CCSI C200 Introduction to Computers and Programming

Spring 2019 Grade Report

Tidmarsh, Clare

Computer Science
School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

June 18, 2019

Assignment 1

Assigned: May 8, 2019
Due: May 10, 2019

Problem 1

problems.py

*Good work! *
*In Code 10, this is valid python. The issue is that greeting is referenced before assignment. *
If you assigned greeting a value before writing this line it would work perfectly.

Score: 45/50

Problem 2

myMath.py

Great job!

Score: 25/25

Problem 3

thecost.py

Awesome

Score: 25/25

Problem 4

myBall.py

Great job!

Score: 50/50

Total Score: 145/150

Assignment 2

Assigned: May 10, 2019
Due: May 14, 2019

Problem 1

myCalc.py

awesome work

Score: 210/210

Problem 2

old.py

Great job!

Score: 60/60

Problem 3

warping.py

-15 for extra print statements, -15 for no return value.

Score: 70/100

Problem 4

condprac.py

*fun1 sometimes returns multiple outcomes at once. *
fun2 still contains all the nots. Code is unchanged from source. (anything after a return is unreachable code and is never executed)

Score: 50/150

Total Score: 390/520

Assignment 3

Assigned: May 14, 2019
Due: May 17, 2019

Problem 1

looping.py

Great job!

Score: 105/105

Problem 2

stringStuff.py

−20 In palindrome, it returns when it checks whether the first and last letter same or not. did not checkwhole letter

Score: 60/80

Problem 3

whiling.py

Great job!

Score: 105/105

Problem 4

whileString.py

*Overall, my advice would be to stop putting your return statements inside the loop. There are cases where it is fine to return inside the loop, but you never have to. You have to be very careful with it unless you really understand it because it has a dramatic impact on the output of the function. Additionally for while loops, you have to update whatever variable is controlling the loop manually (unlike in for loops where it is done automatically) *
*In palindrome, your loop returns on the first iteration (because both the if and else have a return) *

*It never checks past the first and last letter. *
*getCount is an infinite loop. The return is in the wrong place (too early) but still it never returns. *
*getIndex failed all test cases because it also returns after just one iteration. *
areEqual uses the builtin way to test if strings are equal and the while loop is meaningless (function output is the same with or without it).

Score: 20/80

Problem 5

complexing.py

Did not print any of the values, therefore did not verify functions worked

positionalSum (5/30): The logic of going through the loops with the while was a good start, you return inside of the while loop after the first iteration.

scalarMatrix (5/30): Cannot combine a number and a list with the + sign

isSubstring (0/30): Not allowed to use find

whereSubstring (0/30): Not attempted

Score: 10/120

Total Score: 300/490

Assignment 4

Assigned: May 16, 2019
Due: May 21, 2019

Problem 1

functionsAsRec.py, farTest.py

isPalindrome: −20 for reversing the string and comparing. −25 fibonnaci does not use recursion nor does it compute the correct answer. It simply does integer subtraction. −15 for incomplete tests for sum2Dlist and removestring. −20 for not writing removeString.

Score: 80/160

Problem 2

aSort.py

Great sort, but it is not ours...

Score: 15/100

Problem 3

listAndDict.py, lodTest.py

*Nice! *
PrimeList works even without hardcoding all those early primes. It works if you remove all of them except 2.

Score: 100/100

Problem 4

myRec.py, recTest.py

−25 testRect needs to print first 10 values. No testcases for closedFunc1,2,5

Score: 135/160

Total Score: 330/520

Assignment 5

Assigned: May 24, 2019
Due: May 28, 2019

Problem 1

builtinCode.py

great

Score:220 /220

Problem 2

formatRunning.py

Great!

Score: 75/75

Problem 3

People.py, Reading.py, Testing.py

*various errors in people.py. age referenced before assignment in __str__, you did string.lst instead of self.lst in line 19, it doesn't account for whitespace/newlines, you removed the commas from your csv but split on commas. *
In reading.py you call people.getAge but that is not a function defined anywhere in your code. People.Age would work.

Score: 60/90

Total Score: 355/385

Assignment 6

Assigned: May 29, 2019
Due: May 31, 2019

Problem 1

People.py, BloodBank.py, main.py

−50 banks do not fill with blood correctly −40 list of people who can donate to A+ not close, −40 we should be able to transfer to Caleb Hubbard, Ralph Mercer, etc. −20 no blood units were updated for people

Score: 190/340

Problem 2

makeMeMove.py

*−20 for color changing *
Top of the screen: change to A DARK GREEN (current : light blue)
Left of the screen: change to A DARK YELLOW (but light green)
Right of the screen: change to A LIGHT BLUE (but orange)
*Bottom of the screen: change to A ORANG (dark green) *

Score: 180/200

Problem 3

inherits.py

*−20 for no test cases *
*The subclasses do not pass the right (number of) parameters to the superclass. Anything required for the init method of the superclass must be passed, just like any other function. *
*The Assignments class looks great, but the subclasses do not extend it properly. *
*There are a lot of indentation issues. *
The init methods should not be returning grades.
I am perplexed by the roster method in the attendance class.

Score: 40/120

Total Score: 410/660

Assignment 7

Assigned: May 31, 2019
Due: June 4, 2019

Problem 1

bestgraph.py, bgTest.py

−30 adjacencyMatrix() is not working. You just create a 0 contained list.

Score: 100/130

Problem 2

roman.py

−10 you don't print 5 to a line. Otherwise great!

Score: 60/70

Problem 3

balancing.py

−10 test cases only cover True values. The last of our supplied tests gives incorrect output, −10.

Score: 80/100

Problem 4

people.py

*Many errors in the Instructor class. *
*variables referenced before assignment in constructor. *
*Erroneous call to init_subclass in studentLst (which also has many variables referenced before assignment) *
Insufficient test cases and the Instructor class does not have any of the functionality described in the assignment.

Score: 55/100

Total Score: 295/400

Assignment 8

Assigned: June 5, 2019
Due: June 7, 2019

Problem 1

triangle.py

Good work.

Score: 80/80

Problem 2

mm.py

−20 No additional test cases

Score: 78/98

Problem 3

spaceship.py

+10 for attempt, −10 for using matplotlib

Score: 0/60

Problem 4

dfssearch.py

DFS works, but it is supposed to take input as command line args, not from input function

Score: 40/50

Total Score: 198/288

Assignment 9

Assigned: June 8, 2019
Due: June 12, 2019

Problem 1

shishkebab.py

Great!

Score: 260/260

Problem 2

converge.py

Great work

Score: 100/100

Problem 3

myGraph.py, sorting.py

You interpreted the results of the graph backwards. The lower the line is (in other words, the less time it took to sort), the more efficient the algorithm is. So bubble and insertion are actually the least efficient, with a quadratic runtime on average.

Score: 90/100

Total Score: 390/460