

Name: _____ *Keep* Honor Pledge: _____

1. (6) Check the appropriate choice noting which of the following are valid identifiers and which are not.

Valid Invalid

This_is_a_good_identifier	<u>✓</u>	<u>_____</u>
This&That	<u>_____</u>	<u>✓</u>
7UP	<u>_____</u>	<u>✓</u>
R2D2	<u>✓</u>	<u>_____</u>
_StrangeOne	<u>✓</u>	<u>_____</u>
Good4You!	<u>_____</u>	<u>✓</u>

2. (4) What are the values represented by the following arithmetic expressions. Make sure that the value is in the proper form to indicate its type (int or float).

a) $3 + 6 * 4 - 7 \% 3$ <u>26</u>	b) $(3 + 4) / 2 * 5$ <u>15</u>
c) $\text{math.sqrt}(9) * 4 + 3 / 4.0$ <u>12.75</u>	d) $(14 \% 5) / 5.0$ <u>0.8</u>

3. (4) Rewrite the following mathematical expressions into an equivalent Python expression. You may assume that the math library has been imported. Do NOT attempt to transform the expression to an equivalent expression. Rewrite them exactly as they have been presented.

a) $x^3 - 5ab$

*X ** 3 - (5 * a * b)*

b) $4\pi + 6 \div 4$ (make sure you get a floating point result)

*4 * math.pi + 6 / 4.0*

4. (6) Show the list of numbers that would be generated by each of the following range expressions.

a) range(3,10)	<u>[3, 4, 5, 6, 7, 8, 9]</u>
b) range(4, 14, 3)	<u>[4, 7, 10, 13]</u>
c) range(15, 5, -2)	<u>[15, 13, 11, 9, 7]</u>

5. (9) Show the output that would be generated by each of the following program segments.

a) `x = 3`
`y = 12`
`for j in range(0, y, x):`
 `print j,`
 `print j * j`
`print 'done'`

0 0
3 9
6 36
9 81
done

b) `for i in [1, 3, 5, 7]:`
 `print i, ': ', i ** 3`

1: 1
3: 27
5: 125
7: 343

6. (6) What are the values of each of the following assuming that `s = "Hello, I have come."` Assume `string` has been imported. Each line should use the original value of `s`.

`s.capitalize()` Hello, i have come.

`string.capitalize(s)` Hello, I Have Come.

`s.replace('I', 'you')` Hello, you have come.

`s.count('e')` 3

`s.find(',')` 5

`s.lower()` hello, i have come.

7. (4) Show the output that would be generated by each of the following program fragments.

a) `msg = ""`
`for s in string.split("secret", "e"):`
 `msg = msg + s`
`print msg`

secret

b) `for w in string.split('Now is the winter of our discontent...'):`
 `print w`

Now
is
the
winter
of
our
discontent ...

8. (5) What is output from the following print statements?

a) ".3f %0.3f" % (6.4, 6.45896)

6.400 6.459

b) "%7.5f %7.5f" % (2.3, 2.3488)

2.30000 2.34880

c) "There are %d %s %d %s" % (3, 'spams', 4, 'you')

There are 3 spams 4 you

9. (12) What is output from the following loop structures? Use the lines at the right for your answer. There may be extra lines, but the number of lines is sufficient for a correct answer.

a) x = 1

while x <= 7:

 y = 2

 print x + y,

 while y < 15:

 y = (y + x) * 2

 print y,

 y += 2

 print

 x += 2

3 6 18

5 10 30

7 14

9 18

b) for k in range(2,6):

 m = k * 2

 while m < 12:

 print "k = ", k, " m = ", m

 m += 4

k = 2 m = 4

k = 2 m = 8

k = 3 m = 6

k = 3 m = 10

k = 4 m = 8

k = 5 m = 10

10. (4) What are the values of the following boolean expressions assuming x = 6, y = 9, q = True and p = False? Circle the appropriate answer.

a) ☒ T F (x > y-4) and q and not p

b) T ☒ F not(p or q) or x > y

c) T ☒ F x > y + 4 and q

d) ☒ T F x < 10 and y < 10 and q or p

11. (5) What is printed by the following code fragment?

```
myList = []  
for i in range (0,6,2):  
    for k in range(3):  
        myList.append(i+k)  
print i  
print k  
print myList
```

