# CSCI C200 Introduction to Computers and Programming

# Fall 2019 Grade Report

---

Graber, Zachary

Computer Science
School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

---

December 20, 2019

# Assignment 1

Assigned: September 4, 2019
Due: September 11, 2019

---

## Problem 1

**windchill.py**

50 points total
10/10 points for correct Assignment1 folder setup
10/10 points for correct module name
10/10 points for proper variable names in the calculation `(T and V)`
20/20 points for proper calculation

**Score**: 50/50

---

## Problem 2

**creditcard.py**

50 points total
10/10 points for correct Assignment1 folder setup
10/10 points for correct module name
10/10 points for proper variable names in the calculation `(APR, C, P, i)`
20/20 points for proper calculation

**Score**: 50/50

---

**Total Score**: 100/100

# Assignment 2

Assigned: September 12, 2019
Due: September 18, 2019

---

## Problem 1

**mayhem.py**

195 points total

120/120 points for functions [10 points each]:
`speed`, `distance`, `time`, `hours_to_min`, `min_to_sec`, `feet_to_mile`, `miles_to_kilometers`, `kilometers_to_miles`, `miles_to_feet`, `degrees_to_radians`, `parsecs_to_kilometers`, and `lightyears_to_parsecs`.

75/75 points for functions [15 points each]:
`side_length_triangle`, `celsius_to_fahrenheit`, `fahrenheit_to_celsius`, `kelvin_to_fahrenheit`, and `percent_change`.

*Perfect!*

**Score**: 195/195

---

## Problem 2

**2019tax.py**

60 points total
25/25 points for proper implementation of the `unmarriedTax` function.
25/25 points for proper implementation of the `marriedTax` function.
10/10 points for answering observational question.

*Perfect!*

**Score**: 60/60

---

## Problem 3

**lestat.py**

80 points total
40/40 points for implementation of the `receiveFrom` function with correct output.
40/40 points for implementation of the `donateTo` function with correct output.

*Perfect!*

**Score**: 80/80

---

## Problem 4

**coolline.py**

35 points total
10/10 points for changing the title of the graph.
25/25 points for adding the new function to the plot.

*Perfect!*

**Score**: 35/35

---

**Total Score**: 370/370

## Assignment 3

Assigned: September 19, 2019
Due: September 25, 2019

---

### Problem 1

**funwithfunctions.py**

135 points total
135/135 points for functions [15 points each]:

*Great job!*

**Score**: 135/135

---

### Problem 3

**qc1.py**

50 points total
15/15 points for printing a message indicating `complex` or `not complex`.
35/35 points for a correct implementation of the `q` function with appropriate return structure for quadratic solutions.

*Good work!*

**Score**: 50/50

---

### Problem 4

**if.py**

75 points total
75/75 points for conditional statements correctly re-written [15 points for each group]:

*Good work!*

**Score**: 75/75

---

## Problem 5

**precmetal.py**

75 points total
30/30 points for proper implementation of the `preciousMetalToDollars` function.
45/45 points for proper implementation of the `purchase` function.

*Great work!*

**Score**: 75/75

---

## Problem 6

**myclock.py**

25 points total
10/10 points for changing title.
15/15 points for changing font.

*Great job!*

**Score**: 25/25

---

**Total Score**: 360/360

## Assignment 4

Assigned: September 25, 2019
Due: October 2, 2019

---

### Problem 1

**funtriangle.py**

45 points total
0/45 points for correct triangle output [15 points each]:

*Great job 100%*

**Score**: 45/45

---

### Problem 2

**makeitrain.py**

40 points total
0/30 points for correct implementation of `dollars` function.
0/10 point for appropriate return values and structure.

*Great job 100%*

**Score**: 40/40

---

### Problem 3

**donor.py**

60 points total
0/30 points for proper implementation of `red_blood_compatibility` function with appropriate return values.
0/30 points for proper implementation of `transfusion` function with appropriate return values.

*Great job 100%*

**Score**: 60/60

---

## Problem 4

**palindrome.py**

40 points total
0/40 points for correct implementation of `palindrome` function.

*Great job 100%*

**Score**: 40/40

---

## Problem 5

**roman.py**

50 points total
0/50 points for correct implementation of `roman` function.

