Intro to Computer Science CS-UH 1001, Spring 2022

Lecture 7 – 2D Lists and Boards, Condition-controlled Loops, Comments

Today's Lecture

- Recap: Dynamically Creating a 2D List
- Creating and Printing 2D Boards
- Condition-controlled Loops
- Comments in Python

Recap

- List of lists:
  - Create them using

    numbers = [[1,2,3], [1,2,3], [1,2,3]]
  - Access a single element by its index

    numbers[row][column]

Recap: Dynamically Creating a 2D List

```
cnt = 1
outer_list = []
for r in range(3):
    row_list = []
    for c in range(3):
        row_list.append(cnt)
        cnt = cnt + 1
    outer_list.append(row_list)

print(outer_list)
```

[[1, 2, 3], [4, 5, 6], [7, 8, 9]]

How NOT to Create a 2D List

```
cnt = 1
outer_list = []

for r in range(3):
    row_list = []
    for c in range(3):
        row_list.append(cnt)
        cnt = cnt + 1
    outer_list.append(row_list)
```

What are the dimensions of the list?

How NOT to Create a 2D List

```
cnt = 1
outer_list = []

for r in range(3):
    row_list = []
    for c in range(3):
        row_list.append(cnt)
        cnt = cnt + 1
        outer_list.append(row_list)
```

What are the dimensions of the list?

Dynamically Printing 2D Lists

Option 1, using list indices and range():

```
for r in range(3):
        for c in range(3):
                print(outer_list[r][c], end=" ")
        print()
```

Note: You have to know the dimensions of the list!

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Dynamically Printing 2D Lists

Option 2, using list items:

```
for row in outer_list:
        for item in row:
                print(item, end=" ")
        print()
```

Note: You do not have access to the indices of the list!

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Another 2D List Example

- Assume you want to create a database storing information about each student in class
  - Name
  - Age
  - Major
  - Year of graduation

Another example: 2D Lists

- **How about creating a 2D list with each row being the information of one student?**

	0	1	2	3
0	John	20	CS	2020
1	Michael	19	Econ	2022
2	Adam	21	Chemistry	2021

Another example: 2D Lists

```
NUM_STUDENTS = 3
students_list = []

for i in range(NUM_STUDENTS):
    student_info = []
    student_info.append(input("Please enter student's name? "))
    student_info.append(input("Please enter student's age? "))
    student_info.append(input("Please enter student's major? "))
    student_info.append(input("Please enter student's grad year? "))
    students_list.append(student_info)

print (students_list)
```

Another example: 2D Lists

	0	1	2	3
0	John	20	CS	2020
1	Michael	19	Econ	2022
2	Adam	21	Chemistry	2021

- **How to get Michael's major?** [1][2]
- **How to get Adam's age?** [2][1]

Iterate through 2D lists

- A nested for loop can be used to iterate through the list

```
for student in students_list:
    for info in student:
        print(info)
```

First for loop: loops through the students

Second for loop: loops through all student items

Overall Program

```
NUM_STUDENTS = 3
students_list = []
```

```
for i in range(NUM_STUDENTS):
    student_info = []
    student_info.append(input("Please enter student's name? "))
    student_info.append(input("Please enter student's age? "))
    student_info.append(input("Please enter student's major? "))
    student_info.append(input("Please enter students grad year? "))
    students_list.append(student_info)
```

