CSCI C200 Introduction to Computers and Programming

Spring 2020 Grade Report

Molt, Max

Computer Science
School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

April 28, 2020

Assignment 1

Assigned: January 22, 2020
Due: January 29, 2020

Problem 0

C200-Assignments

25 points total
10/10 Assignment 1 folder setup correctly
15/15 windchill.py, creditcard.py, and pizza.py exist and are named correctly (5 pts each)

Score: 25/25

Problem 1

windchill.py

25 points total
5/5 Windchill function looks correct
20/20 Passes test cases (5 pts each)

Nice job!

Score: 25/25

Problem 2

creditcard.py

25 points total
15/15 A reasonable attempt was made
10/10 Code works correctly

Very nice!

Score: 25/25

Problem 3

pizza.py

25 points total
5/5 A function was used
20/20 Graded on effort

Score: 25/25

Total Score: 100/100

Assignment 2
Assigned: 2020-01-30
Due: 2020-02-05

Mayhem

mayhem.py

195 points total
0/85 Function is syntactically correct (5 for each)
0/60 Points of functions (5 for each)(speed, distance, time, hours_to_min, min_to_sec, feet_to_mile, miles_to_kilometers, k_to_m, miles_to_feet, degrees_to_radians, p_to_k, ly_to_p)
0/50 Points for each (10 for each) (loc_c, c_to_f, f_to_c, k_to_f, pc)

−5: pc is incorrect.

Score: 190/195

Taxing

2019tax.py

50 points total
0/25 Proper implementation of the `unmarriedTax` function.
0/25 Proper implementation of the `marriedTax` function.

good work!

Score: 50/50

Circuit

circuit.py

34 points total
0/10 Utilizes `and`, `or`, and `not`
0/24 Correct output (3 points per each output)

good work!

Score: 34/34

Plotting

coolline.py

55 points total
0/25 Program runs
0/15 Title name changed
0/15 Function added to graph

good work!

Score: 55/55

Total Score: 329/334

Assignment 3
Assigned: 2020-02-06
Due: 2020-02-12

Structuring

35 points total
10/10 Assignment 3 folder setup correctly
25/25 funwithfunctions.py, qc1.py, if.py, precmetal.py, and myclock.py exist and are named correctly (5 pts each)
0/0 General notes to be aware of, not a deduction of points

Great job 100%

Score: 35/35

Problem 1

funwithfunctions.py

150 points total
90/90 Functions produce correct output (10 pts for each)
45/45 Functions look correct/ a reasonable attempt was made (5 pts each)
5/5 Answering the deliverable question
10/10 Provided code, i.e, test code, is from original document

Great job 100%

Score: 150/150

Problem 3

qc1.py

100 points total
10/10 Code runs
20/20 Will print a message correctly depending on Complex or Not Complex
20/20 Function properly returns a tuple of values
40/50 Functions return proper values

−10: Code failed 2 test cases.

Score: 90/100

Problem 4

if.py

60 points total
0/15 Part 1 converted from ifs to if-elif-else
0/15 Part 2 converted from 1 if to 2 ifs
0/15 Part 3 convert from many nots to fewer nots
0/15 Part 4 convert from one line to if-elif(s)-else

−60: no code submitted.

Score: 0/60

Problem 5

precmetal.py

80 points total
20/20 preciousMetalToDollars: returns the cost without modifying the original amount values. (5 pts per amount)

25/25 purchase: returns print statement, 2 lines if removing value, 1 line if not enough amount
25/25 purchase: Modifies the value correctly and does not change unless enough value is there
10/10 Did not modify original code

Great job 100%

Score: 80/80

Problem 6

myclock.py

30 points total
10/10 Clock runs
10/10 Title name changed to the student's name
10/10 Font changed from 'arial' to 'gothic'

Great job 100%

Score: 30/30

Total Score: 385/455

Assignment 4
Assigned: 2020-02-12
Due: 2020-02-19

Structuring

45 points total
10/10 Assignment 4 folder setup correctly
35/35 funtriangle.py, makeitrain.py, donor.py, palindrome.py, roman.py, moreloops.py, and farm.py exist and are named correctly (5 pts each)
0/0 General notes to be aware of, not a deduction of points

Great job 100%

Score: 45/45

Problem 1

funtriangle.py

45 points total
0/10 Triangle 1 created correctly
0/20 Triangle 2 created correctly with accurate pattern
0/15 Triangle 3 created correctly

−45: no submission.

