CSCI C200 Introduction to Computers and Programming

Spring 2020 Grade Report

Schneider, Jeremy

Computer Science
School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

June 19, 2020

Assignment 1

Assigned: 2020-05-12
Due: 2020-05-16

Structuring

30 points total
10/10 Assignment1 folder setup correctly
20/20 In folder: problems.py, myMath.py, thecost.py, myBall.py (5 each)
0/0 Any over comments that are in general

Great Work!

Score: 30/30

Problems

problems.py

65 points total
5/5 Code runs
20/20 Comments out the error line for each problem or fixed the error (preferablly commented the error out) (2 pts each)
40/40 Reasonable description for the error (4 pts each) (0/4 if exact copy of the error)

Nice job!

Score: 65/65

Math Head

myMath.py

50 points total
10/10 Code runs
20/20 Completed the math statements
20/20 Has 2 print statements (with the correct values)

Great!

Score: 50/50

The Cost

thecost.py

30 points total
10/10 Code runs
20/20 Completed the TODO statement

Great!

Score: 30/30

Bouncing Ball

myBall.py

55 points total
20/20 Code runs
10/10 Asks for input
15/15 Completes the calculation
10/10 Output matches pre-made situation

Nice job!

Assignment 2

Assigned: 2020-05-14

Due: 2020-05-20

___

Structuring

30 points total

10/10 Assignment2 folder setup correctly

20/20 In folder: myCalc.py, old.py, warping.py, condprac.py (5 each)

0/0 Any over comments that are in general

Nice job!

Score: 30/30

___

Equations

myCalc.py

115 points total

10/10 Code runs

61/63 Functions look correct (calculates (2 pts), returns (1pt))

38/42 Test Cases (2 pts per function)

The following functions failed test cases: parsecs2kilometer and kilometer2parsecs (−4 points).

(A quick explanation on test cases: The test cases that we used to test your functions were different from what was used in the base code. The goal of this was to ensure that your code works on lots of different numbers. We always encourage students to create their own test cases to find potential

errors in the code. This means that getting the correct output from the provided test cases does not guarantee a perfect score on the homework.)

For the parsecs2kilometer and kilometer2parsecs functions, you have the arithmetic reversed (−2 points).

Score: 109/115

Old

old.py

40 points total
10/10 Code runs
8/10 Created function 'themath' with 1 parameter and returns one value (38....)
10/10 Created function 'thecost' with 2 parameters and returns one value
10/10 Created function 'bouncing' with 1 parameter and returns the volume

In themath, you should have taken the square root of Themath before returning it (−2 points).

Score: 38/40

Warp Factor

warping.py

50 points total
10/10 Code runs
10/10 Returns a warp speed for all 10 warp factors
8/10 Conditions are correct (lower bounds)

20/20 Test values

For warp factor 0, the lower bound should be less than 1. For warp factor 1, the lower bound should be if the value is greater than or equal to 1. (−2 points)

Score: 48/50

Conditional Practice

condprac.py

52 points total
10/10 Code runs
8/8 Outputs match at the bottom for fun1 and fun1reworked
9/9 Outputs match at the bottom for fun2 and fun2reworked
10/10 fun1: Adjusted the conditions (2 pts each) to use ifs (no elses or elif are allowed). If condition contains elif or else, 0 point for condition
0/5 fun2: boolean condition looks equivalent to original statement
10/10 Output matches pre-made situation

For fun2_reworked, you were supposed to write a boolean condition similar to what was in fun2, not with if/else statements (−5 points).

Score: 47/52

Total Score: 272/287

Assignment 3

Assigned: 2020-05-18

Due: 2020-05-23

Structuring

35 points total

10/10 Assignment3 folder setup correctly

25/25 In folder: looping.py, stringStuff.py, whiling.py, whileString.py, complexing.py (5 each)

0/0 Any over comments that are in general

Nice job!

Score: 35/35

Looping FOR

looping.py

73 points total

10/10 Code runs off the bat

42/42 Each function uses a for loop, returns at the correct place (not too early), and body looks correct (2 pts for each 3 sub parts), 6pts per function (7 Functions)

21/21 Test Cases (1 pts per test per function (3pts for each function))

Nice job!

Score: 73/73

String Stuff

stringStuff.py

76 points total
10/10 Code runs
20/20 Each function uses the input parameter(s) correctly, has a loop, uses the loop correctly, returns, and looks correct (5pts per function)
14/14 Test Cases for Palindrome (14 pts, 2pts per test case)
24/24 Test cases for getCount and getIndex (4pts per test case for each function, 12 pts for each function)
8/8 Test cases for areEqual (2 pts for each test case)

Nice job!