Populates the list with student information

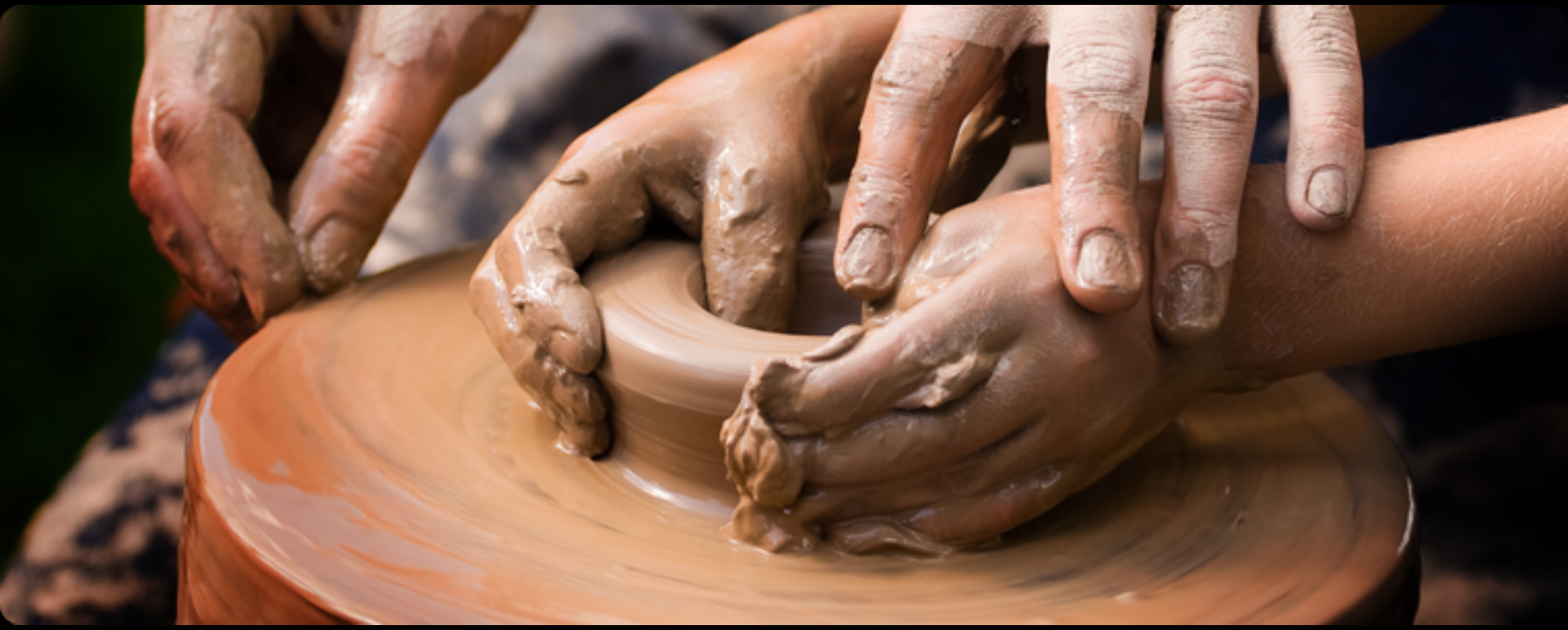
```
for student in students_list:
    for info in student:
        print (info, end=" ")
    print ("\n------------------")
```

Nested for loop: Prints the student's information

Output:

```
John 20 CS 2020
------------------
Michael 19 Econ 2022
------------------
Adam 21 Chemistry 2021
------------------
```

Breakout session I:
Nested loops and 2D lists



Calculating student grades (ex_7.1.py)

You have the following 2D list:

```
student_grades = [['John', '9', '10', '7', '6'],
                  ['Mary', '9', '8', '8'],
                  ['Smith', '8', '4'],
                  ['Adam', '6', '4', '7', '5', '10']]
```

This 2D list has the grades of quizzes for four students

1. Calculate the average grade per student
2. Calculate the average grade of all grades

Creating and Printing 2D Boards

Creating 2D board

- How to create a 8x8 board for a game?

.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Note: The . is used in this example instead of a whitespace character

Creating 2D board

Constant Variable

```
NUM_ROWS = 8
NUM_COLS = 8
board = []

for row in range(NUM_ROWS):
    row_list = []
    for col in range(NUM_COLS):
        row_list.append('.')
    board.append(row_list)

print(board)
```

board

	0	1	2	3	4	5	6	7
0	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.

```
[['.', '.', '.', '.', '.', '.', '.', '.'], ['.', '.', '.', '.', '.', '.', '.', '.'], ['.', '.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.', '.'], ['.', '.', '.', '.', '.', '.', '.', '.'], ['.', '.', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '.', '.', '.', '.'], ['.', '.', '.', '.', '.', '.', '.', '.']]
```

