

NAME:

CIS 210 Winter 2020 Midterm Exam

Instructions: You are allowed to use one 3×5 notecard with handwritten notes. No other outside sources are allowed. This exam consists of 3 parts: multiple choice (use scantron), short answer (in space provided) and coding (provide hand-written code in space provided).

When turning in your exam: raise your hand and someone will come by to check your seat assignment AND your ID.

MULTIPLE CHOICE

1. Given the following Python code:

```
def char2fnd(s, c):  
    '''(str, str) -> None  
    Prints all the indices in a string s that hold char c  
>>> char2fnd('hello, everyone', 'e')  
1  
7  
9  
14  
'''  
    idx = 0  
    str_remain = s  
  
    for ch in s:  
        if ch == c:  
            idx2 = str_remain.find(ch)  
            str_remain = str_remain[idx2:]  
            print(idx2 + idx)  
            idx = idx + idx2 + 1  
  
    return None
```

1-1: if is a Python

a) primitive element b) identifier c) namespace d) keyword e) library module

2-1: ch is a Python

a) primitive element b) identifier c) namespace d) keyword e) library module

3-1: def is a Python

a) primitive element b) identifier c) namespace d) keyword e) library module

4-1: idx is a Python

a) primitive element b) identifier c) namespace d) keyword e) library module

5-1: 1 is a Python

a) primitive element b) identifier c) namespace d) keyword e) library module

6-1: Which of the following tests would determine an error in the code?

a) char2fnd('eee', 'e') b) char2fnd('every', 'e')

c) char2fnd('', 'e') d) char2fnd('e f g', 'e')

e) All of the above

7. What is the output when the following code is executed?

```
def paramTest(aStr, aType):
    if(isinstance(aStr, aType) != False):
        print(aStr, 'is a str')
    return None

paramTest('101', str)
```

- a) 101 is a str b) '101 is a str' c) Nothing is printed
- d) TypeError: incompatible types
- e) SyntaxError: invalid syntax

8. What is the output of the following *untested and potentially bug-y* code?
(hint: read/manually trace carefully)

```
def add_digits2a(n):
    '''(int) --> int

    >>> add_digits2a(789)
    24
    >>> add_digits2a(101)
    2
    >>> add_digits2a(000)
    0
    '''
    digit_sum = 0
    ctr = 1
    while ctr in range(2):
        digit = n % 10
        n = n // 10
        digit_sum += digit
        ctr += 1

    return digit_sum

print(add_digits2a(156))
```

- a) 6 b. 11 c) 12 d) syntax error: e) none of the above

9. What is the output of the following code?

```
def thrice(b):  
    y = addThree(b, b, b)  
    return y  
  
def addThree(a, b, c):  
    a = 23  
    return a + b + c  
  
b = 3  
thriceOutput = thrice(b)  
print(thriceOutput)
```

- a) 29 b) 9 c) Parameter error: mismatched parameters
d) None e) none of the above

10. Fill in the missing parts of code:

```
def sscount1 (needle, haystack):  
    ''' ?? -> ?? 10-10  
  
    Given a "needle" string to search for in a "haystack"  
    string, return the count of the number occurrences  
    of the needle in the haystack. Overlapping substrings  
    are included. Using string startswith method  
    simplifies code a bit.  
  
    >>> sscount1('ses', 'assesses')  
    2  
    >>> sscount1('an', 'trans-Panamanian banana')  
    6  
    >>> sscount1('needle', 'haystack')  
    0  
    >>> sscount1('!!!', '!!!!!!')  
    3  
    >>> sscount1('o', 'pneumonoultramicroscopicsilicovolcanoconiosis')  
    9  
    '''  
    ctr = 0  
    for i in ??- 11-10:  
        if haystack[i:].startswith(needle):  
            ctr += 1  
    return ??-12-10
```

10-10.

- | | |
|----------------------|------------------------|
| a) (str, str) -> int | b) (str, str) -> float |
| c) (str, str) -> str | d) none of the above |

11-10.

- | | | |
|------------------|-----------------------|-------------------------|
| a) haystack | b) range(len(needle)) | c) range(len(haystack)) |
| d) len(haystack) | e) none of the above | |

12-10.

