

**Clearly print your computing id:**

**Clearly print your name:**

**Pledge:**

**Test rules**

- *Check that you uploaded your six solutions. Do not ask later to submit a forgotten solution.*
- *The only device you may access during the exam is your laptop. The only open windows allowed are PyCharm and a browser with tabs linked from the class website.*
- *The only code you may access are ones that you develop for this test.*
- *You may not access class notes, epistles, examples, artifacts, solutions on the web, or your own past assignments during the test.*
- *PyCharm cannot be used for the short answer questions of Part I.*
- *Code should follow class programming practices; e.g., whitespace, identifier naming, etc.*
- *Whether code is testable is important. Comment out or delete all debugging print() statements before submitting.*
- *Print only what is requested and nothing else.*

**Short answer questions answers**

- |           |           |
|-----------|-----------|
| 1. _____  | 11. _____ |
| 2. _____  | 12. _____ |
| 3. _____  | 13. _____ |
| 4. _____  | 14. _____ |
| 5. _____  | 15. _____ |
| 6. _____  | 16. _____ |
| 7. _____  | 17. _____ |
| 8. _____  | 18. _____ |
| 9. _____  | 19. _____ |
| 10. _____ | 20. _____ |

**THIS PAGE ALMOST BLANK**

**Part I. Short answers (2 points each)**

**Important: When indicating values: If the value is integer, do not use a decimal point. If the value is decimal, use a decimal point. If the value is a string, enclose it within quotes.**

1. What is the value of x after the following statement is executed?

`x = 5 / 2`

2. What is the value of x after the following statement is executed?

`x = 5 // 2`

3. What is the value of x after the following statement is executed?

`x = 2 * 0.5`

4. What is the value of x after the following statement is executed?

`x = 12 % 10`

5. What is the value of x after the following statement is executed?

`x = 10 * ( 1 // 10 )`

6. What is the value of variable x after the following statement is executed?

`x = 5 * 2+3`

7. What is the value of variable x after the following statement is executed?

`x = 2+3 * 5`

Suppose the following definitions are in effect for remaining questions on this page.

`r = "2"`  
`s = "peace"`

8. What is the value of x after the following statement is executed?

`x = r + r`

9. What is the value of x after the following statement is executed?

`x = 3 * r`

10. What is the value of x after the following statement is executed?

`x = s[ 1 ]`

11. What is the value of x after the following statement is executed?

`x = s.replace( "e", "-" )`

12. What is the value of x after the following statement is executed?

`x = s.find( "e" )`

13. What is the value of s after the following statement is executed?

`s.upper()`

14. True or False: The following statement assigns x the value 1112.

```
1112 = x
```

15. True or False: 2b4u is a valid variable name.

16. True or False: dayTime is a valid variable name that does not follow the snake case style rule.

17. True or False: The following assignment correctly swaps the values of x and y.

```
y, x = x, y
```

18. True or False: The following code segment always assigns x the same value.

```
import random
random.seed( 1112 )
x = random.randrange( 1, 100 )
```

19. True or False: A Python program is executed one statement at a time from top to bottom.

20. True or False: The following code segment assigns x the integer list [2, 4, 6].

```
s = "2 4 6"
t = s.split()
x = int( s )
```

**Part II. Programming (10 points each)**

21. Implement program *pb.py*. The program prompts and gets an integer time in jiffies. The program then computes and prints the equivalent number of seconds, where in Computer Science a *jiffy* is one-hundredth of a second. There should be no other output. In performing its computation, your program should make use of the constant

$$\text{JIFFIES\_PER\_SECOND} = 100$$

Sample runs could be.

```
Enter time (jiffies): 1
0.01
```

```
Enter time (jiffies): 1112
11.12
```

22. Implement program *stray.py*. The program prompts and gets a text string. The program then computes and prints the list of characters from that string. There should be no other output.

Sample runs could be.

```
Enter text: apple
['a', 'p', 'p', 'l', 'e']
```

```
Enter text: banana
['b', 'a', 'n', 'a', 'n', 'a']
```

23. Implement program *foreign.py*. The program prompts and gets an integer,  $n$ . The program then computes and prints the total of  $n$  values.

$$\frac{0}{n} + \frac{1}{n} + \frac{2}{n} + \dots + \frac{i}{n} + \dots + \frac{n-1}{n}$$

In the above formula, the division operator is decimal division. Some program runs could be

```
Enter integer: 10
4.5
```

```
Enter integer: 49
24.0
```

There should be no other output than the total.

24. Implement program *double\_vision.py*. The program prompts and gets a text string. The program then computes and prints a list where each word in the input, occurs twice in the outputted list. There should be no other output.

Sample program runs could be.

```
Enter text: wah hoo wah
['wah', 'wah', 'hoo', 'hoo', 'wah', 'wah']
```

```
Enter text: do the right thing
['do', 'do', 'the', 'the', 'right', 'right', 'thing', 'thing']
```

25. Implement program *triangle.py*. The program prompts its user for a string. The program determines the length  $n$  of that string. The program then prints  $n$  output lines. The first output line is the first character of the input string, the second output line is the first two characters of the input string, the third output line is the first three characters of the input string, and so on. There should be no other output.

Sample program runs could be.

```
Enter text: abc
a
ab
abc
```

```
Enter text: quest
q
qu
que
ques
quest
```

26. Implement module *ability.py*. The program analyzes a web file from the web folder

<http://www.cs.virginia.edu/~cs1112/words/>

To do so, the program first prompts and gets the name of a web file from that folder. It then prompts and gets the word of interest for that web file. The program next gets the contents of the web file, and computes and prints the number of times that word appears in the file. It then computes and prints the number of words in the web file. Lastly, it computes and prints the word's percentage of the total number of words in the web file. There should be no other output.

Sample runs could be.

```
Enter filename: haiku.txt
Enter word: the
2
15
0.13333333333333333
```

```
Enter filename: dream.txt
Enter word: have
20
1651
0.012113870381586917
```