Printing 2D board

```
NUM_ROWS = 8
NUM_COLS = 8
# board was created on previous slide

for row in range(NUM_ROWS):
    for col in range(NUM_COLS):
        print(board[row][col], end = ' ')
    print()
```

Output:

```
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .
. . . . . . . .
```

Printing a 2D Board with Grid and Numbers

```
   0   1   2   3   4   5   6   7
 +---+---+---+---+---+---+---+---+
0| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
1| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
2| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
3| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
4| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
5| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
6| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
7| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
```

Do you think it is a good idea to add the grid and numbers to the board itself?

Printing 2D Board with Grid and Numbers

```
for cols in range(NUM_COLS):
    print('   ' + str(cols), end='')

print("\n +" + "---+" * NUM_COLS)

for row in range(NUM_ROWS):
    print(str(row) + '|', end=' ')
    for col in range(NUM_COLS):
        print(board[row][col] + ' | ', end='')
    print("\n +"+"---+"*NUM_COLS)
```

```
   0   1   2   3   4   5   6   7
 +---+---+---+---+---+---+---+---+
0| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
1| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
2| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
3| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
4| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
5| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
6| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
7| . | . | . | . | . | . | . | . |
 +---+---+---+---+---+---+---+---+
```

	0	1	2	3	4	5	6	7
0	L	e	t	s				
1		t	a	k	e			
2			a					
3				b	r	e	a	k

Condition-controlled Loops

Loops

- There are two kinds of loops:
  - A count-controlled loop (for loop)
  - A condition-controlled loop (while loop)

- Syntax:

```
while condition_expression:
    # indented code block that is being executed as long as the
    # condition expression evaluates to True
```

- A while loop iterates
  - as long as the condition is satisfied (True), it executes the indented code block
  - otherwise (False), it will stop the loop

Simple while Loop Example

```
count = 0

while count < 6:
    print("The count is: ", count)
    count = count + 1

print("Good bye!")
```

Output:

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
Good bye!
```

Another while Loop Example

- Assume you want to input all the names of students in this class into a list
  - Will you write 20+ input statements?
- What if it was the whole University?
  - Will you write 500+ input statements?
- What if you don't know how many inputs you need?
  - Solution: while loop with initial/exit condition

while Loop Example with Initial/Exit Condition

```
students_list = []
input_more_students = True

while input_more_students == True:

    student = input("Please enter student name: ")
    students_list.append(student)
    response = input("Input more, enter 'yes': ")
    if response != 'yes':
        input_more_students = False

