CS312 Course Introduction

“There is so much to gain, so much to learn, and so much to teach in the technology industry. The possibilities are virtually limitless. Interactive websites can teach language, educate children, and keep language alive.”

Andrea Delgado-Olson
Program Manager at Anita Borg Institute
Masters Student in CS, Mills College
Member, Ione Band of Miwok Indians
https://anitab.org/community-voices/i-am-a-miwok-indian-and-a-womenintech/

Chand T. John
chand@cs.utexas.edu
The Instructor

- CS & Math @ UT
- Ph.D. CS @ Stanford
- Robotics software engineer
  - Honda
  - Google + Johnson & Johnson medical robotics startup
- Asst. Prof. of Instr. @ UTCS
- Owner & creator, https://cs-comics.com
- Author, TED-Ed lesson on sorting, almost 2M views (https://www.youtube.com/watch?v=WaNLJf8xzG4)
Women-in-STEM Comics

See more @ my website https://www.cs.utexas.edu/~chand/
What We Will Do Today

- Introductions and administrative details
- Start Java Basics
What is this class?

- Computer science is the study of managing information using the weird constraints of tools called computers.

- Computer **programming** is a skill needed in computer science.

- This class will give you a foundation in Java, a widely used programming language.

- If you’ve never programmed a computer before, you belong here!
Learn to design and implement programs to solve complex problems.


2. Piazza: Make sure you can read messages

3. Practicelt: Create an account

4. CS department account: Request one from https://apps.cs.utexas.edu/udb/newaccount/
Optional Book

BUILDING JAVA PROGRAMS
A BACK TO BASICS APPROACH
4TH EDITION

Stuart Reges & Marty Stepp
Question

Which of these best describes you?
A. First year at UT and first year college student
B. First year at UT, transferring from another college or university.
C. In second year at UT.
D. In third year at UT
E. Other
Syllabus

- cs.utexas.edu/~chand/cs312
Assignments

- Start out easy but get **much, much** harder
- Individual – do your own work
- Programs checked automatically with plagiarism detection software
- Turn in the right thing - correct name, correct format or you will lose points / slip days
- Slip days
  - 8 for term, max 2 per assignment
  - don’t use frivolously
Succeeding in the Course

- Practice, practice, practice!
- Ask questions: lectures, sections, office hours, Piazza
- Make mistakes, learn from them
- Study with others
How to Get Help

- Piazza Post
- Office/lab hours
- Email instructor or TAs
  - Prefer Piazza
- Class examples
- Examples from book
- Discuss with other students at a high level
Succeeding in the Course - Concrete

- Whole course is cumulative!
- Material builds on itself
  - failure to understand a concept leads to bigger problems down the road, so …
- do the readings
- start on assignments early
- get help from the teaching staff when you get stuck
- attend lecture and discussion sections
- participate on the class discussion group
- do extra problems (Practice It! http://practiceit.cs.washington.edu/)
- study the old tests
- study individually and in groups
- ask questions and get help when needed
Success in course – the key

- Design patterns
- Knowing what a for loop is, is trivial.
- Training yourself on how to solve a problem like, reverse all the characters in a String, using a for loop, is key, especially for exams.
Succeeding in the Course

- Cannot succeed via memorization.
- The things I expect you to do are **not** rote.
- Learn by doing.
- If you are brand new to programming or have limited experience, I **strongly** recommend you do lots and lots of practice problems.
  - Practice It! web site
  - JavaBat
Common Mistakes

- Not getting help
- Not turning in the correct file(s) for programming assignments
- Going to the wrong section
Course Materials and Procedures

- Software
  - can work in CS department microlab, 1\textsuperscript{st} and 3\textsuperscript{rd} floor of Gates, north wing (GDC)
  - login via CS account name and password
  - work on your own computer if you wish
  - Java
    - Web page has details under Software. - JDK 8.0
  - Optional IDE
    - Recommended IDE is Eclipse (free)
Programming is problem solving

- Break up problem into key parts
- Write code for each part
- Like planning a party
How many computers do you own?
Programs are like recipes

**INGREDIENTS**

- 3 cloves garlic
- 16 oz canned garbanzo beans
- 3 tbsp lemon juice
- 3 tbsp tahini
- ¼ tsp salt

**INSTRUCTIONS**

1) Peel and chop garlic.
2) Open garbanzo can and drain the fluid.
3) In a blender, blend garlic.
4) Add all other ingredients and blend.

(c) 2018 Chand T. John
How computers run programs

WELL, BEFORE RUNNING THE PROGRAM, A PROGRAMMER HAS TO RUN A SPECIAL PROGRAM CALLED A COMPILER...

...WHICH TRANSLATES THE PROGRAM INTO A LANGUAGE CALLED ASSEMBLY LANGUAGE...

```assembly
movl  $0x2,-0xc(%rbp)
movl  $0x3,-0x8(%rbp)
mov   -0xc(%rbp),%edx
mov   -0x8(%rbp),%eax
add   %edx,%eax
mov   %eax,-0x4(%rbp)
mov   $0x0,%eax
pop   %rbp
retq
```

...AND THEN AN ASSEMBLER CONVERTS THAT INTO THE MACHINE LANGUAGE OF 0s AND 1s THE COMPUTER SPEAKS!

THESE 0s AND 1s REPRESENT THE PROGRAM'S INSTRUCTIONS AND GET STORED IN THE TEXT SEGMENT OF THE PROGRAM'S VIRTUAL ADDRESS SPACE.
High-level languages

“Only by standardizing high-level programming languages can progress in computer applications be accomplished efficiently and in a cost-effective manner.”

Rear Admiral Grace Hopper, Ph.D.