Score: 76/76

Looping WHILE

whiling.py

87 points total
10/10 Code runs off the bat
56/56 Each function uses a while loop, condition is proper, returns at the correct place (not too early), and body looks correct (2 pts for each 4 sub parts), 8 pts per function (7 Functions)
21/21 Test Cases (1 pts per test per function (3pts for each function))

Nice job!

Score: 87/87

String Stuff WHILE

whileString.py

80 points total

10/10 Code runs

24/24 Each function uses the input parameter(s) correctly, has a loop, loop condition proper, uses the loop correctly, returns, and looks correct (6pts per function)

14/14 Test Cases for Palindrome (14 pts, 2pts per test case)

24/24 Test cases for getCount and getIndex (4pts per test case for each function, 12 pts for each function)

8/8 Test cases for areEqual (2 pts for each test case)

Nice job!

Score: 80/80

Complexing

complexing.py

66 points total

10/10 Code runs

24/24 Each function uses the input parameter(s) correctly, has a loop, loop condition proper, uses the loop correctly, returns, and looks correct (6pts per function)

32/32 Test Cases (8 pts for each function, 2pts per test case, 4 functions)

Nice job!

Score: 66/66

Total Score: 417/417

Assignment 4
Assigned: 2020-05-21
Due: 2020-05-27

Structuring

45 points total
10/10 Assignment4 folder setup correctly
35/35 In folder: functionsAsRec.py, farTest.py, aSort.py, listAndDict.py, lodTest.py, myRec.py, recTest.py (5 each)
0/0 Any over comments that are in general

Nice job!

Score: 45/45

Recursive Functions

functionsAsRec.py

94 points total
10/10 Code runs off the bat
42/42 Function has a base case, inductive step, returns the correct information (6pts per function (2 pts per part)) (7 Functions)
12/21 Test cases: 3 pts per function (7 functions)
0/21 EXTRA CREDIT: fibonnaci (10 pts at most). Memoization done correctly

Your functions for lengthOfString, removePos, and fibonnaci failed test cases (−9 points).

Score: 64/94

Recursive Functions TEST

farTest.py

38 points total
10/10 Code runs off the bat
28/28 2 test cases for each function (2 pts per test case made) (7 functions)

Nice job!

Score: 38/38

Sorting Algorithms

aSort.py

44 points total
10/10 Code runs
30/30 Conversion looks reasonably correct, doesn't heavily deviate from psuedo-code
4/4 Test Cases (2pts each)

Nice job!

Score: 44/44

Lists and Dictionaries

listAndDict.py

72 points total
10/10 Code runs off the bat
25/25 letterFrequency: Returns a dictionary, loops correctly (nested) and is not hard coded
25/25 primeList: Not hard coding all the primes needed, Function loops correctly, returns a list.

12/12 Test cases: 3 tests per function, 2pts each

Nice work

Score: 72/72

Recursion and Closed Form

myRec.py

96 points total
10/10 Code runs
0/15 Functions 1, 2, 3, 4, 5 (3pts each) has a base case, inductive step, passes back correct information
15/15 Closed Functions 1, 2, 5 (3 pts each): Does not do recursion, uses equation
22/56 Test Cases (7 each)

function1 failed 6 of 7 test cases (−6 points). func2 failed all test cases (−7 points). func3, func4, func5 failed all test cases (−21 points).

function1, func2, func3, func4, and func5 should have been recursive but they were not (−15 points).

Score: 47/96

Recursion and Closed Form test case

recTest.py

90 points total
10/10 Code runs

80/80 For loop properly displays the first 10 values for each function (handling different starts)

Nice job!

Score: 90/90

Total Score: 400/479

Assignment 5
Assigned: 2020-05-25
Due: 2020-05-29

Structuring

30 points total
10/10 Assignment5 folder setup correctly
20/20 In folder: buiitinCode.py, formatRunning.py, decode.py, code.txt (5 each)
0/0 Any over comments that are in general

Nice job!

Score: 30/30

List and Strings

builtinCode.py

98 points total
10/10 Code runs off the bat
32/33 Each function is a couple of lines or less and returns (3 pts per function, 11 functions)
22/22 Test cases wrote by students (2 per function, 1 pt per test)
30/33 Test cases: 3 pts per function (11 functions)

In handlesValueIssue in myInsertList, place should be the first variable passed into insert and new should be the second. (−1 point)

getIndex failed all test cases (−3 points).

