Files to Submit

1. `CoursePrep.java`
2. `WindChill.java`
3. `DayOfWeek.java`
4. `GreatCircle.java`
5. `ThreeSort.java`
6. `ThreeDice.java`

Before you submit:

- Make sure your programs meet the input and output specifications by running the following command on the terminal:

```
$ python3 run_tests.py -v [<problems>]
```

  where the optional argument `<problems>` lists the problems (`Problem1`, `Problem2`, etc.) you want to test, separated by spaces; all the problems are tested if no argument is given.

- Make sure your programs meet the style requirements by running the following command on the terminal:

```
$ check_style <program>
```

  where `<program>` is the `.java` file whose style you want to check.