Lecture 1

Intro

Some slides were borrowed from Josh Hug and Adam Jundt

Reminders

- Mic

Plan for today

- Introduction
- Why JAVA??
- Class Components (very brief)
- Start an actual lecture

About me

- Marina Langlois: Marina, Dr. Langlois, Mrs. Langlois

- **Before UCSD:**
  - PhD from UIC (Chicago)
  - Lecturer at UIC
  - Lecturer at Yeshiva (NYC)

- **At UCSD:**
  - CSE 8A, 12, 20, 150
  - DSC 10, 20, 30, 80 and 95

- **Not working:**
  - Having fun with my family

Class overview

Class Overview

- Introduction to the **basic** data structures used in computer science:
  - Understand each data structure in detail
  - Analyze the algorithms that use them
  - Know limitations of each data structure
- Practice designing and applying data structures to real world problems through coding.
- Learn *some* Java as a programming language.

Why Java for Data Science?

- Java is fast
  - Great for developing back end of machine learning algos
- Has good selection of libraries:
  - For ML
- Understand someone else's code
- Understand terminology
- Resume looks better
- For this class in particular

I am scared of Java

What do you think about Java now?

I knew it already	13 respondents	**18 %**
got used to it, it is not very scary.	36 respondents	49 %
Not very comfortable, but not too bad	17 respondents	23 %
Still feel lost often	5 respondents	7 %
Something else	3 respondents	4 %

What do you think about Java now?

I knew it already	6 respondents	**8 %**
got used to it, it is not very scary.	51 respondents	64 %
Not very comfortable, but not too bad	18 respondents	23 %
Still feel lost often	4 respondents	5 %
Something else	1 respondent	1 %

In particular...

- **Efficiency**
  - Good algorithms.
  - Good data structures.
- **Programming(efficiently)**
  - Designing, building, testing, and debugging large programs.
  - Use of programming tools.
    - git, IntelliJ and JUnit
  - Java (not the main focus of this class)

Why study algorithms and Data Structures?

- To find a job.
- To keep a job.

Grade components:

- Reading quizzes: 10%
- **Participation in class: 2% EXTRA CREDIT**
- Discussions: 4%
- Programming Assignments: 36%
- Two midterm exams: 20%
- Final Exam: 30%

Reading quizzes: 10%

- You will have a reading assignment for **each** class
  - Check the schedule (website: DSC30.org)
- Reading quizzes will be assigned for **each** class
  - Either on Canvas (3 attempts to complete it. Take the max score).
  - Or Zybooks. It has “activities” => will be considered as reading quizzes (demo)
- **Ideally** be completed before each class **(Deadline is Friday for all)**

Zybooks policies

- If you decide to drop the class within a few weeks, zybooks gives your money back: Link
- If I use it again and you have to retake the class it is either free or huge discount: Link

Participation in class: 2% LUMVJP

- I ask questions, you answer them :) using webclicker (like last quarter)
- At least **70%** of the questions to count.
- Make sure to use your UCSD email address (i.e., @ucsd.edu) when creating an account. Only use lowercase when entering your email address.
- Your password should have a length of at least 6
- For student identifier, put in your PID
- Make sure you are on the **guest/public** wifi for it to work.
- Make sure your geolocation is on.

https://webclicker.web.app/

Discussions: 4%

The purpose of discussions is to prepare you for taking exams on paper.

Process:

- You will be given a few problems emulating exam questions and a limited time.
- After the time is up, you will give your work to the tutor and then he/she will go over the solutions with you.
- In order to record your participation, you will the weblicker app, so please bring your phones

Notes:

- 1 lowest discussion will be dropped.
- If your exam score is above 90%, then 3 more discussions will be dropped.

Programming Assignments (PAs): 36%

- You will have ~one assignment per week (sometimes longer). Including weeks with midterms.
- Write-up will on the website, I will announce it on Ed Discussion when published.
- **After each PA I will have a feedback form: 3 points for you PA (not EC)**

Assignments. Late submissions

- You will have one assignment per week.
- Write-up will be on the website

- **5 slip days**: you can submit late within 24 hours without penalty.
- Redeem: you can re-submit your HWs and get up 70% points back after the grades are released (1 week, only autograded part).

* Unless something exceptional had happened. Documentation will be required in some cases.

Exams: 50% in total

- Midterms are during the class period
- You can’t skip/miss midterm unless you have a good reason.
  - Unless something exceptional had happened then you can make it up.
- Final exam:
  - Gives you opportunity to replace your midterm (s) grade with the higher one.