String Format

formatRunning.py

85 points total
10/10 Code runs off the bat
15/15 String1: $\{\} - \{\}$
15/15 String2: $\{:> 10\}$
15/15 String3: $\{: 06.2f\}$
15/15 String4: $Does\{\} == \{\}? -> \{\}$
15/15 String5: $\{: {}^{1}0\}$

Nice job!

Score: 85/85

Decoding ASCII

decode.py

75 points total
15/15 Code runs
15/20 read-file: File returns a list of strings (no newlines and spaces). (-5 if they don't close the file (it is alright if they use a context manager))
20/20 convert: Loops through a list of strings to produce a string (not in ASCII)
20/20 write-file: Successfully writes the file. (-5 if they don't close the file (it is alright if they use a context manager))
30/0 extraCredit: (At most 30). Successfully returns a string (15) and replaced the string 'Camisa' with their name (15). Points for if successful in next section.

You should remove new line characters when you are processing the text in read_file (−5 points).

Message Text File

message.txt

10 points total
10/10 File decoded correctly
5/0 Extra Credit: 'Camisa' string replaced with their name (5)

Nice job!

Score: 15/10

Total Score: 324/298

Assignment 6

Assigned: 2020-06-01

Due: 2020-06-06

Structuring

35 points total

10/10 Assignment6 folder setup correctly

25/25 In folder: Person.py, BloodBank.py, main.py, makeMeMove.py, inherits.py (5 each)

0/0 Any over comments that are in general

Nice job!

Score: 35/35

I need some blood P1

Person.py

57 points total

15/15 Constructor makes instance variables for name, address, birthday, type, bloodInBody (names don't have to match) (3pts each)

5/5 Instance variable 'birthday' uses datetime.strptime

4/4 getAge: Calculates the years. They can either do it by subtracting years or calculating the days between and divide by 365

2/4 isInjured: Returns a boolean 'self.bloodInBody ¡ 5200 or self.bloodInBody ¿ 5600' (If ¡= or ¿= is used, subtract 1 for each used). Doesn't have to be the exact line

4/4 getBloodCount: Returns the instance variable representing blood in body

3/3 transferBlood (P1): Return -1 when count ¿ 0 and the transfer won't be over 5600

3/3 transferBlood (P2): Return -1 when count ¡ 0 and the transfer won't

be under 5200

3/3 transferBlood (P3): Return 1 after adding / subtracting blood from bloodInBody. (Note: count is negative or positive)

3/3 resetBloodCount: sets bloodInBody to be 5500

3/3 str: name   age   bloodtype   bloodcount ml (ml is at end)

10/10 Person file runs (doesn't matter output)

It was not necessary to use <= or >= in isInjured (−2 points).

Score: 55/57

I need some blood P2

BloodBank.py

59 points total

3/3 Completed DONATE-TO dictionary

3/3 Completed DONATE-FROM dictionary

16/16 constructor: instance variables for name, location, users (empty list) and bankBlood (blood types start at 0) (4 pts per variable)

3/3 totalUnitsOfBlood (p1): If no blood type is provided, returns total across all dictionary

3/3 totalUnitsOfBlood (p2): If blood type is provided, accesses dictionary and returns the number

3/3 findPeople: Returns a list of people that are able to donate to (compatibility)

3/3 addPerson: Adds person to list of users

3/3 giveBlood (p1): If there is not enough blood of that type, return -1

3/3 giveBlood (p2): If there is enough blood for that type, try transferring. If the transfer isn't possible, return -2. (-1 points if they remove the quantity before verifying it could transfer

3/3 giveBlood (p3): If there is enough blood for that type and the transferBlood returns 0, subtract the quantity from the bloodbank.

3/3 receiveBlood: attempts to transfer -quantity. If result is -1, return -1. If result is not -1, add quantity to bloodbank.

3/3 registeredUsers: Returns length of users
10/10 Code runs off the bat

Nice job!

Score: 59/59

I need some blood P3

main.py

80 points total
10/10 Code runs off the bat
20/20 listOfPeople: Function returns a list of people instances.
50/50 Output looks reasonably correct

Nice job!

