CS1315

Introduction to Media Computation

Instructors

Mark Guzdial
Office: TSRB 324
Email: mark.guzdial@cc.gatech.edu

Bill Leahy
Office: CCB 104B
Email: bleahy@gatech.edu

Teaching Assistants

Katherine "Kate" Harlan
Araya Zaesim
Samera Ahmad
Lauren Liou
Mary-Ann Ionascu
Stacey Jones
Robert Allen
Jennifer Armbruster
Elizabeth Schunk
Brett Lyle
Kristen Hopper
Catherine Bennett
Hazel Shah
Annie Mei
Nicolas Fabre
Anna Hedden
Casey Evanish

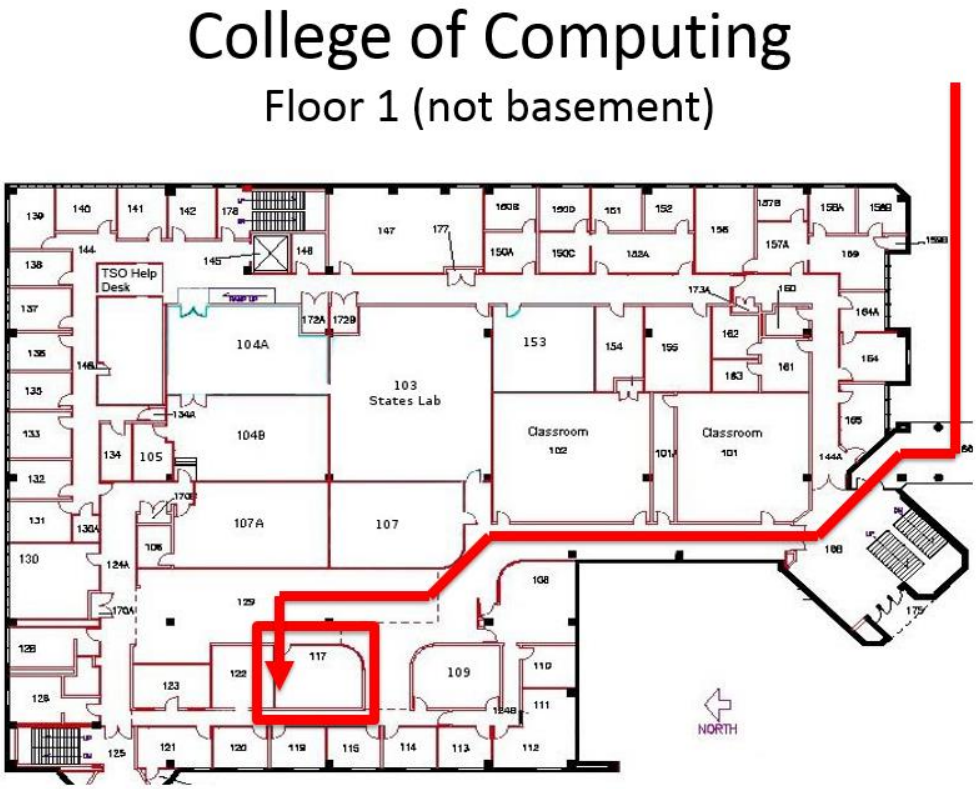
Where to Find a TA

Help Desk or TA Office Hours will be held in the College of Computing – CCB 117

That room is affectionately known as
“the fish bowl”

Schedule of Help Desk hours will be posted on T-Square

You are free to see any CS1315 TA for help



Course Objectives

Write programs for mathematically based calculations, image and sound manipulation, and text (e.g. HTML) creation/manipulation

Apply basic concepts of computer science (especially data representations, algorithms, encodings, forms of programming) to media related problems

Apply a range of useful computing skills

Purpose

CS 1315 is an introduction to the concepts of computing for students with no previous background in CS and who are majoring in subjects other than CS or engineering. Students learn computing through the design of programs to manipulate media and multimedia objects such as digitized pictures, sounds, animations/movies and web pages.

Learning Outcomes

(Competency) Calculate size requirements for media objects given size and quantity of elements and knowledge of encoding schemes (e.g. bits in a picture, given a knowledge of RGB encoding and the width and height in pixels; size of a digitized sound given its sampling rate and duration).

(Competency) Demonstrate ability to write simple programs using function invocation, iteration, conditionals and simple sequential data structures (e.g. lists and text strings).

Learning Outcomes

(Competency) Apply knowledge of media encoding schemes (e.g. digitized pictures, sounds, movies and web pages) and basic programming syntax and semantics to write simple media creation and manipulation programs.

(Competency) Infer behavior of short media programs from their text, including detection of flaws in incorrect programs.

(Competency) Explain basic database, networking and algorithmic concepts (e.g. relational DBMSs, protocols, intractability, respectively).

Learning Outcomes

(Accomplishment) Write several programs that creatively manipulate media or multimedia objects and save or display the results.

(Experience) (For any student self-described as “not a computer person”: reflect on how computers are entirely controlled by people and depend on explicit instructions.)

