CMSC201
Computer Science I for Majors

Lecture 01 – Introduction

Introductions

- Dr. Katherine Gibson
  - Education
    - BS in Computer Science, UMBC
    - MS & PhD in CS, University of Pennsylvania
  - Likes
    - Video games
    - Dogs
    - Nail polish

Introductions

- Dr. Krystle Wilson
  - Education
    - MS, PhD in Computer Science

      Mississippi State University
  - Likes
    - Teen Titans Go!
    - Sports

Course Overview

Course Information

- First course in the CMSC intro sequence
  - Followed by CMSC 202
- CMCS majors must get a B or better
- CMPE majors must get a B or better
  - Unless you entered UMBC prior to Fall 2016
- No prior programming experience needed
  - Some may have it

What the Course is About

- Introduction to Computer Science
  - Problem solving and computer programming
- We're going to come up with algorithmic solutions to problems
  - What is an algorithm?
- We will communicate our algorithms to computers using the Python language

Class Objectives

- By the end of this class, you will be able to:
  - Use an algorithmic approach to solve computational problems
  - Break down complex problems into simpler ones
  - Write and debug programs in the Python programming language
  - Be comfortable with the UNIX environment

Why Learn to Program?

- Programming skills are useful across a wide range of fields and applications
  - Many scientific professions utilize programming
  - Programming skills allow you to understand and exploit “big data”
  - Logical thinking learned from programming transfers to many other domains

Grading Scheme

- This class has:
  - 6 Homeworks (40 points each)
    - Small programming assignments
  - 3 Projects (80 points each)
    - Larger programming assignments
  - 10 lab assignments (10 points each)
  - 4 mandatory surveys (5 points each)
  - A midterm (200 points)
  - A comprehensive final exam (200 points)

A Note on Labs

- Your “discussion” section is actually a lab
  - In the Engineer building (ENG)
- Labs are worth 10% of your grade
- You must attend your **assigned** section
  - No credit for attending other sections

Submission and Late Policy

- Homeworks and projects will be submitted over the GL server with the **`submit`** command

- Homeworks will always be due at 8:59:59 pm
- Late homeworks will receive a ***zero***
- (In other words, there are no late homeworks)

Submission and Late Policy

- It is not recommended that you submit close to the deadline
  - Sometimes the server gets overloaded with everyone trying to submit
  - Developing programs can be tricky and unpredictable
- Start early and submit early (and often!)

Academic Integrity

Academic Integrity

- We have homeworks and projects in this class
- You should never, *ever,* ***ever*** submit work done by someone else as your own
- If you submit someone else's code, both students will get a 0 on the assignment

Things to Avoid

- Downloading or obtaining anyone else's work
- Copying and pasting another person's code
- Leaving your computer logged in where another student can access it
- Giving your code to another student
  - Or explaining it in explicit detail to another student
- Attempting to buy code online
  - This will result in an immediate F in the class

Things that are Always Okay

- And encouraged!

- Talking to a classmate about a concept
- Getting help from a TA or instructor
- Comparing program output
- Discussing how to test your program
- Working on practice problems together

Collaboration Policy

- We want you to learn all these things:
  - The course material
  - How to work independently
  - How to work collaboratively
- Some assignments will be “individual work” while others will be “collaboration allowed”
  - These will be clearly marked on each assignment
  - You may only collaborate with current 201 students

What Is Allowed?

Action	Allowed for Individual Work	Allowed when Collaborating
Getting help from an instructor or TA	**Allowed**	**Allowed**
Brainstorming general solutions to the assignment		
Creating, sharing, or copying course notes		
Purchasing solutions		
Borrowing verbatim from the course slides or book		
Giving (or receiving) a detailed explanation		
Looking for solutions or help online		
Looking at someone else's code		

What Is Allowed?

Action	Allowed for Individual Work	Allowed when Collaborating
Getting help from an instructor or TA	**Allowed**	**Allowed**
Brainstorming general solutions to the assignment	Not Allowed	**Allowed**
Creating, sharing, or copying course notes	**Allowed**	**Allowed**
Purchasing solutions	Not Allowed	Not Allowed
Borrowing verbatim from the course slides or book	**Allowed**	**Allowed**
Giving (or receiving) a detailed explanation	Not Allowed	Not Allowed
Looking for solutions or help online	Not Allowed	Not Allowed
Looking at someone else's code	**It Depends**	**It Depends**

What Is Allowed?

