Welcome to CS210 Discussion

A Quick intro before we get into it

About Me

- Senior CS Major (Graduating in August or December)
- Took variation of this course with Professor Iyer in Spring 2020
- I enjoy music, video games, and the outdoors
- Went to UMass Lowell for 3 semesters, then transferred here to UMB
- Favorite Languages: C and Java
- Least Favorite Languages: Javascript, Python, Scheme/Lisp

For this course I am responsible for Discussions, Answering Piazza Questions, and Holding Office Hours

Contact Info

Email: Kyle.hackett001@umb.edu

Website: https://kylehackett99.github.io/cs210

Post your Questions on Piazza, helps you and other students

Posting Options:

- Concept Question
- Anonymous Concept Question
- Private Question (Grading, Personal, Code Snippets)

My Office Hours

Monday and Wednesday: TBA @ TBA (Sometime between 12 and 3:50pm)

Friday (by appointment): 12:30pm - 2:30pm @ Zoom - link will be on piazza

Please make appointment before 12:30pm

What you'll need for Discussion

Pen and Paper or Laptop

Yourself

Questions!

What you should expect in Discussion

- Week 1 after Project Release (Week 1 of project cycle)
  - Exercises and Concept review
- Week 2 after Project Release (Week 2 of project cycle)
  - Project Review
- Week Before exam will be exam review
  - Will review concepts and questions for exam
- Exceptions for the Weekly plan apply when there is overlap between projects and exams, in those cases you will be responsible to the exercises
- Exceptions will also apply in cases where Professor deviates from Professor Iyer's course
- Come with Questions!

What I expect in Discussion

- AT LEAST Read the project pdf before class 1 of a project cycle
- Also good to have reviewed the code itself before
- AT LEAST start the project problems before class 2 of a project cycle
    - Try your best and don't stress it, Discussion is for YOU and your questions
- Attendance is not mandatory but attending will do nothing but help ;)

Coding Environment

You should have a coding environment going by now

You should definitely try using the native set up as it is much better than using the VM

Having the Coding Environment set up is important for practice so make sure you get that going as soon as possible

If it is something many students are struggling with we can discuss it toward the end of this session

Resources

- SI Sessions, Office Hours, Discussion, Piazza
- Course page: https://www.cs.umb.edu/~cyrus/cs210/cs210.html#
- Prof Iyer's Page: https://www.swamiiyer.net/cs210/
- Discussion Slides + resources: https://kylehackett99.github.io/cs210
- Java Brushup/Learning
  - www.Sololearn.com
  - https://web.stanford.edu/class/archive/cs/cs108/cs108.1082/JavaBasicRefresher.pdf

Sample Report (NOTE: I will not be grading the reports so this is just a ballpark idea, as Bang will likely not have exactly the same expectations, but this will at least help in your understanding):

https://kylehackett99.github.io/cs210/sample-report.txt

Python VS Java (110 vs 210)

https://www.geeksforgeeks.org/difference-between-python-and-java/

- Read if you're interested in reading the differences between the two in application

```python
# Declaring variables
x, y = 12, 10
isTrue = True
greeting = "Welcome!"
```

```java
public class Main {
    public static void main(String[] args) {
        // Declaring variables
        int x = 12, y = 10;
        boolean isTrue = true;
        String greeting = "Welcome!";
    }
}
```

Open for Questions!

Compile and run

Compile (Single Class): Javac -d /out /src/xyz.java

Compile (Multi Class): Javac -d /out /src/xyz.java /src/abc.java

Run: Java xyz <parameters>

Inheritance

- Inheritance allows for a class to also carry the methods, instance variables, etc of its super class (class that its inheriting attributes from)
- For example, you have a Flushed out Person class, and you are building a new class for a mythical creature that has attributes of a Person, but with a twist
    - Extend your new mythical creatures class with Person to also take on Person's methods and instance variables
- A Java class can only inherit ONE class!

Why And When To Use "Inheritance"?