- | | | | |
|-------------|-----------|--------|----------------------|
| a) haystack | b) needle | c) ctr | d) none of the above |
|-------------|-----------|--------|----------------------|

13. What is the output of the following code?

```
index = 0
while index < 2:
    thing += thing
    index += 1
print(thing)
```

- a) 1 b) 1111 c) 4
d) Type error: unsupported operand type(s) e) none of the above

14. What is the output of the following code?

```
bool0 = True
bool1 = False
print(bool0 and bool1)
print(bool0 or bool1)
```

- a) True b. True
 True False

c) False d) bool1bool2
 True bool0

e) none of the above

15. Given the following Python code:

```
def taxable(inc, exempt, STD_E, STD_D):
    '''(number, int, number, number)

    Adjust gross income (inc) to taxable income
    by applying standard deduction and exemptions.
    CALLED BY: est_tax
    >>> taxable(20000, 1, 4150, 6500)
    9350
    '''
    #print(income)
    #print(salary)
    taxable_income = inc - STD_D
    exempt_adjust = STD_E * exempt
    taxable_income = taxable_income - exempt_adjust
    return(taxable_income)

def est_tax(income, exemptions):
    '''(number, int) -> None

    Generates an estimate for federal income tax.
    CALLS: taxable
    >>> est_tax(20000, 1)
    2000.0
    '''
    STD_EXEMPT = 4000
    STD_DEDUCT = 6000
    TAX_RATE = .20
    taxable_income = taxable(income,exemptions,STD_EXEMPT,STD_DEDUCT)
    estimated_tax = taxable_income * TAX_RATE

    #print('Estimated tax is:', estimated_tax)
    return None

1 def main(salary, exemptions):
2     '''driver for estimated tax functions'''
3     result = est_tax(salary, exemptions)
4     print(result)
5     print(salary)
6     print(taxable_income)
7     return None

salary = 50000
exemptions = 10
main(salary, exemptions)
```

- 15-15. After line 4 in main is executed, what will be printed?
a) 2000.0 b) 50000 c) None d) NameError e) none of the above
- 16-15. After line 5 in main is executed, what will be printed?
a) 2000.0 b) 50000 c) None d) NameError e) none of the above
- 17-15. After line 6 in main is executed, what will be printed?
a) 2000.0 b) 50000 c) None d) NameError e) none of the above
- 18-15. If the `#print (income)` line of code in taxable were uncommented, what would be printed?
a) 2000.0 b) 50000 c) None d) NameError e) none of the above
- 19-15. If the `#print (salary)` line of code in taxable were uncommented, what would be printed?
a) 2000.0 b) 50000 c) None d) NameError e) none of the above

SHORT ANSWER

20. What is the output of this Python program?

```
num1 = 5
if num1 >= 91:
    num2 = 3
else:
    if num1 < 6:
        num2 = 4
    else:
        num2 = 2
x = num2 * num1 + 1
print(x, x%7)
```

ANSWER HERE:

21. What is the output of this Python program?

```
s = 'hello'
s2 = ''
for ch in s:
    s2 = ch + s2
print(s2[1:])
```

ANSWER HERE:

22. What is the output of this Python program?

```
def q22(s):
    '''(str) -> int
    Returns the length of the longest single-character
    string in s.
    >>> q22('abcccddef')
    3
    >>> q22('')
    0
    '''

    if len(s) != 0:
        prev_char = s[0]
        dup_ct = 1
        high_ct = 1
    else:
        high_ct = 0
        dup_ct = 0

    for i in range(1, len(s)):
        if s[i] == prev_char:
            dup_ct += 1
        else:
            prev_char = s[i]

        if dup_ct > high_ct:
            high_ct = dup_ct
            dup_ct = 1

    if dup_ct > high_ct:
        high_ct = dup_ct

    return high_ct

>>> mystr = 'aa'
>>> q22(mystr)
```

ANSWER HERE:

CODING

23. Write Python code to perform the following:

In economics, the percentage rate of inflation for a period of time is calculated based on the final value F of a commodity and the initial value I of the commodity, using the formula $((F - I)/I) \times 100$. In the space below, write a Python function `inflation_rate(initial, final)` to compute and return the inflation rate given the initial and final values of a commodity. Your code should be written using CIS 210 style guidelines:

- include a docstring (type contract and examples of use)
- use whitespace between operators and operands
- use descriptive variable names
- add appropriate comments