Action	Allowed for Individual Work	Allowed when Collaborating
Getting help from an instructor or TA	**Allowed**	**Allowed**
Looking at someone else's code	**It Depends**	**It Depends**

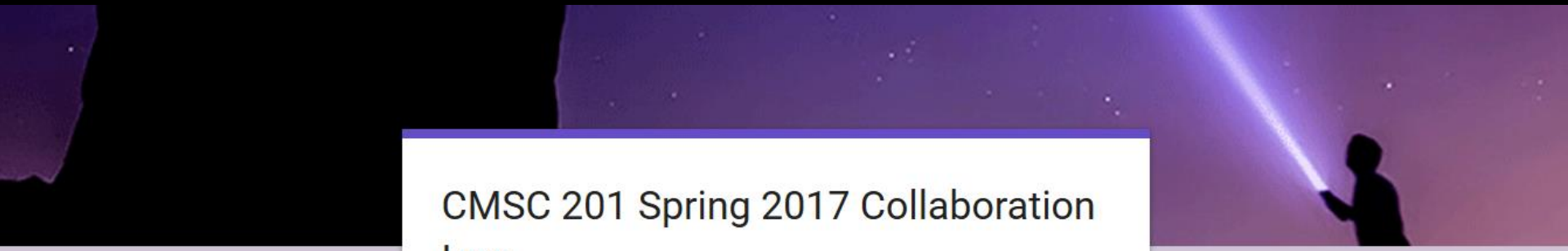
You may never look at someone else's code without their permission
You may never look at someone else's code on your computer

When collaborating, you may look at someone else's code on their screen and with their permission
When working individually, you may not look at anyone else's code

Acknowledging Collaboration

- If you work with another student, you must fill out the Collaboration Log
  - Other student's name and email
  - What you discussed
- Even if you only gave help

- Needs to be done within 24 hours
  - Do it as soon as you're done collaborating



CMSC 201 Spring 2017 Collaboration Log

Use this form to log every instance of working with another CMSC 201 student on an assignment.

Your email address (**k38@umbc.edu**) will be recorded when you submit this form. Not you? Sign out

* Required

Collaborator Name *
Include their first AND last name

Your answer

Collaborator's Email *

Your answer

Who helped whom? *

- ◯ Other student helped me
- ◯ I helped other student
- ◯ We both helped each other

Why So Much About Cheating?

- Every semester, a large number of students get caught for sharing code
  - They're often friends
  - One student is trying to help the other out
  - Or two students working to solve one problem
- They both endanger their entire academic career when they get caught
  - And the friend gets a zero for helping them

Alternatives to Cheating

- Turn in a partially done assignment
  - Still get partial points
  - (Better than a zero for cheating)
- Discuss concepts with other students, but not assignment details
- Come get help in office hours!

Becoming a Good Programmer

- We are strict about academic integrity because we want everyone to succeed in this class

- Understanding the assignment solutions means you will do better on the exams
- Learning the course material means you will do better in your future courses and career
- Seeking help when you need it will help you grow as a student and as a computer scientist

Getting Help

Where to Go for Help

- There are a number of places you can go if you are struggling!
  - All of the TAs happy to help
  - If the TAs aren't working out, come by the instructors' office hours (this should not be your first resort for help)
- Office hours will be posted on the website
  - Nearly 40 hours of office hours each week

CMSC 201 TAs

- You are welcome to go to **any** TA for help
  - If you need help with an assignment
  - If you have a question about an assignment
- Final schedule posted on the website
- Over 25 hours each week where a TA is available in ITE 240
  - ITE 240 will be ***busy*** on the due date

ITE 240

- This is a computer lab in the ITE building used to hold 201, 202, and 341 office hours
- The 201 TAs will…
  - Be wearing bright yellow lanyards
  - Have their names on the whiteboard in the front
- Sign in using a google form, to ensure everyone is helped in a timely manner

CMSC 201 Office Hour Sign In Spring 2017

Your email address (**k38@umbc.edu**) will be recorded when you submit this form. Not you? Sign out