- It is useful for code reusability: reuse attributes and methods of an existing class when you create a new class.

https://www.w3schools.com/java/java_inheritance.asp

Interface

- Another form of abstraction in java
- Provides empty methods to a class that need to be implemented by the developer
- Allows for multiple interfaces, can be separated by comma
- Like Inherited classes, they cannot be turned into objects by themselves
    - Ex. You can have an Animal, but not Mammal (per my example)

Good for wanting classes to have the same methods but to be different for each class

Generic Methods

Generic Methods are able to be used with any data type (provided the logic inside the method is not type dependant)

All generics type parameter is < X > (with x being what you want it to be)

Generic Classes

A Class that can be any data type

Project 1 Exercises

https://www.swamiiyer.net/cs210/project1.pdf

Exercise 1. (*Great Circle Distance*) Write a program called `GreatCircle.java` that accepts x_1 (double), y_1 (double), x_2 (double), and y_2 (double) as command-line arguments representing the latitude and longitude (in degrees) of two points on earth, and writes to standard output the great-circle distance (in km) between the two points, given by the formula

$$d = 6359.83 \arccos(\sin(x_1)\sin(x_2) + \cos(x_1)\cos(x_2)\cos(y_1 - y_2)).$$

```
>_ ~/workspace/project1
$ java GreatCircle 48.87 -2.33 37.8 -122.4
8701.387455462233
```

Exercise 2. (*Counting Primes*) Implement the static method `isPrime()` in `PrimeCounter.java` that accepts an integer x and returns `true` if x is prime and `false` otherwise. Also implement the static method `primes()` that accepts an integer n and returns the number of primes less than or equal to n — a number x is prime if it is not divisible by any number $i \in [2, \sqrt{x}]$.

```
>_ ~/workspace/project1
$ java PrimeCounter 1000
168
```

Exercise 3. (*Euclidean Distance*) Implement the static method `distance()` in `Distance.java` that accepts position vectors x and y — each represented as a 1D array of doubles — and returns the Euclidean distance between the two vectors, calculated as the square root of the sums of the squares of the differences between the corresponding entries.

```
>_ ~/workspace/project1
$ java Distance
5
-9 1 10 -1 1
5
-5 9 6 7 4
13.0
```

Exercise 4. (*Matrix Transpose*) Implement the static method `tranpose()` in `Transpose.java` that accepts a matrix x — represented as a 2D array of doubles — and returns a new matrix that is the transpose of x.

```
>_ ~/workspace/project1
$ Transpose
3 3
1 2 3
4 5 6
7 8 9
3 3
   1.00000    4.00000    7.00000
   2.00000    5.00000    8.00000
   3.00000    6.00000    9.00000
```

Exercise 5. (*Rational Number*) Implement an immutable data type called `Rational` that represents a rational number, ie, a number of the form a/b where a and $b \neq 0$ are integers. The data type must support the following API:

Rational	
`Rational(long x)`	constructs a rational number whose numerator is `x` and denominator is 1
`Rational(long x, long y)`	constructs a rational number given its numerator `x` and denominator `y` (†)
`Rational add(Rational other)`	returns the sum of this rational number and `other`
`Rational multiply(Rational other)`	returns the product of this rational number and `other`
`boolean equals(Object other)`	returns `true` if this rational number is equal to `other`, and false otherwise
`String toString()`	returns a string representation of this rational number

† Use the private method `gcd()` to ensure that the numerator and denominator never have any common factors. For example, the rational number 2/4 must be represented as 1/2.

```
>_ ~/workspace/project1
$ java Rational 10
a           = 1 + 1/2 + 1/4 + ... + 1/2^10 = 1023/512
b           = (2^10 - 1) / 2^(10 - 1) = 1023/512
a.equals(b) = true
```

Exercise 6. (*Harmonic Number*) Write a program called `Harmonic.java` that accepts n (int) as command-line argument, computes the nth harmonic number H_n as a rational number, and writes the value to standard output.

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n-1} + \frac{1}{n}.$$

```
>_ ~/workspace/project1
$ java Harmonic 5
137/60
```