

CS1210 Lecture 44

Dec. 10, 2021

- HW 11 due Sunday, 8pm
- Scores are up to date.
 - One DS score has been dropped.
 - Only HW11 is not included.
 - Depending on your other HW scores, you can get from 0-7 additional HW points based on your HW11 score
- Course grading scales on next slide
- Optional final exam: Monday, Dec. 13, 12:30-2:30pm
 - Complete “Will you take the final exam?” ICON quiz/survey before noon Monday to let me know whether or not you will take the exam
 - Exam will be 80 minutes.

TODAY

- Python can answer an important question 😊 ...
- UI CS courses beyond this one
- The Halting Problem
- Questions about courses, careers, life?
- Information about the final exam

Final Exam is optional. The default is that you are NOT taking it. You must OPT IN and notify me if you want to take it

Grade scales the are same percentage-wise except for rounding differences

Without final

Possible points: 168

Grade	Points	approx %
A+	163	97
A	148	88.1
A-	141	83.9
B+	133	79.2
B	122	72.6
B-	117	69.6
C+	109	64.9
C	92	54.8
C-	83	49.4
D+	80	47.6
D	73	43.5
D-	67	39.9

Without final max: 168

HW: $10 \times 6 = 60$

DS: $10 \times 3 = 30$

Quizzes: $18 + 3 * 20 = 78$

HW11: 0-7 additional
depending on what
HW & DS assignments
you will drop

With final scale

Possible points: 200

Grade	Points	approx %
A+	194	97
A	176	88
A-	167	83.5
B+	158	79
B	145	72.5
B-	139	69.5
C+	130	65
C	110	55
C-	99	49.5
D+	95	47.5
D	87	43.5
D-	80	40

NOTE: POINTS ARE THE OFFICIAL SCALE – NOT PERCENTAGE

With the skills learned in this course you can
now answer the age-old question:
Which came first, the chicken or the egg?

```
def whichCameFirst():  
    chickenAndEgg = ['🥚', '🐔']  
    answer = sorted(chickenAndEgg)[0]  
    return answer
```

```
>> whichCameFirst()
```

??? What do you think will be printed ???

Comments on more other CS/Informatics courses

Informatics major (minus cognate)

CS2110
Programming
for Informatics

CS2420
Analyzing Data
for Informatics

CS2520 Human-
computer
Interaction

CS2620
Server-side
Development

Elective

CS3910
Informatics
Project

Related majors:
Data Science
Computer Science & Engineering)

CS major

CS2210
Discrete
Structures

CS2230 CS2: Data
Structures

CS2630
Computer
Organization

CS3330
Algorithms

CS2820 Object
Oriented
Software

CS3820
Programming
Languages

CS3620, 3640
(Operating Systems,
Networks)

CS4340, 4340, or
4350 . Theory, limits,
or logic in CS (BS
only)

Electives in: mobile and embedded computing, web programming, cloud computing, machine learning, HCI, security, databases, distributed systems, numerical and scientific computing, software engineering (different numbers required for BA/BS)

The Halting Problem

- it's important to know what we can and can't compute
- It turns out that we cannot create a program that can check *all* other programs for infinite loops
- see, e.g., http://en.wikipedia.org/wiki/Halting_problem
- First: demonstrate that we can write programs that create and execute new programs/functions.
testProgramOnInput.py
- Informal proof that we can't write doesItHalt
 - why can't we create fully correct doesItHalt function? (doesItHalt.py)
 - To see why, consider function test in doesItHaltTest.py

Halting Problem

- Consider trying to write program

doesItHalt(programString, dataString)

where

programString is a string representation of a program, e.g. “def foo(n): \n\treturn(n+1)”
and dataString represents the input to that program

such that doesItHalt returns

“Yes” if the program would halt on the specified input, and

“No” if the program would not halt (i.e. would go into an infinite loop)

- *It is not obvious that such a program can't be written.* But it should be clear that doesItHalt can't simply execute the specified program on the specified input. (Why?) Instead, doesItHalt would need to rely on more sophisticated analysis
- *HOWEVER, we can prove that doesItHalt cannot exist*

doesItHalt cannot exist

Informal proof:

Suppose doesItHalt exists. I.e. doesItHalt correctly determines/prints, for any possible program and input, whether or not the program halts on that input

Given assumption that `doesItHalt` exists,
we'll define function `test` as follows:

```
def test(programString):  
  
    result = doesItHalt(programString, programString)  
  
    if result == "No":  
        print("I'm done (hey, in fact, I halt)")  
  
    else:  
        loopFinished = False  
        while(not loopFinished):  
            print ("I'm gonna live forever ...")
```