* Required

What do you need help with today? *
For example: "HW2", "submitting", or "loops"

Your answer

What is the exact question you would like answered? *
Good: "My program is exiting before it should", "I don't understand how to use Boolean logic" .....
Bad: "My program doesn't work", "I don't know what to do next", "This assignment is confusing"

Your answer

How long have you, personally, already spent working on this problem or question? *

- ○ < 5 minutes
- ○ 6 - 15 minutes
- ○ 16 - 30 minutes

Additional Help

- Tutoring from the Learning Resources Center
  - By appointment
- Computer help from DoIT
  - By phone or in person
- See the syllabus for more info

Announcement: Note Taker Needed

A peer note taker has been requested for this class. A peer note taker is a volunteer student who provides a copy of his or her notes for each class session to another member of the class who has been deemed eligible for this service based on a disability. Peer note takers will be paid a stipend for their service.

Peer note taking is not a part time job but rather a volunteer service for which enrolled students can earn a stipend for sharing the notes they are already taking for themselves.

If you are interested in serving in this important role, please fill out a note taker application on the Student Disability Services website or in person in the SDS office in Math/Psychology 212.

Programming Mindset

Taking Time

- For every credit hour, you should spend at least 1 - 4 hours studying each week
  - For CMSC 201, that means 4 - 16 hours
- Amount of time spent depends on assignment load and course difficulty
- You won't pass this class by spending an hour a week on the material and assignments

Time Spent In Class

- In class, we'll mostly focus on
  - Concepts
  - Ways of thinking
  - Common mistakes
  - More concepts
- We'll only spend a small amount of time on
  - Writing programs
  - Actual coding

Time Spent Out of Class

- Learning to code and think like a programmer is like learning any new skill
  - You only get better if you ***practice*** a lot!

- Assignments are designed to be practice for the skills you need
  - Spend the time to really understand them!
  - Experiment! ("What happens if I do …?")

"Failure"

- No one gets everything right on the first try
  - Especially in programming
- Everyone makes mistakes when coding
  - Including the professors
    - You'll see me do it almost every time we code in class
  - Including the TAs
  - Including you

Making Mistakes

- A mistake is not a failure!
- Don't give up after one error or setback
  - Learn from your mistakes, and get better
- Don't underestimate yourself
  - You're learning an entirely new skill set, it would be weird if you "got it" right off the bat

Programming at UMBC

UMBC Computing Environment

- We develop our programs on UMBC's GL system
  - GL is running the Linux Operating System
    - GUI – Graphical User Interface
    - CLI – Command-Line Interface
- Lab 1 will walk you through using the UMBC computing environment

How Do I Connect to GL?

- Windows
  - Download Putty (Lab 1 has a video about this)
  - Hostname:

    `gl.umbc.edu`
  - Make sure you pick "SSH"
  - Put in username and password

- Mac
  - SSH client is already installed
  - Go to the Application folder and select Utilities
  - Open up a terminal window
  - Enter the following:

    `ssh -l username gl.umbc.edu`
  - Put in your password

You won't see any asterisks appear when you type in your password, but it is working!

Linux Commands

- See: http://www.csee.umbc.edu/resources/computer-science-help-center/#Resources

- Here's a few basic commands:

  `ls` – list contents

  – List files and directories in your current directory

  – Directory is just another word for folder

More Basic Commands

- **Important!!** Commands are case sensitive

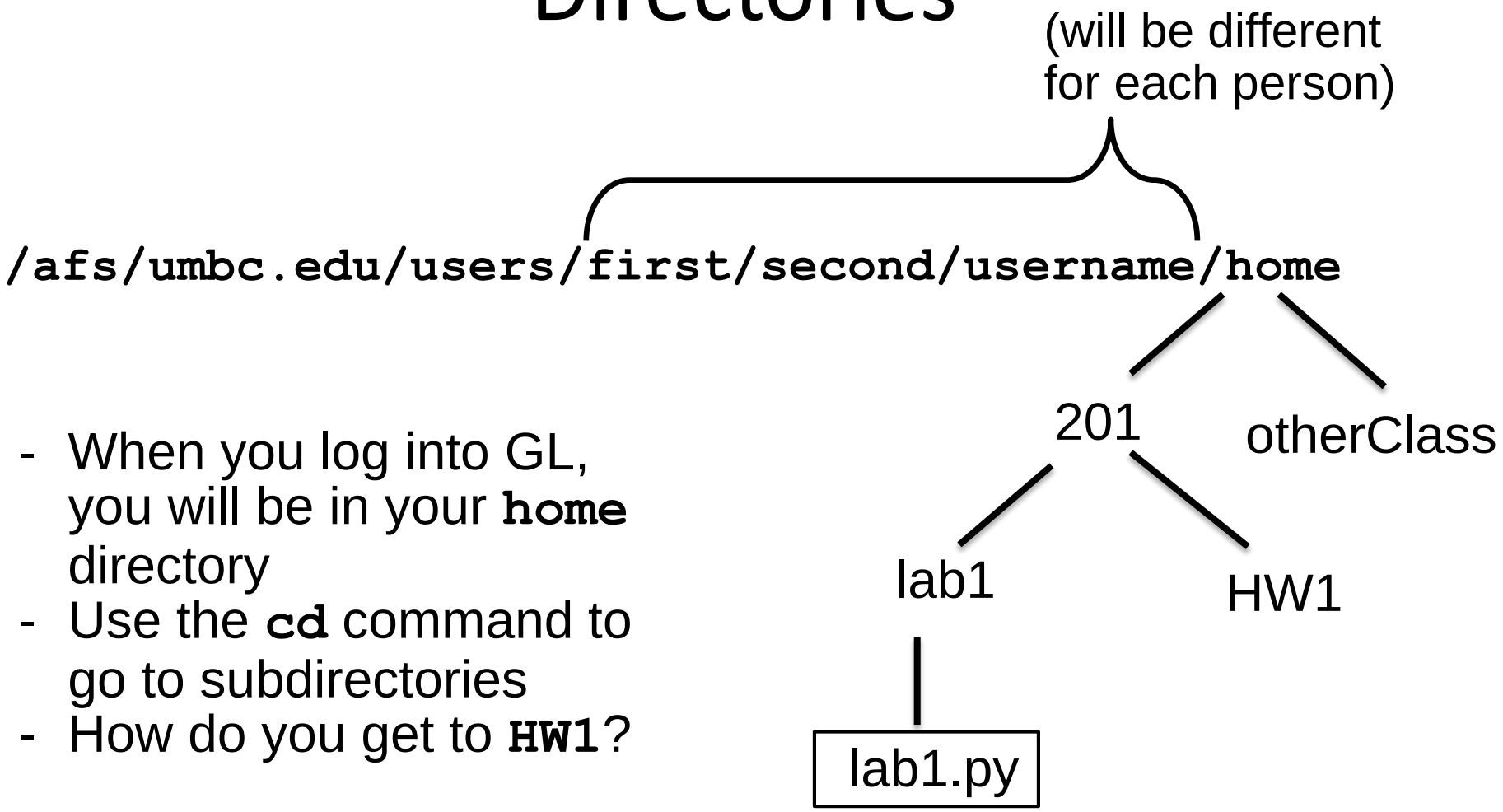
`cd NAME` – change directory

`cd ..` – go to parent directory

`cd .` – stay in current directory

`mkdir NAME` – make a new directory

Directories



- When you log into GL, you will be in your **`home`** directory
- Use the **`cd`** command to go to subdirectories
- How do you get to **`HW1`**?

emacs – A Text Editor

- Will use emacs to write our python code
- emacs is CLI, not GUI
  - Need to use keyboard shortcuts to do things
- Reference:
  - http://www.csee.umbc.edu/summary-of-basic-emacs-commands/

Keyboard Shortcuts for emacs

- To open a file (new or old)

  `emacs filename_goes_here.txt`

- To save a file

  `CTRL+X` then **`CTRL+S`**

- To save and close a file

  `CTRL+X` then **`CTRL+C`**

- To undo

  `CTRL+_` (that's "CTRL + Shift + -" for underscore)

Announcements

- Lab 1 this week is an online lab
  - Comes out Tuesday
- In-person labs won't begin until next week

- Make sure to log into the course Blackboard
  - Let us know if you have any problems
  - (Students on the waitlist may not have access yet)