Score: 0/45

Problem 2

makeitrain.py

45 points total
5/5 Code runs
10/10 Returns a list (or a tuple) with 4 values
10/10 Code looks reasonably correct
15/20 Test values

−5: Failed one test case.

Score: 40/45

Problem 3

donor.py

80 points total
15/15 Did not modify code outside of function
10/10 redbloodcompability: Returns the desired type (List or tuples of blood types)
5/5 transfusion: Returns a 0 or 1
10/10 transfusion: Modifies bank IF enough of the blood type
10/10 transfusion: Rejects if types not compatible or not enough in bank
15/15 Passes test cases for function redbloodcompability
15/15 Passes test cases for function transfusion

Great job 100%

Score: 80/80

Problem 4

palindrome.py

45 points total
0/20 Utilizes a loop
0/10 Returns a boolean
0/15 Passes test cases

−45: no submission.

Score: 0/45

Problem 5

roman.py

60 points total
15/15 Did not modify code outside of function
15/15 Returns a string
15/15 Did not modify code outside of function
5/15 Passes test cases

−10: Failed 2 test cases.

Score: 50/60

Problem 6

moreloops.py

175 points total
0/15 Function maxFor was reasonably attempted, run, and returns correct type (5 for each part)
0/15 Function maxWhile was reasonably attempted, run, and returns correct type (5 for each part)

0/15 Function minFor was reasonably attempted, run, and returns correct type (5 for each part)
0/15 Function RemoveEvens was reasonably attempted, run, and returns correct type (5 for each part)
0/15 Function myReplace was reasonably attempted, run, and returns correct type (5 for each part)
0/15 Function sumOdd was reasonably attempted, run, and returns correct type (5 for each part)
0/15 Function StringConcat was reasonably attempted, run, and returns correct type (5 for each part)
35/70 Passes test cases

−35: Funcitons − maxWhile, sumOdd, StringConcat failed all test cases. minFor failed 2 test cases.

Score: 35/175

Problem 7

farm.py

50 points total
15/15 Calculated f correctly
0/15 Utilizes the vol function (inside main or in f, either or) correctly
0/15 Does a brute force inside of main
0/5 Passes test cases

−35: Failed all test cases.

Score: 15/50

Total Score: 265/545

Assignment 5

Assigned: 2020-02-21
Due: 2020-02-24

Structuring

30 points total
10/10 Assignment 5 folder setup correctly
20/20 add.py, magic.py, votetally.py, votes.txt (5 for each)
0/0 General notes to be aware of, not a deduction of points

Great job 100%

Score: 30/30

Problem 1

add.py

45 points total
5/5 Code runs
10/10 Does not modify code outside of function
30/30 Gets the function to modify the label

Great job 100%

Score: 45/45

Problem 2

magic.py

59 points total
15/15 Code runs

10/10 Returns a number
14/14 Code contains all of the operations (2 points per each)
20/20 Test values

Great job 100%

Score: 59/59

Problem 3

votetally.py

75 points total
10/10 Makes it easy to determine what the number of yes and no's are
20/20 Values are correct
20/20 Path to read is 'Assignment5/votes.txt' per the directions
10/10 Reads the file properly
15/15 Code runs

Great job 100%

Score: 75/75

Total Score: 209/209

Assignment 6
Assigned: 2020-03-05
Due: 2020-03-11

Structuring

50 points total
0/10 Assignment 6 folder setup correctly
0/40 entropy.py, alpha.py, happy.txt, ones.py, nines.py, squares.py, luddy.py, wish.py (5 for each)
0/0 General notes to be aware of, not a deduction of points

