TCSI201 Hon. Intro. CS.

- Skim now/read later Prospectus/Syllabus Details.
- “Classes” MWF: 9:20AM-10:15 reading/hw readiness quizzes, active learning exercises and discussions, not much lecturing!
- Starting TODAY: “Labs” Mondays ES-B19 Team exercises with lab Myro kits. 4:15PM-6:15.
- Tutorial meetings (prof or TAs): ES-B19 Times to be arranged. FILL OUT Time Questionnaire.
- Discussion feature of Blackboard (UA online).
Active and Team Learning

- We will form 4 teams to do (almost?) all Lab activities each together. TODAY, we will form teams based on questions and randomness.

- The questionnaire could have been done online, except you (rather than the computer) will run the questionnaire program yourself because a chief goal is: **Learn to read and write programs!**

- **Active** puzzling/problem solving and seeking/giving help to classmates makes learning much faster and more fun. **ROBOTS and GRAPHICS even MORE!**
Individual Responsibilities

- On homework: DO the problem solving, coding, testing, debugging, refining ON YOUR OWN STEAM, with consulting BUT NOT COPYING.
  - OK to share robot/fluke, NOT your homework files.

- When you get help: Question until you (think you) understand, then DO IT YOURSELF on blank paper or a new file. (Then, you'll know for sure.)
  - ONLY WAY to get skills useful for doing / building.
  - Assessment of individual skill is the major grade factor.
Individual Responsibilities

- Do assigned readings and exercises on time.
- Learn some motivation, details and concepts from studying that will NOT be gone over in class.
- Besides doing readiness quizzes, PROVIDE the class and professor QUESTIONS about what you need to know or what you find confusing.

I've been thinking in the subject terms for 40 years. It's hard to tell what beginners find easy or hard, or how long they need to use something obvious to me.
Up to you: Don't Cheat!

- It makes grading unfair and inhibits your learning.
- Warning if code or solutions appear to have some copying.
- Major copying (from classmates or others) on hw: 0 grade.
- Midterm or Final exam cheating or computer/network vandalism: Course Failure.
- All penalties are reported to the Honor's Dean and Office of U.G. Studies per UAlbany regulations and may be referred to the UAlbany Judicial System.
YOU will soon read a small but REAL program!

< 5 minutes of vocabulary

- The IDLE software runs a Python interpreter:
  - You type an operation; computer/robot executes (does) it; the returned value if any is printed.

- **Sequential** execution:
  - (usually) Operations are executed in the order they are written.

- Python function: a pre-typed sequence invoked by name so you don't have to retype it repeatedly.
Small but REAL programs are easy to read.

< 5 minutes of vocabulary

- The IDLE software runs a **Python interpreter**: a **programming language**
  - You type an **operation**; computer/robot **executes** (does) it; the **returned value** **if any** is printed.

- **Sequential** execution:
  - (usually) Operations are executed in the order they are written.

- **Python function**: a pre-typed sequence **invoked** by **name** so you don't have to retype it repeatedly.
Small but REAL programs are easy to read.

< 5 minutes of vocabulary

- The IDLE software runs a Python interpreter: a programming language
  - You type an operation; computer/robot executes (does) it; the returned value if any is printed.

- Sequential execution: It's like a calculator.
  - (usually) Operations are executed in the order they are written.

- Python function: a pre-typed sequence invoked by name so you don't have to retype it repeatedly.
Small but REAL programs are easy to read.

< 5 minutes of vocabulary

- The IDLE software runs a **Python interpreter**: a programming language
  - You type an **operation**; computer/robot **executes** (does) it; the **returned value** if any is printed.

- **Sequential** execution: It's like a calculator.
  - (usually) Operations are executed in the order they are written. *Easy idea: USING it is critical! Robots help you learn to.*

- **Python function**: a pre-typed sequence **invoked** by **name** so you don't have to retype it repeatedly.
Small but REAL programs are easy to read.

< 5 minutes of vocabulary

- The IDLE software runs a **Python interpreter**: a programming language
  - You type an **operation**; computer/robot **executes** (does) it; the **returned value** if any is printed.

- **Sequential** execution: **It's like a calculator.**
  - (usually) Operations are executed in the order they are written. *Easy idea: USING it is critical!* Robots help you learn to.

- **Python function**: a pre-typed sequence **invoked** by **name** so you don't have to retype it repeatedly.
  *Computer programs ARE such stored sequences.*
Program Demo

- Start IDLE. Demonstrate some operations, values, functions, sequences......
- Open the file containing the program.
- Run the file contents.
  - Now the functions are stored.
- >>> (Type in:) main( ) (Enter)
- The operations in function main are executed in sequence, including invocations of other functions, such as getExperienceNumber
### TCSI201 UAlbany Honors Intro. to Computer Sci. Questionnaire for getting acquainted and gauging prior experience.

