

# 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 2, 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 (preferably 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!*

**Score:** 55/55

---

**Total Score:** 230/230

## 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**