Score: 80/80

Pygame Practice

makeMeMove.py

130 points total
10/10 Code runs off the bat
25/25 Converted the list to a pygame.Rect object
30/30 Velocity is randomly generated (at some point). Velocity is random between 1 to 5 (inclusive). x and y direction (15 pts each)
15/15 Bounces off the wall successfully (Not required to change speed).
30/0 Extra Credit: Generates a new velocity (randomly) after detecting the wall (+30)
40/40 Changes color to a DARK GREEN (10), DARK YELLOW (10), LIGHT BLUE (10), and ORANGE (10). Dark and light means they aren't just the color green or blue. ex. (0, 255, 0) is not allowed for dark green.

10/10 Code runs off the bat

Nice job!

Randomized velocity added when the cube bounces off of a wall (+30 points).

Score: 160/130

Inheritance

inherits.py

140 points total
10/10 Code runs. Does not matter the output
0/25 Superclass Assignment/Assignments (either name is fine)
25/25 Created a couple of sub-classes, that utilizes inheritance
25/25 Reasonable instance variables are made for the superclass and sub-classes
25/25 Reasonable instance methods/functiosn are made for the superclass and subclasses
15/15 Comments explaining thought process
15/15 Some print statements showing off.

Your superclass should be Assignment or Assignments per the assignment file (−25 points).

Score: 115/140

Total Score: 504/501

Assignment 7

Assigned: 2020-06-04

Due: 2020-06-11

Structuring

35 points total

10/10 Assignment7 folder setup correctly

25/25 In folder: bestgraph.py, bgTest.py, roman.py, balancing.py, people.py (5 each)

0/0 Any over comments that are in general

Good Job!

Score: 35/35

Best Graph

bestgraph.py

85 points total

10/10 File runs off the bat. Does not need to produce output

15/15 addedge: Given a tuple, return a -1 if either node is not in the graph. Return a -1 if the edge already exists. Otherwise, add pair to edge list and return 1 (5 pts per part)

15/15 addnode: Given a node, return -1 if the node exists in the graph already (5pts). Otherwise, add to the node list AND the edge list and return 1 (4 pts adding to node list, 4 pts adding to edge list, 2 for return 1)

15/15 delnode: Return -1 if the node doesn't exist (5pts). Otherwise, remove from nodes list (2pts) and node edge list from edge list (2pts). Also remove the node from other nodes in the edge list (going through the dictionary) (4pts) and return 1 (2 pts)

15/15 deledge: Takes in a tuple or 2 parameters (depends on implementation) (4 pts). Returns -1 if edge does not exist (4pts). Checks to make

sure the nodes exist in the graph (3pts). If able to remove, removes edge and return 1 (4 pts)

15/15 adjacencyMatrix: Returns a list of lists. (15pts). Does not store this matrix in an instance variable (-2 pts if done)

Great!

Score: 85/85

Best Graph Test

bgTest.py

100 points total
10/10 Code runs off the bat.
15/15 Testing adding original edges (5pts per test)
20/20 Testing add node (5pts per test)
25/25 Testing add edge (5 pts per test)
10/10 Testing delete nodes (5 pts per test)
10/10 Wrote 2 tests for Adjacency Matrix (does not need to be fancy) just reasonable (5pts each)
10/10 Wrote 2 tests for Adjacency Matrix (does not need to be fancy) just reasonable (5pts each)

Great!

Score: 100/100

Roman

roman.py

83 points total
30/30 Created a function that takes in one value and returns a string. At most 5 points if there are over 22 pre-made roman numerals (hard coding)

33/33 Uses a for loop to display 99 numbers and the corresponding roman numeral
0/20 Successfully splits, wtih each line starting on a multiple of 5 (first line starts with 1)

Only 3 numbers were shown. Needed to display only 5 numbers on each line.

Score: 63/83

Balancing Parentheses

balancing.py

100 points total
15/15 Uses a Stack class (by import or by putting it at the top of the file)
15/15 Handles ([(5 pts each)
15/15 Handles)] (5 pts each)
10/10 Successfully loops through (can end early if detecting not balanced)
45/45 Passes test cases (5 pts each)

Good Work!

Score: 100/100

People

people.py

125 points total
10/10 Code runs off the bat
25/25 People class is the superclass
20/20 Student and Instructor class are subclasses (both inherit correctly (5 pts each) and call the super class (5pts each))

9/9 Superclass has: name, year of birth, and a list of friends instance variables (3 pts each).

9/9 Students have a major, credits earned, student ID. (3 pts each). Subtract 1 each time they initialize a instance variable that should be in the superclass.

6/6 Instructors have a salary and office hours (3pts each). Subtract 1 each time they initialize a instance variable that should be in the superclass.

10/10 Created a reasonable amount of tests to show off what was created.