*Great job 100%*

**Score**: 50/50

---

## Problem 6

**moreloops.py**

115 points total
0/75 points for correct implementation [15 points each] of
`maxFor`, `maxWhile`, `minFor`, `myReplace` functions, `StringConcat`

0/40 points for correct implementation [20 points each] of
`RemoveEvens`, `sumOdd`

*−2 maxFor, maxWhile, minFor should return empty list when provided empty list as input*

**Score**: 113/115

---

## Problem 7

**farm.py**

50 points total
0/50 points for correct implementation of `roman` function.

*Great job 100%*

**Score**: 50/50

---

**Total Score**: 398/400

## Assignment 5

Assigned: October 3, 2019
Due: October 9, 2019

---

### Problem 1

**entropy.py**

60 points total
0/30 points for correct implementation of the `makeProbability` function:
0/30 points for correct implementation of the `entropy` function:

*good job.*

**Score**: 60/60

---

### Problem 2

**magic.py**

60 points total
0/40 points for correct encantation [8 points each]
0/10 points for correct order of encantation
0/10 points for correct return value

*good job.*

**Score**: 60/60

---

### Problem 3

**ones.py**

40 points total
0/40 points for correct implementation of the `lr` function

*good job.*

**Score**: 40/40

---

## Problem 4

**nines.py**

40 points total
0/40 points for correct implementation of the `div_9` function

*good job.*

**Score**: 40/40

---

## Problem 5

**squares.py**

40 points total
0/40 points for correct implementation of the `sq` function

*good job.*

**Score**: 40/40

---

## Problem 6

**luddy.py**

70 points total
0/15 points for correct implementation of the `area` function
0/15 points for correct implementation of the `f` function
0/20 points for brute force solution
0/20 points for numpy solution

*good job.*

**Score**: 70/70

---

## Problem 7

**wish.py**

50 points total
0/50 points for correct implementation of the `is_magic` function

*good job.*

**Score**: 50/50

---

**Total Score**: 360/360

## Assignment 6

Assigned: October 11, 2019
Due: October 14, 2019

---

### Problem 1

**alpha.py**

80 points total
10/10 points for correctly opening and reading the file from the correct location
20/20 points for correctly reading the file contents
10/10 points for setting up the dictionary
30/30 points for counting lowercase letters
10/10 points for properly returning the dictionary of counted letters

*Good Work!*

**Score**: 80/80

---

**Total Score**: 70/80

## Assignment 7

Assigned: October 23, 2019
Due: October 30, 2019

---

### Problem 1

**recpractice.py**

190 points total
150/150 points for correct implementation of the ten (recursive and non-recursive) functions [15 points each]
25/25 points for including a (single) for-loop to show the first ten values of each function
15/15 points for answering critical thinking questions

*Perfect! Love the comments.*

**Score**: 190/190

---

### Problem 2

**minime.py**

95 points total
95/95 points for correct implementation of the six min functions [15 points all bu MIN, which is 20]

*Great work*

**Score**: 95/95

---

### Problem 3

**twoMax.py**

35 points total
40/40 points for proper implementation of `twoMax` function.

*Nice work*

**Score**: 40/40

---

## Problem 4

**isogram.py**

30 points total
30/30 points for correct implementation of `is_isogram` function.

*Very nice*

**Score**: 30/30

---

## Problem 5

**hexagram.py**

35 points total
35/35 points for correct implementation of `hex_dec` function.

*Good job*

**Score**: 35/35

---

## Problem 6

**doctor.py**

60 points total
30/30 points for correct implementation of appendicitis prediction logic
10/10 points for correct input functionality
10/10 points for correct and informative output
10/10 points for meaningful and informative comments

*Great*

**Score**: 60/60

---

## Problem 7

**astronomy.py and stellar.py**

50 points total
20/20 points for completing the `astronomy.py` module.
30/30 points for completing the functions in the `stellar.py` module.

*Wonderful*

**Score**: 50/50

---

**Total Score**: 500/500

## Assignment 8

Assigned: November 1, 2019
Due: November 6, 2019

---

### Problem 1

**fignewton.py**

50 points total
20/20 points for correct implementation and integration of user input: function and initial estimate [10 points each].
30/30 points for correct implementation and integration of user input: threshold and iteration bound [15 points each].

*Nice work*

**Score**: 50/50

---

### Problem 2

**mybisect.py**

50 points total
15/15 points for correct implementation of the `sign` function.
35/35 points for correct implementation of the `bisect` function.

*Nice work*

**Score**: 50/50

---

### Problem 3

**game1.py**

50 points total
50/50 points for proper implementation of color changing square.