print("Good bye")
```

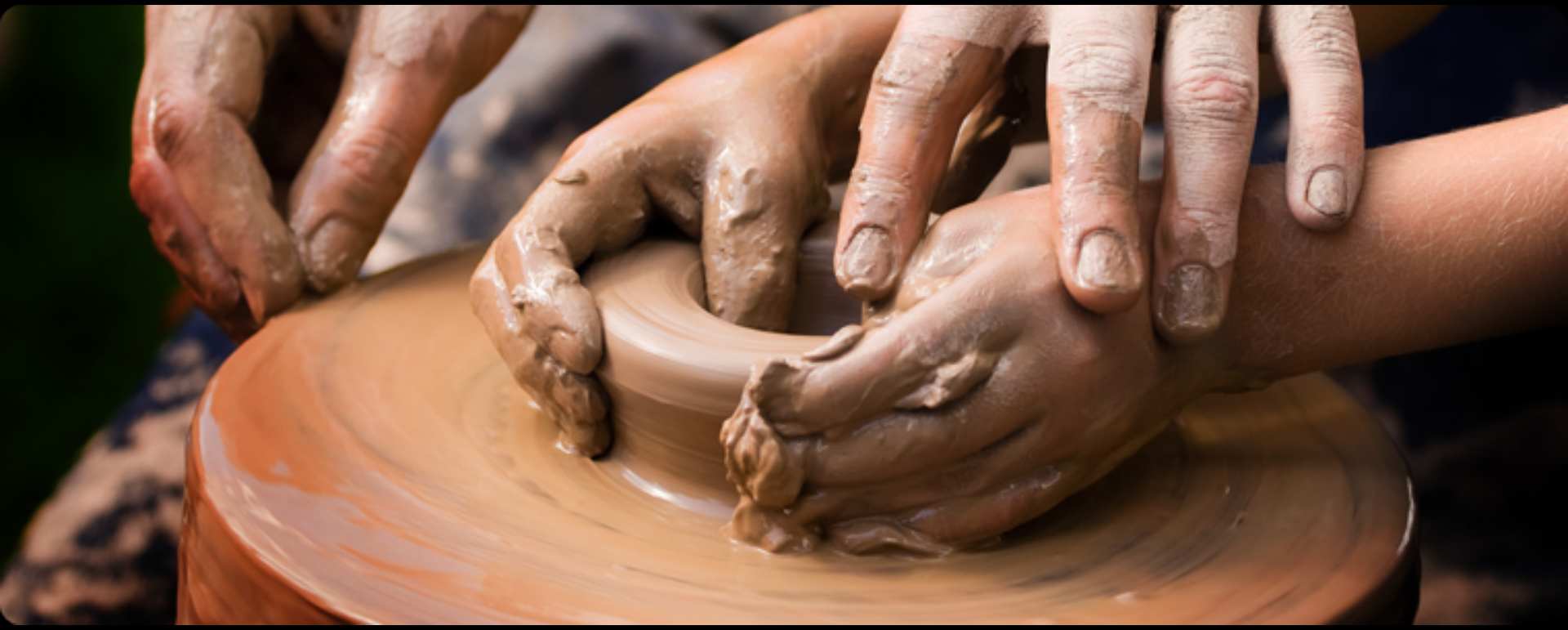
The input_more_students variable is initialized with True to ensure that the while loop will at least execute once

Change the initial condition to False to stop the loop

Infinite Loops

- A while loop becomes an infinite loop if the condition expression never becomes False
  - An infinite loop never ends and runs forever
  - In 99.9% of the cases, you don't want to use an infinite loop
- Programs with an infinite loop can be stopped by using Ctrl+c

Breakout session II:
While loop



Dice guessing (ex_7.2.py)

1. Write a program that emulates a dice roll and ask the user to guess the dice value (values between 1 and 6)
2. Inform the user if the guess was correct or wrong
3. Print the outcome of the dice as:

```
*---*
| 6 |
*---*
```

Hint: To emulate a random dice roll, you can use the random module:

```
import random   #add this to line 1 of your .py file

random_value = random.randint(FromINT, ToINT)
```

This will randomly generate an integer between the values FromINT and ToINT (both are inclusive, both must be integers).
For example, random.randint(1,2) would either return 1 or 2

Dice guessing (v2) (ex_7.2.py)

Use the dice guessing program from earlier

1. Extend it so that the program keeps asking the user to guess until the guess is correct

2. Add error checks for the input:
   - can the input be casted to an integer? (use the string method .isdigit() to check if the input is a digit)
   - is the number within the dice range?

break in Loops

A loop can be stopped from iterating using the break command

- Example for loop:

```
for i in range(5):
    if i > 3:
        break
    print(i)

print("Good bye")
```

- Example while loop:

```
i = 0
while True:
    if i > 3:
        break
    print(i)
    i = i + 1

print("Good bye")
```

Comments

Python Comments

- Comments can serve as notes to yourself, or they can be written with the intention of other programmers being able to understand what your code is doing
- Python comments are non-executable statements and start with a #
- All characters after the # and up to the end of the line are part of the comment and will not be executed by Python

- Example:

```
# This is a comment
```

Python Comments

- Comments are helpful to document your code (and you should do that!)

- Option 1:

```
# This will print 'Hello World'
print("Hello World")
```

- Option 2:

```
print("Hello World") # This will print 'Hello World'
```

Long Python Comments

- Python does not have a syntax for multi line comments, but there are still 2 options:

- Option 1:

```
# This is a comment
# written in more than
# one line
```

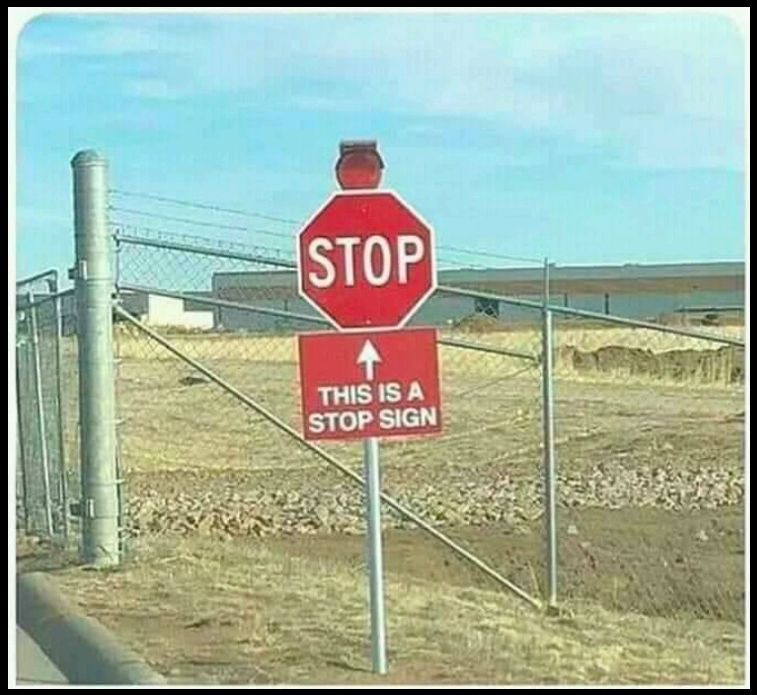
- Option 2 (Triple quotes):

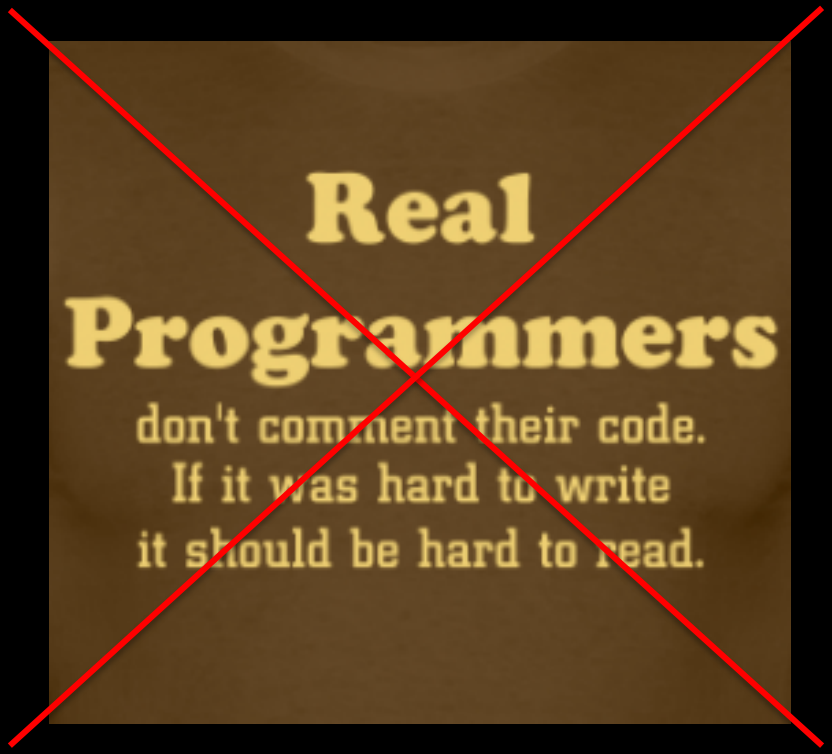
```
''' This is a comment
written in more than
one line '''
```

  – Python will ignore strings that are not assigned to a variable

How NOT to Comment Code!

- x = 10 # set the value of the variable x to 10
- print(”Hello World”) # This will print ’Hello World’





Next Class

- Lab on 2D Lists and Nested Loops
- Functions
- Modules