Not submitted / incorrect

Score: 0/50

Problem 1

entropy.py

80 points total
0/5 makeProbability returns a list of floats
0/15 Code for makeProbability looks correct
0/5 entropy returns a float
0/15 Code for entropy looks correct
0/20 Test cases for makeProbability
0/20 Test cases for entropy

Not submitted / incorrect

Score: 0/80

Problem 2

alpha.py

60 points total
0/10 Path used is 'Assignment6/happy.txt', as shown in the assignment file
0/10 The dictionary is prepared. They don't hard code specific letters ('z':0). Allowed to fill the dictionary with default values for ALL letters.
0/26 Counts are correct
0/6 Printed out in order (all or nothing) without using built-in sort function
0/8 Each letter is printed on own line (all or nothing)

Not submitted / incorrect

Score: 0/60

Problem 3

ones.py

70 points total
0/20 Uses a loop to count the number of occurences 1s in a row.
0/10 Returns an integer
0/40 Test cases

Not submitted / incorrect

Score: 0/70

Problem 4

nines.py

70 points total
0/10 Returns a boolean

0/20 Reduces the integer to a single value
0/40 Test cases

Not submitted / incorrect

Score: 0/70

Problem 5

squares.py

65 points total
0/25 Correct number of stars for a given n
0/25 Number of stars on each level is correct
0/15 Function will work for any n greater than 1

Not submitted / incorrect

Score: 0/65

Problem 6

luddy.py

115 points total
0/5 Function area returns a boolean
0/20 function area looks correct
0/5 Function f returns a float or integer
0/10 Function f utilizes the formula correctly
0/15 Function f looks correct
0/30 Implements brute force correctly. (nested loops and updates the minimum in the loop).
0/10 Test cases for area
0/20 Test cases for f

Not submitted / incorrect

Score: 0/115

Problem 7

wish.py

70 points total
0/10 Returns a float
0/20 Function utilizes multiple loops (in order to find the average of the whole matrix)
0/20 Code looks reasonably correct
0/20 Test values

Not submitted / incorrect

Score: 0/70

Total Score: 0/580

Assignment 50

Assigned: 2020-03-05
Due: 2020-04-03

Structuring

10 points total
10/10 Assignment 50 folder setup correctly, including data.csv
0/0 General notes to be aware of, not a deduction of points

Nice job!

Score: 10/10

Problem 1

expect.py

290 points total
10/10 data.csv is in the repo
10/10 Relative path to the data file is correct
0/30 Plot is saved to a png file
0/30 Plot has a legend for the 6 data types, with separate colors and symbols
0/30 Plot has correct title
70/90 Proper data processing and analysis (including correct county and year)
60/90 Correct graph of life expectancy for male and female.

−30 No legend

−30 Plot is not saved as a png

−30 No title

−50 1)The y axis is not legible at all. 2)There should only be 2 colors for the lines, and dashes to denote what the line is showing. 3) None of the information displayed is correct

Score: 150/290

Total Score: 160/300

Assignment 60

Assigned: 2020-03-05
Due: 2020-04-01

Structuring

0 points total
0/0 General notes to be aware of, not a deduction of points

Score: 0/0

Problem 1

letters.py

25 points total
10/10 foot function passes test cases (0 if tail recursion isn't used)
15/15 foot function looks correct

Nice job!

Score: 25/25

Problem 2

travel.py

25 points total
5/5 distance.txt file has the data in the correct format to be read in
10/10 d function passes test cases
10/10 d function looks correct

Nice job!

Score: 25/25

Problem 3

sorting.py

20 points total
0/10 Code works correctly
3/10 Code looks correct

lambda function only returns first argument.

Score: 3/20

Problem 4

recur.py

30 points total
10/10 x and y test cases
20/20 x and y functions look correct

Nice job!

Score: 30/30

Total Score: 83/100