18/18 Constructors have the correct number of parameters (6 pts per class)

18/18 String methods present a reasonable amount of information (6 pts per class)

Good Work!

Score: 125/125

Total Score: 508/528

Assignment 8

Assigned: 2020-06-10

Due: 2020-06-17

Structuring

35 points total

10/10 Assignment8 folder setup correctly

25/25 In folder: triangle.py, mm.py, dfssearch.py, shishkebab.py, myGraph.py (5 each)

0/0 Any over comments that are in general

Nice job!

Score: 35/35

Sierpinski Triangle

triangle.py

67 points total

15/15 Function triangle: returns a list of 3 points representing the top left corner of the triangle, top right corner of the triangle and the bottom point. This function does not make it smaller, just uses the point and width to make the triangle

7/7 Function s (part 1): draws triangle based on the point based in (so modification of the point)

15/15 Function s (part 2): Recursive call 1 calls the top left corner with half of the width

15/15 Function s (part 3): Recursive call 1 calls the top middle of the large triange with half of the width

15/15 Function s (part 4): Recursive call 1 calls the middle point of the left side of the large triange with half of the width

Nice job!

Score: 67/67

Sierpinski Triangle Extra Credit

ecTriangle.py

0 points total

0/0 Modified drawtriangle to take in a color (5pts) and utilizes a new color (5pts) (each is all or nothing)

0/0 In the s function, the top left triangle makes a random shade of red (ensure that the tuple is not all random) (4 pts)

0/0 In the s function, the top right triangle makes a random shade of green (ensure that the tuple is not all random) (4 pts)

0/0 In the s function, the bottom triangle makes a random shade of blue (ensure that the tuple is not all random) (4 pts)

0/0 In the s function, each color is passed recursively (2 pts for each s function call)

No code in file.

Score: 0/0

Matrices

mm.py

72 points total

15/15 Function mm: Multiplies 2 matrices together.

15/15 Function sm: Returns each elements multiplied by the constant

15/15 Function tp: Transposes the matrix

15/15 Function addm: adds 2 matrices of the same size together

12/12 Added 3 different test cases for each function (1 pts per function per test)

Nice job!

<div align="right">

Score: 72/72

</div>

DFS Search of Directories

dfssearch.py

60 points total

3/6 Successfully imports isfile, isdir from os.path (3 pts each)

10/10 Command Line Arguments: imports sys (3 pts) and passes in the 2nd arguemnt 'sys.argv[1]' into dfsFiles (7 pts)

14/14 getDirs: Replaces first TODO (for loop) with listdir (7 pts) and the second TODO with isdir (if statement) (7 pts)

5/5 dfsFiles (part 1): Has a call to get the directories (getDirs) from the passes in parameter

5/5 dfsFiles (part 2): The loop is based on the amount of items in the result from getDirs

5/5 dfsFiles (part 3): remove something from the list of directories

5/5 dfsFiles (part 4): prints the removed item and gets all the directories of the one just printed

10/10 dfsFiles (part 5): Adds to the beginning of the list that is being checked in the while loop

You are missing isfile in your imports (−3 points).

<div align="right">

Score: 57/60

</div>

Lamdba, map, filter, list comprehension

shishkebab.py

130 points total

10/10 Function q1: List comprehension solves the problem. Has a print statement (1 pt for print)

10/10 Function q2: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

10/10 Function q3: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

10/10 Function q4: List comprehension solves the problem. Has a print statement (1 pt for print)

10/10 Function q5: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

10/10 Function q6: List comprehension solves the problem. Has a print statement (1 pt for print)

10/10 Function q7: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

10/10 Function q8: List comprehension solves the problem. Has a print statement (1 pt for print)

10/10 Function q9: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

5/10 Function q10: List comprehension solves the problem. Has a print statement (1 pt for print)

5/10 Function q11: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

5/10 Function q12: List comprehension solves the problem. Has a print statement (1 pt for print)

5/10 Function q13: Lambda/map/filter solves the problem. Has a print statement (1 pt for print)

Functions q10, q11, q12, and q13 should be lists of strings, not lists of lists. (−20 points).

Score: 110/130

Messing with matplotlib

myGraph.py

80 points total
30/30 Creates 2 dots on the graph as (-5, 60) and (5, 60) (15 each)

40/40 Creates 1 smooth curved line on the graph (x ** 2) and plotting it with '-g' (-10 if the line does not look smooth)
10/10 Successfully draws the plot

Nice job!

Score: 80/80

Total Score: 421/444