NAME ____________________

```python
def getExperienceNumber():
    x = 0
    # The # made this line a comment. The computer ignores comments.
    # x is now DEFINED and has value 0, (number zero, not letter Oh).
    x = x + askQuestion01("Do you now have either your own or a shared Myro Robot?")
    # What is your answer? (Yes or No)
    # What is x now? It is 1 if the answer was Yes, 0 otherwise.
    # Write the current value of x here:
    
    x = x + askQuestion01("Did you make the robot move with Myro commands?")
    # What is x now? Write it here:
    # Why must you have done addition?
    
    x = x + askQuestion01("Have you written a web page in html or xhtml?"
    # What is x now? Write it here:
    
    x = x + askQuestion01("Have you ever written a computer program, even a simple one?"
    # What is x now? Write it here: If your answer was yes, what language(s)
    
    x = x + askQuestion01("Did you take a programming course?"
    # What is x now? Write it here:
    
    x = x + askQuestion01("Did you take college programming or AP computer science?"
    # What is x now? Write it here:
    
    x = x + askQuestion01("Did you do any robotics before?"
    # What is x now? Write it here:
    
    x = x + askQuestion("How many semesters of calculus did you take, including AP", [0, 1, 2, 3])
    # What is x now? Write it here:
    
    x = x + askQuestion("Enter 1 if you're a freshman, 2 soph, 3 jr, 4 sr.", [1, 2, 3, 4])
    # What is x now? Write it here:
    
    x = x + askQuestion("From your birthdate, select the one's digit.", range(0, 10)) % 2
    ########### The remainder when a is divided by b is a % b.
    # Take a guess at what is a longer way of writing range(0, 10). Write it here:
    
    import random
    x = x + random.randint(0, 1)
    # Toss a coin. Add 0 to x if tails, 1 to x if heads.
    # What is x now? Write it here:
    
    return(x)  # Write the value of x that is returned:
```

```python
def askQuestion01(question):
    # This function asks a given yes-no question.
    # If the answer is no, return 0.
    # If the answer is yes, return 1.
    reply = askQuestion(question, ['Yes', 'No'])
    if reply == 'Yes':
        return 1
    else:
        return 0
```

```python
def main():
    print getExperienceNumber()

# Congratulation! Unless you know Python, you have read your first Python computer program.
```

---

Start here, at main()!

Intended for web reading only!

Zoom in to read. (This material on paper was distributed, written in and collected in class.)
Form the Teams..
x is a variable

- Within the function `getExperienceNumber`, x is a variable.
- **Assignment** statement `x = 0` expresses the operation to MAKE THE VALUE OF x become 0.
- The function calculates your experience number by repeatedly changing the value of x by adding 0 or 1 according to your answers.
- The other assignment statements `x = x + .......` (with **addition operators**) might CHANGE the value of x by adding a number to it. **IS ADDING 0 AN OPERATION?**..vote.. discuss..
We use naturally this and other core vocabulary

- interpreter
- operation
- execute
- value
- returned value
- sequential execution
- function

- invoke (or) call a function
- function return operation
- variable
- value of a variable
- assignment to a variable.

Study guidance!
Can you: (1) Explain each in your own words? (2) Give examples? (3) “Trace” a Python program in terms these things doing these actions?
Today's 4:15 Team activity.

- One laptop/Myro kit/team; 1 copy K1-3 / person.
- Play with direct robot commands from K1-3 and variations of them.
- Compose commands into functions, save and rerun function definition code from a file.
- CREATE an interesting robot demo with at least the K1-3 materials, coded in a file to be shared.
- Teams: Explain and share demo. code in last ½ hour and some in Wednesday's class.
First Homework

- If you haven't done so, download (or get CD) and install the Myro software; and, if you have a Fluke & Robot, try it out.

- Review Kumar Ch1-3 (after lab.) and go through ThinkCSPy Ch 1-3 by actively trying the Python examples and your variants in the IDLE interpreter. (They are short and I think easy.)

- Take handout on using Python on UA computers. USE it if you can't try the ThinkCSPy otherwise, and you will use it here Wed. or Friday.
First Homework

- Do not dwell too long on anything you find confusing. Study ThinkCSPy Ch1-3 at the computer only 1-2 hours now to Wed. AM. Ask someone else, take a break, and write notes or questions and raise them in class. Even better, post on Blackboard!

- There will be a quiz on Wednesday, after we address questions.

- Plan: quiz, finish Lab. work sharing, explore “while” loops. You will be ready after Lab.&HW.