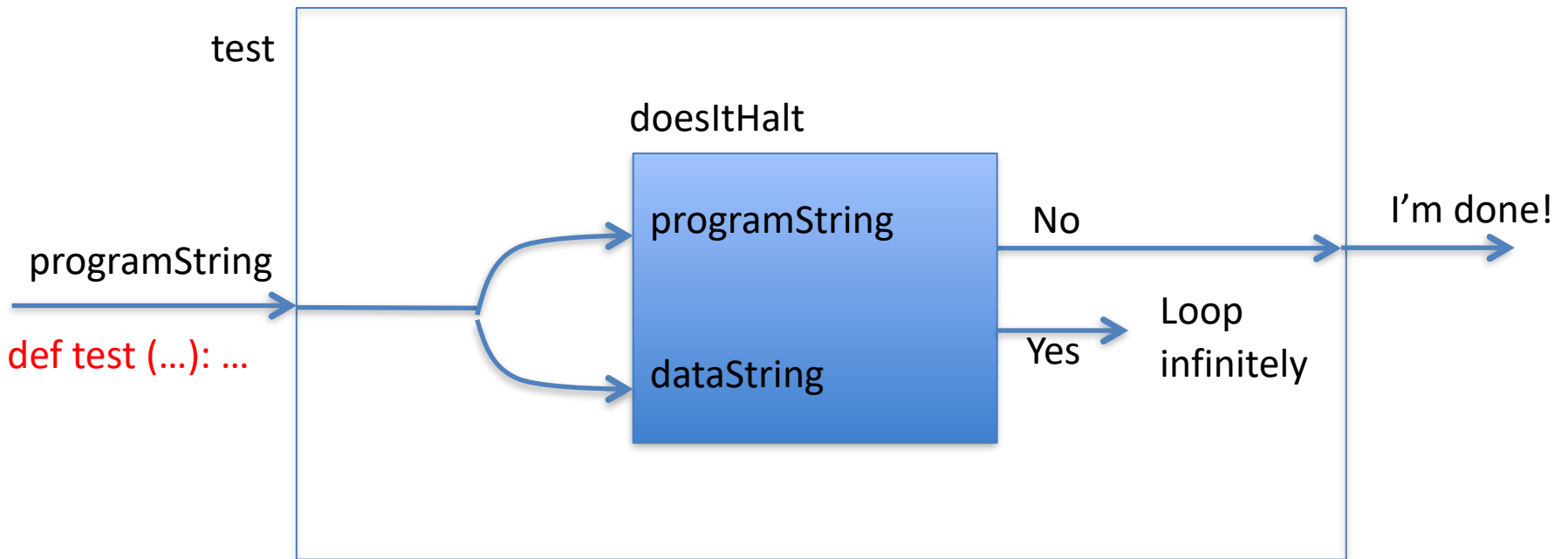
Consider: what happens when you
execute `test("def test ...")`?

doesItHalt does not exist

Informal proof:

1. Suppose doesItHalt exists (i.e. correctly states, for any possible program and input, whether or not program halts on that input)
2. Create function 'test' of previous slide. This is real Python code that works.
3. Now consider test("def test ...")
 - a. test("def test ...") first executes doesItHalt("def test ..", "def test .."), saving returned value in variable result
 - b. if result was "No" test("def test ..") clearly halts and returns.
 - c. If result was "Yes" test("def test ..") clearly loops forever.
 - d. BUT NOTICE! result would be "No" if doesItHalt determined that test("def test ...") would not halt! And would be "Yes" if doesItHalt determined that test("def test ...") would halt!
 - e. THUS, test("def test ...") halts if and only if test(def test ...") does not halt!!
4. This is a contradiction, so we must conclude that the original assumption, that doesItHalt exists, is false.

Depiction of this on next slide might be easier to follow



When does `test('def test ...')` print 'I'm done' (i.e. when does it halt)?

It halts when `doesItHalt('def test ...', 'def test ...')` returns No

But, by assumption, `doesItHalt('def test ...', 'def test ...')` returns No if and only if it determines that function `test` would not halt given 'def test ...' as input.

Thus, `test('def test ...')` halts if and only if `test('def test...')` does not halt.

When does `test('def test ...')` loop infinitely? (i.e. when does it not halt)?

It loops infinitely when `doesItHalt('def test ...', 'def test ...')` returns Yes

But, by assumption, `doesItHalt('def test ...', 'def test ...')` returns Yes if and only if function `test` would halt given 'def test ...' as input.

Thus, `test('def test ...')` does not halt if and only if `test('def test ...')` halts.

BOTH SITUATIONS LEAD TO A CONTRADICTION, SO THE ASSUMPTION THAT (A CORRECT) `doesItHalt` HALT EXISTS MUST BE FALSE.

Optional final exam

Setting: on Zoom, 12:30-2:30 pm, Monday, Dec. 13

Length: 5 - 8 questions, 80 minutes

To take the exam, you must answer “yes” on the “Will you take the final exam?” assignment on ICON by noon Monday.

Important note: If you open/begin the exam, it WILL be graded and your course grade will be calculated based on the 200-point with-final scale.

Main goal of exam: to test your ability to understand, analyze, and write small programs involving lists, dictionaries, strings, loops, functions, objects/classes

Material:

- all the material of Quizzes 1 through 4
- Possibly a programming question involving GUI programming with tinter
- Possibly a question related to HW10 & 11 but not requiring knowledge of Google or Twitter APIs. E.g. DS 11 would be an appropriate question.

Types of questions

- The same as those you’ve seen in previous exams, plus perhaps some multiple choice/matching questions
- Analyze, understand, explain code
- Write code

Specific topics:

- Expressions, variables, assignments, and functions
- Conditional expressions - if/elif/else
- Iteration/looping - while and for
- Lists and dictionaries
- Recursion
- Basic running time complexity – Big O notation
- Defining and using classes
- Sorting, binary search
- Graph representations – adjacency lists for directed and undirected graphs
- GUI programming with tinter
- basic familiarity with (but not specific knowledge of) programming with GUI/Google maps/Twitter material. E.g. questions like DS11 that don’t depend on specific API knowledge

Have a great break!

- I'm always available if you have questions about future courses, the major, etc. Send me email ...



Do something GOOD with your skills – the world needs innovative and creative thinking now and in the future 😊