Collaboration

Asking questions is highly encouraged

- *Discuss* all questions with each other (except exams and checkpoint quizzes)
- Submit homeworks **individually (or only with the partner when allowed)**, but feel free to discuss with others (general approach, the best data structure et)
- **Discussion: does NOT answer the question “HOW?”**

Collaboration

Asking questions is highly encouraged

- *Discuss* all questions with each other (except exam checkpoint quizzes)
- Submit homeworks **individually (or only with the pa allowed)**, but feel free to discuss with others (general approach, the best data structure et)

The limits of collaboration

- **Don't share solutions** with each other or **look at** someone's code
  - Including Google
- **Academic integrity violations will result in failing the course**

ChatGPT: is it allowed?

Short Answer: No but Yes :)

NO: You can't use it to solve the PAS. Remember the purpose of this class: practice as much as possible.

YES: One of the great advantage I see is the ability to generate questions that suit your needs.

- Ask chatGPT to generate a problem without a solution.
- Try to solve it.
- Then ask for a solution and compare or ask for explanation. Be careful: it is wrong sometimes!

Bottom Line: Do not use PAS questions since you have nothing left to practice on for real!

Not quite alone

- Team of tutors (their zoom/lab hours will be on the calendar).

Questions?

First Java Program "Hello World"

Wiki

REMINDER

- mic

Basic rules -1: Demo

1. All code in Java must be part of **a class**
2. For code to run you need to have

```
public static void main(String[] args)
```

1. We mark the *beginning* and *end* of segments of code using

   { and }
2. All statements in Java must end in a semi-colon: ;

Print all numbers from 1 to 5

```
#print numbers from 1 to 5

x = 1

while x <= 5:

    print(x)

    x  = x + 1
```

Basic rules - 2

1. Before Java variables can be used, they must be ***declared***
2. Java variable must have a specific type:
    a. int, String, double, boolean etc
3. Types can *never* change
4. Types are verified *before* the code even runs
    a. Big difference between Python and Java

Discussion Question - 1

How many errors can you find in the code on the right?

A: 1
B: 2
C: 3
D: 4
E: 5 or more

```
public class Discussion {
    public static void main(String[] args) {

        double y = 5.6
        x = 10;

        if (x < y):
            System.out.println(y is smaller);

        else {
            x = x * y;
            System.out.println(y);
        }

}
```

Functions

```
# defining functions
def smaller(x, y):
    """ Returns smaller of the two """

    if (x < y):
        return x
    return y

 print(smaller(3, 4))
```

Defining Functions. Basic Rules

1. Functions must be declared as part of a class in Java
    a. A function that is inside a class is called a “method”
    b. All functions in Java are methods
2. To define a function in Java we use “public static”
    a. Other ways are later
3. All parameters must have a declared type
4. Return value of the function must have a declared type
5. Functions in Java return only one value

Discussion Question

How many errors can you find?

A: 1
B: 2
C: 3
D: 4
E: 5 or more

```
public class Discussion {
    public static String discussion(int x, str){
        while (x<str.length()){
            x = x + 2;
        }
        if (x % 2 == 0){
            return "x is even";
        }
        else
        {
            return x+1;
        }
    }
    public static void main(String[] args) {
        System.out.println(discussion(5.5, "first class"));
    }
}
```

