

**Ever so clearly print your email id:**

**Ever so clearly print your name:**

**Pledge:**

### Notices

- Based on your past educational achievements, I expect you to do well on this test.
- Answer the questions in any order that you want.

### Test rules

- Before you leave the room, check that you uploaded all of your solutions. Do not ask afterwards whether you can submit a forgotten solution.
- This pledged exam is closed notes. The only device you may access during the test is your laptop.
- Any cheating can result in failing the class and the incident being referred to the Honor Committee.
- Do not access class examples artifacts, web solutions, or your own past assignments during the test; that is, the only code you may access or view are ones that you develop for this test.
- The only windows allowed on your laptop are PyCharm and a single browser with tabs reachable from class website.

### PyCharm

- PyCharm can be used for developing the modules to be submitted. It **cannot be used** for the short answer questions of Part 1.

### Program

- Modules should follow class programming practices; e.g., whitespace, identifier naming, and commenting if you think it is needed, etc.
- Whether a program runs is important.
- Comment out or delete all debugging `print()` statements before submitting.

THIS PAGE IS ALMOST BLANK

**Part 1: short answers**

1. (26 points) Briefly explain the purpose of the following Python programming elements.

whitespace	
non-header comment	
literal	
variable	
input()	
print()	
/ operator	
// operator	
% operator	
import	
style rule	
assignment operator	
loop	

2. (8 points) Suppose the code segment has just been executed.

```
s = "Happy"  
s.lower()
```

What are the values printed by the following statements? If your answer is alphabetic, underline uppercase letters.

<code>print( s [ 1 ] )</code>	
<code>print( s[ 1 : 3 ] )</code>	
<code>print( s.find( "p" ) )</code>	
<code>print( s )</code>	

3. (2 points) Write a for loop that prints the characters of a string `t`, one per line.

4. (2 points) Write a for loop that prints the elements of a list `x`, one per line.

5. (2 points) Write a for loop that prints the integers from 1 to 100, one per line.

**Part 2: Programming**

6. (12 points) Complete program *car.py* that prints out the string *car* in uppercase letters. A run of the program should always produce the following output.

```
CAR
```

7. (12 points) Complete program *road\_trip.py* that separately prompts for two integer values regarding a car trip: the number of miles driven and the number of hours taken to drive those miles. The program computes and prints the *decimal* miles per hour for the trip. There should be no other output. Two possible program runs follow.

```
Enter number of miles driven on trip: 325
Enter number of hours used to drive those miles: 5
65.0
```

```
Enter number of miles driven on trip: 114
Enter number of hours used to drive those miles: 4
28.5
```

8. (12 points) Complete program *yze.py*. The program has one prompt to get a list of integer numbers. The program computes and prints on separate lines: number of inputs, minimum input, maximum input, and *integer* average of the inputs. There should be no other output. Two possible program runs follow.

```
Enter integers: 3 1 4 1 5 9 2
7
1
9
3
```

```
Enter integers: 2 -8 4 -16 8 32
6
-16
32
3
```

9. (12 points) Complete program *flipper.py*. The program separately prompts and gets three inputs from its user. The first input is to be used as a seed value for the Python random number generator; the second input is a list of strings; the third input is the integer number of selections to be made on the strings. For the indicated number of selections, the program randomly picks from one of the strings and prints it out on a separate line. There should be no other output. Two possible program runs follow.

```
Enter seed: value
Enter choices: yes no
Enter flips: 5
no
yes
yes
no
yes
```

```

Enter seed: cs1112
Enter choices: sunny snowy foggy rainy
Enter flips: 4
snowy
foggy
rainy
foggy

```

10. (12 points) Complete program *wet.py*. The program analyzes the CS 1112 dataset *raindrops.csv*. Be aware the contents of the dataset will change during the grading of the exam.

The dataset does not have a header row. Each dataset row has two values. A row represents a response for the question *what do you call it when rain falls while the sun is shining*. The first value in a row is a name for the meteorological phenomenon. The second value in a row is the country of the person giving that name.

The program separately prompts and gets a name of the phenomenon and a country. The program displays on separate lines, the number of dataset entries that call the phenomenon by the first supplied input, and the number of entries whose country is the second supplied input. There should be no other output. Two possible program runs follow.

```

Enter terminology name: Sunshower
Enter country: Honduras
41
1

```

```

Enter terminology name: Liquid sun
Enter country: USA
4
75

```

Below is the partial listing of the *raindrops.csv* dataset.

```

Sunshower,          USA
Sunshower,          USA
No term,            USA
Monkey's wedding,  USA
...
Rainbow weather,   USA
No term,            USA
No term,            USA
Sunshower,          USA
Sunshower,          USA
No term,            USA
Liquid sun,         USA
Sunshower,          Canada
Sunshower,          USA
No term,            USA
Sunshower,          USA
Sunshower,          USA
Sunshower,          USA

```