**Name:**

**Advanced Programming in Java**

**Lab Exercise 9/23/2020**

In this exercise, you will explore various methods available in the Math class.

1. Read Lesson 6 in Blue Pelican.
2. Light travels at 3 \* 108 meters per second. A light year is the distance a beam would travel in one year. Write a program that calculates and displays the value in meters of a light year.
3. Write a program that allows the user to enter the length and width of a rectangle and prints out the diagonal distance from corner to corner.
4. **Great circle.** Write a program GreatCircle.java that takes four values; x1, y1, x2, and y2 (the latitude and longitude, in degrees, of two points on the earth) and prints out the great-circle distance between them. The great-circle distance d (in nautical miles) is given by the formula derived from the law of cosines:



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, -2.33° W) and San Francisco (37.8° N, 122.4° W). You should get 8947 kilometers.

Note: the shape of the earth is more like a flattened spheroid than a sphere, so the formula above is only an approximation (up to around 0.5% error). Also, this formula is unreliable for small distances because the inverse cosine function is ill-conditioned.

For a more accurate calculation, here is the Haversine formula:

|  |  |
| --- | --- |
|  | a = sin²(Δφ/2) + cos φ1 ⋅ cos φ2 ⋅ sin²(Δλ/2) c = 2 ⋅asin  d = R ⋅ c |
|  | φ*is latitude,*λ*is longitude,*Δ signifies delta, R*is earth’s radius (mean radius = 6,371km or 3956 miles);*  Δφ and Δλ signify the difference in latitude and longitude respectively.  *note that angles need to be in radians to pass to trig functions!* |

**Note: The distance from San Francisco to Paris is 5547.69 miles or 8947.93 kilometers.**

**When you have completed your programs, submit your documented source code.**