Learning Outcomes

(Integrated performance) Write a 100+-line program that integrates multiple media creatively (e.g. “turtles” moving in choreography across a pictorial background and to music) demonstrating mastery of programming concepts (sequencing of commands, function invocation, repetition and conditionals).

Python

To help learn about computer science and programming, use of a real computer programming language is helpful

We'll be using Python. (Actually Jython to be exact.)

Jython?

Jython is the result of a python front end and some Java power under the hood

We won't worry about the Java part

Software - JES

JES stands for Jython Environment for Students

It's an example of an IDE (integrated development environment)

basically you can create/edit your code

save your code

run your code

and see the results all within the same software

Free and available for download from T-Square under resources/software

Book Info

Textbook

Guzdial, M. J. and B. Ericson, Introduction to Computing and Programming in Python, **3rd edition**. ISBN-10: 0132923513 ISBN-13: 9780132923514

buy the paperback,

rent it,

or there is a Kindle version

Online Free Resource

Elkner, J., A.B. Downey and C. Myers, How To Think Like a Computer Scientist: Learning with Python (2nd Edition, Including Python 2.x)

URL available now on T-Square under resources

- http://interactivepython.org/runestone/static/thinkcspy/index.html

Grading Breakdown

40% homework (actual computer programming)

40% in-class quizzes (Every week!)

20% final exam

Quiz Policy

There will be a about 10 quizzes. They will be weekly starting in week 2 and run through about week 12.

If you miss a quiz without a valid excuse you receive a 0.

If you have a valid excuse there will be two makeup quizzes given after the normal quizzes are over (weeks 13 and 14). If you miss more than two quizzes for valid excuses we will handle that on a case by case basis.

Valid excuses are: documented illness, judicial procedures, military service, or official school functions. Documentation must be provided on letterhead with the signature of a physician, supervisor, or other appropriate official.

If you arrive to a quiz late, you may be refused admittance and get a zero.

Final Exam Policy

The final exam will be optional. If you are satisfied with your grade at the end of the normal term you may elect to skip the final and receive your grade as is.

Grading Disputes

Present grading disputes to the Head TA rather than your individual grading TA as this helps fairness and oversight

Follow the chain of command, if the Head TA does not resolve the matter, see your Instructor

All grading disputes must be initiated within one week of the grade being made available

Dispute sooner rather than later!

This applies to all graded material

- Homework
- Quizzes
- Final (usually taken up with the instructor)

Workshop

Each Thursday you will attend a workshop - see Oscar

The session consists of 50 or fewer students – better for hands on help and easier interaction

Session is led by two TAs

Workshop starts this Thursday – be there!

Take your laptop

Attendance

Lecture and Workshop attendance is required.

We will assume you are attending.

Weekly Meetings with TA

You and your TA will pick a time to meet each week. Maximum 30 minutes.

Will review your quiz

Will demo your homework

If you are a sinner you will be assigned penance

If you perform the penance you will be redeemed for up to 80%! Hallelujah!

As the semester progresses only a fraction of the class will be required to meet with their TA each week.

Email Policy

Email us from your official GT email account

Include [CS1315] and a descriptive subject, such as the following:

> [CS1315] Lost on HW2

Be professional - sign your email with your name

Academic Honor Code

We expect every student to read, understand, and follow the Georgia Tech Academic Honor Code

http://www.honor.gatech.edu/

Collaboration Policy

Academic misconduct is taken very seriously in this class.

While homework assignments are collaborative, quizzes, the midterm, and the final exam are individual work.

In addition, your homework assignments may be evaluated via demo or code review.

During this evaluation you will be expected to be able to explain every aspect of your submission.

Each individual programming assignment must be coded by you. You may work with others, but each student must turn in their own version of the assignment.

Submissions that are basically identical will receive a zero and may be sent to the Dean of Students Office of Academic Integrity.

Be very careful when considering supplying your work to a classmate that promises just to look at it. If he or she turns it in as his or her own you will both be charged. (Don't do it!)

You are never required to show your code to another student!

T-Square

https://t-square.gatech.edu/portal

Syllabus

Assignments

Turn-ins

Announcements

Grades - feedback

TA Information posted soon

Not forgiving about due dates!

T-Square Turn-In

- Check for your confirmation email
- No email means nothing was turned in (it's tricky!)
- Double check submission by downloading and running
- It's the only way to be sure
- Can resubmit updated files until the due date/time
- Resubmit *all* files if you do resubmit
- T-Square is hardcore about the due time. Imagine a train taking off whether or not you are fully onboard. It has no love.

Tips

Try the code we do in class – don't be afraid to edit it, change it, and see what happens

In fact, type along during class!

Do your homework! Learning to program is like learning a sport. It takes actual practice to get comfortable and proficient

Keep up with the reading

Be prepared when you go for help from a TA. Bring your work and laptop with you!

Start assignments early

Programming errors can be tricky

Starting early gives you time to seek help if needed

Take initiative. Begin early and be determined to succeed!

Reminders

Workshop starts week 1

See Oscar for the room/time details

Take your laptop!

Start reading How to Think Like a Computer Scientist - see link on T-Square under Resources