*Nice work*

**Score**: 50/50

---

## Problem 4

**secant.py**

50 points total
50/50 points for correct implementation of `secant` function.

*Nice work*

**Score**: 50/50

---

## Problem 5

**easycalc.py**

50 points total
50/50 points for correct implementation of `simpson` function.

*Nice work*

**Score**: 50/50

---

## Problem 6

**rec.py**

200 points total
200/200 points for correct implementation of `even`, `odd`, `b`, `btr`, `bm`, `gg`, `gtr`, `gm` functions [25 points each].

*Nice work*

**Score**: 200/200

---

**Total Score**: 450/450

## Assignment 9

Assigned: November 7, 2019
Due: November 13, 2019

---

### Problem 1

**randomwalk.py**

60 points total
0/60 points for correct implementation of the `step` function.

*Awesome job!*

**Score**: 60/60

---

### Problem 2

**mymap.py**

50 points total
25/25 points for adding the geolocation of IU's Musical Arts Center.
25/25 points for adding it to the list of points and plotting it.

*Great job!*

**Score**: 50/50

---

### Problem 3

**complex.py**

90 points total
0/40 points for correct implementation of the `__sub__` and `__truediv__` functions [20 points each].
0/50 points for correct implementation of the `molulus` and `polar` functions [25 points each].

*Amazing job! Keep up the great work!*

**Score**: 90/90

---

**Total Score**: 200/200

## Assignment 10

Assigned: November 14, 2019
Due: November 20, 2019

---

### Problem 1

**haversine.py**

60 points total
45/45 points for correct implementation of the `hd` function.
15/15 points for answering the analysis questions.

*Nice job!*

**Score**: 60/60

---

### Problem 2

**lineclass.py**

50 points total
25/25 points for correct implementation of `__init__` class function.
25/25 points for correct implementation of `__mul__` class function.

*Great!*

**Score**: 50/50

---

### Problem 3

**weird.py**

60 points total
25/25 points for correct implementation of `f` function.
25/25 points for correct implementation of `g` function.
10/10 points for answering the analysis question.

*Nice!*

**Score**: 60/60

---

## Problem 4

**matrix.py**

120 points total
120/120 points for correct implementation of matrix functions [30 points each].

*Good job!*

**Score**: 120/120

---

## Problem 5

**correlation.py**

80 points total
20/20 points for correct implementation of the `mean` function.
20/20 points for correct implementation of the `sd` function.
20/20 points for correct function calls to calculate the correlation coefficient.
20/20 points for correctly producing a plot named `stock.png`.

*Great!*

**Score**: 80/80

---

**Total Score**: 370/370

## Processing Real Data

Assigned: November 7, 2019
Due: November 13, 2019

---

### Part 1

#### Deliverables

10 points total
10/10 points for expect.py and graph state_county.png

*Nice!*

**Score**: 10/10

---

### Part 2

#### Functionality

120 points total
0/120 points for correct data processing and analysis.

*Great!*

**Score**: 120/120

---

### Part 3

#### Graph

170 points total
25/25 points for correct title.
25/25 points for correct legend.
120/120 points for correct graph of life expectancy for male and female.

*Nice job!*

**Score**: 170/170

---

**Total Score**: 300/300

## Assignment 10

Assigned: November 14, 2019
Due: November 20, 2019

---

### Problem 1

**haversine.py**

60 points total
45/45 points for correct implementation of the `hd` function.
15/15 points for answering the analysis questions.

*Nice job!*

**Score**: 60/60

---

### Problem 2

**lineclass.py**

50 points total
25/25 points for correct implementation of `__init__` class function.
25/25 points for correct implementation of `__mul__` class function.

*Great!*

**Score**: 50/50

---

### Problem 3

**weird.py**

60 points total
25/25 points for correct implementation of `f` function.
25/25 points for correct implementation of `g` function.
10/10 points for answering the analysis question.

*Nice!*

**Score**: 60/60

---

## Problem 4

**matrix.py**

120 points total
120/120 points for correct implementation of matrix functions [30 points each].

*Good job!*

**Score**: 120/120

---

## Problem 5

**correlation.py**

80 points total
20/20 points for correct implementation of the `mean` function.
20/20 points for correct implementation of the `sd` function.
20/20 points for correct function calls to calculate the correlation coefficient.
20/20 points for correctly producing a plot named `stock.png`.

*Great!*

**Score**: 80/80

---

**Total Score**: 370/370