4
2
[0, 1, 2, 2, 3, 4, 4, 5, 6]

12. (6) Write a function that **returns** the area of a triangle given the length of its three sides as parameters (a, b and c). Use the following formulas for your computation. Make sure the value returned is a float. Assume **math** has been properly imported.

$$s = \frac{a + b + c}{2}$$

$$\text{area} = \sqrt{s(s - a)(s - b)(s - c)}$$

```
def area (base, h1, h2):  
    s = (base + h1 + h2) / 2.0  
    a = math.sqrt(s * (s - base) * (s - h1) *  
                  (s - h2))  
    return a
```

13. (8) Assume:

```
strList = ['goodbye', 'cruel', 'world']  
numList = [17, 8, 14]
```

What is output by each of the print statements below?

```
numList[1] = 11  
print strList + numList  
strList.append(numList)  
print strList  
numList.sort()  
print numList  
numList.extend([65, 43, 22])  
print numList  
  
print numList.pop()  
  
print numList  
numList.insert(3, 12)  
print numList  
numList.reverse()  
print numList
```

['goodbye', 'cruel', 'world', 17, 11, 14]
['goodbye', 'cruel', 'world', [17, 11, 14]]
[11, 14, 17]
[11, 14, 17, 65, 43, 22]
22
[11, 14, 17, 12, 65, 43]
[43, 65, 12, 17, 14, 11]

14. (3) What is the value of **List1** after the following code fragment has been executed?

```
L = ['Always', 'look', 'on', 'the', 'bright', 'side', 'of', 'life']  
List1 = [[i.upper(), i.lower()] for i in L if len(i) == 4]
```

[['LOOK', 'look'], ['SIDE', 'side'], ['LIFE', 'life']]

15. (2) What does the following code print?

```
def myFunc(bList):  
    bList[0] = 100  
    aList = [1,2,3]
```

```
aList = [5,6,7]  
myFunc(aList)  
print aList
```

[100, 6, 7]

16. (5) Assuming the following dictionary:

```
passwd = {'guido': 'awesome', 'turing': 'genius', 'bill': 'monopoly'}
```

Follow the following code segment. Indicate what is printed on the lines next to each print statement. Make sure the format of your answer is correct.

```
passwd['newuser'] = 'ImANewbie'
```

```
print passwd.keys() ['guido', 'turing', 'bill', 'newuser']
```

```
print passwd.values() ['awesome', 'genius', 'monopoly', 'ImANewbie']
```

```
print passwd.items() [('guido', 'awesome'), ('turing', 'genius'),  
('bill', 'monopoly'), ('newuser', 'ImANewbie')]
```

```
print passwd.has_key('bill') True
```

```
print 'fred' in passwd False
```

17.(5) What is the list created from the following list comprehensions?

a) [(x,y) for x in range(2,9,3) for y in range(1,9,3) if x > y]

[(2,1), (5,1), (5,4), (8,1), (8,4), (8,7)]

b) [ch for ch in 'When can we go home'.lower() if ch not in ['a', 'e', 'i', 'o', 'u', ' ']] (Do NOT include quotes in answer.)

[w, h, n, c, n, w, g, h, m]

18. (7) Assume the following nonsensical code. Based on the inputs below, what will be the output.

```
x,y,z = raw_input('Enter three integer values: ')
```

```
if x == y:
    if x == z:
        print 'Cheesecake'
    elif x > z:
        print 'Hot dogs'
    else:
        print 'Shrimp of any kind'
elif x > y:
    if x == z:
        print 'Cardigans'
    elif x < z:
        print 'Sweatpants'
    else:
        print 'High heels'
else:
    if x == z:
        print 'Hawaii'
    elif x > z:
        print 'Aruba'
    else:
        print 'Europe'
```