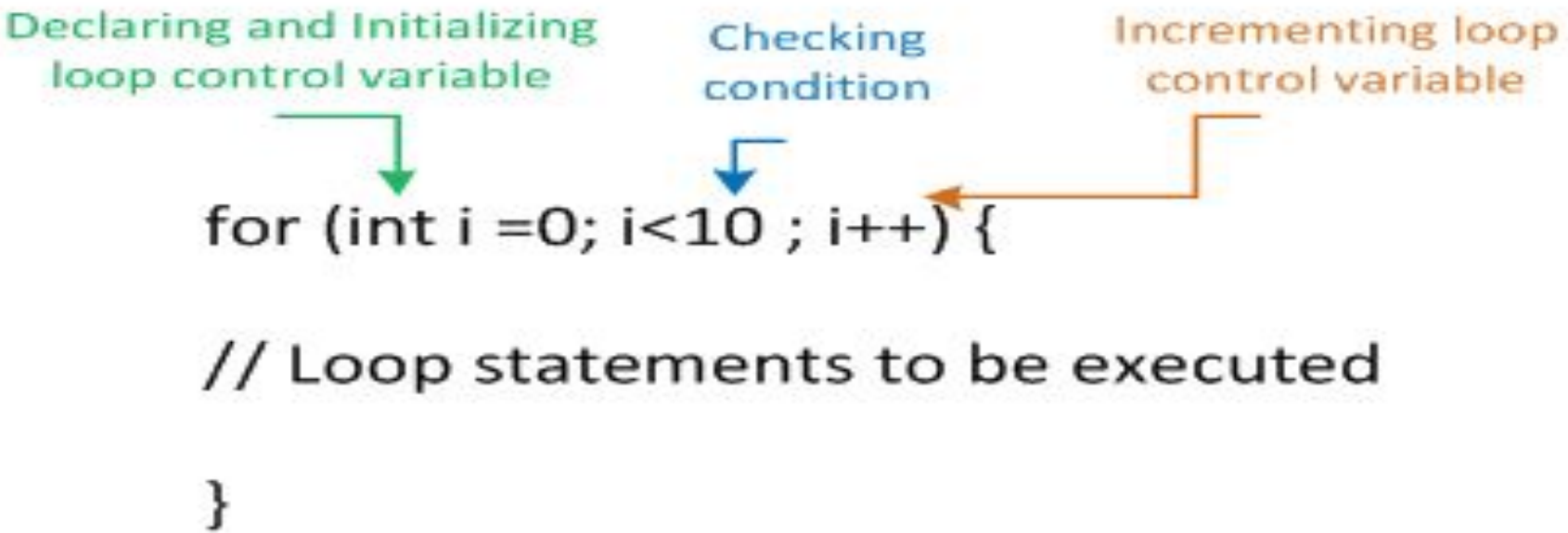
Documenting your code with Javadoc

```python
def smaller(x, y):
    """ Returns smaller of the two """

    if (x < y):
        return x
    return y
```

```java
class SmallerFunction {
    /** Returns the smaller of the two */
    public static int smaller(int x, int y) {
        if (x < y){
            return x;
        }
        return y;
    }
```

For loop in java



For loop in java

What is the output? *

A: 1, 3, 5, 7, 9

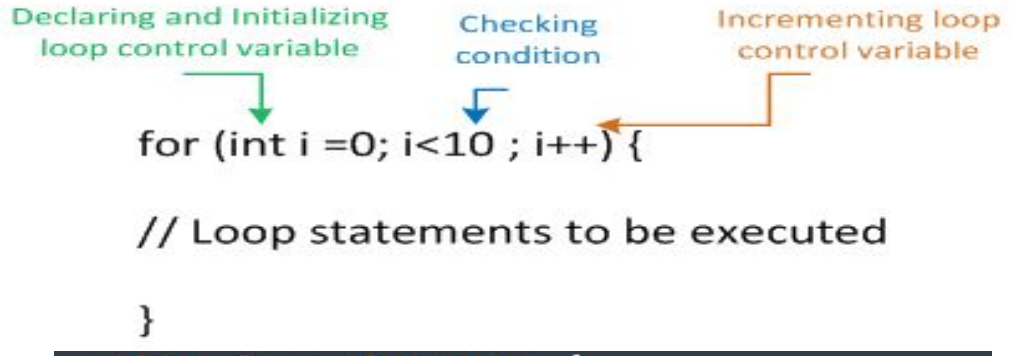
B: 1, 3, 5, 7, 9, 11

C: 1, 4, 7, 10

D: 1, 4, 7

E: None of the above

** Assume that the output does not have commas and each number is on a new line.*




```java
public class Discussion {

    public static void main(String[] args) {

        for (int i=1; i<10; i=i+2){
            System.out.println(i);
            i = i + 1;
        }
    }
}
```

Main difference

- You can modify “i” inside the loop and the change will be reflected in the header as well.

Reflections on Static Typing

The Good:

-

The Bad:

-

Talk to each other about pros and cons

Reflections on Static Typing

The Good:

- Debugging is a lot easier, type errors are avoided.
- Code on the user side has no type errors, which means phones won't crash because of type errors.
- Programs run more efficiently in time and memory.
- Self-documenting: YOU KNOW WHAT YOU'VE GOT.

The Bad:

- Code is more verbose.
- Code is less general.There is a way around this in Java (Generics)

Short practice

Write a function *expand* that takes an integer and returns an integer array with numbers 1, 2.. up to (including) the parameter:

Example:
 Input: 5
 Output: [1, 2, 3, 4, 5]