a) 5, 5, 5

Cheesecake

b) 5, 6, 7

Europe

c) 7, 6, 5

High heels

d) 7, 6, 8

Sweatpants

e) 7, 7, 9

Shrimp of any kind

f) 5, 4, 5

Cardigans

g) 5, 7, 5

Hawaii

19. (5) Assume the following class structure. What is output from the following piece of code?

```
class newClass(object):
    def __init__(self, val, mult = 3):
        self.value = val
        self.mult = mult

    def __str__(self):
        return "The value %d times the multiplier %d is %d" % (self.value, self.mult, self.value * self.mult)

    def getVal(self):
        return self.value

    def getMult(self):
        return self.mult

    def setMult(self, mult):
        self.mult = multiplier
```

```
inst1 = newClass(6,5)
inst2 = newClass(4)
```

```
print inst1 The value 6 times the multiplier 5 is 30
```

```
print inst2 The value 4 times the multiplier 3 is 12
```

```
print inst2.getVal() * inst1.getMult() 20
```

```
print inst1.getVal() * inst2.getMult() 18
```

```
inst2.setMult(4)
```

```
print inst2 The value 4 times the multiplier 4 is 16
```

20.(3) Given dictionary D, rewrite this code using exceptions.

```
if x in D:
    D[x] += 1
else:
    D[x] = 1
```

try:
 D[x] += 1
except KeyError:
 D[x] = 1

21. (10) Assume the following program. What is output when it is executed? Put your results in the lines below. Make sure your output is in the proper order (as it would output in the shell).

```
def first (a, b):
    c = a * b + 4
    b = c - a
    print "First: ", a, b, c
    return b
```

```
def second (a, b):
    c = b + (a * b)
    a = b + a - c
    print "Second: ", a, b, c
    return a
```

```
a, b, c = 5, 4, 3
print "Main1: ", a, b, c
b = first (c, a)
c = second(b, c)
print "Main2: ", a, b, c
```

$$\begin{array}{r} a \\ 5 \\ \hline b \\ 4 \\ \hline c \\ 3 \\ \hline \end{array}$$

$$\begin{array}{r} a \\ 3 \\ \hline b \\ 16 \\ \hline c \\ 19 \\ \hline \end{array}$$

$$\begin{array}{r} a \\ 16 \\ \hline b \\ 3 \\ \hline c \\ 51 \\ \hline \end{array}$$

$$\begin{array}{r} 16 \\ 3 \\ \hline 48 \\ \hline \end{array}$$

$$\begin{array}{r} 51 \\ 19 \\ \hline 32 \\ \hline \end{array}$$

main1: 5 4 3

First: 3 16 19

Second: -32 3 51

main2: 5 16 -32

22. (5) If mySet = set('xyz') and yourSet = set('uvwxyz'), what are the values of: (use correct set notation).

myset.union(yourset) set(['u', 'v', 'w', 'x', 'y', 'z'])

myset.intersection(yourset) set(['x', 'y', 'z'])

myset.difference(yourset) set([])

yourset.difference(myset) set(['u', 'v', 'w'])

myset.symmetric_difference(yourset) set(['u', 'v', 'w'])

23. (4) Convert 1732 (base 10) to binary and hexadecimal.

Binary: 11011000100

Hexadecimal: 6C4

24. (4) Convert **1011011011** (base 2) to hexadecimal and decimal.

Hexadecimal: 5B7

Decimal: 1463

25. (4) Using an **8-bit byte**, show the two's complement representation for **98**. Then show how **-98** is represented in two's complement. (**Again be sure to use 8 bits.**)

98 in two's complement: 01100010

-98 in two's complement: 10011110

26. (4) Using **8-bit bytes**, perform the following add instructions assuming the number are in two's complement notation. Note if the answer has a problem with overflow.

$$\begin{array}{r} 11011001 \\ +11110101 \\ \hline 11001110 \end{array}$$

$$\begin{array}{r} 01100101 \\ +01011110 \\ \hline 11000011 \end{array}$$

Overflow (Y / N)

Overflow (Y / N)

27. (8) Convert the following number to binary notation and then to floating point notation using the floating point representation discussed in class with **1 bit for the sign bit, 3 bits for the exponent in excess-4 notation and 4 bits for the fraction.**

-2 3/4

binary: 10.11

floating-point: 11101011

5/64

binary: .000101

floating-point: 00011010

28. (2) Assuming the same 8-bit represent for floating-point numbers, convert **01011110** from floating point to decimal. Leave your final answer as a decimal fraction.

Decimal: 1 3/4