

- Be sure to enter your full name, student ID and your answers to questions on your answer sheet, being distributed to you separately from this exam booklet.
- Do not turn this page until instructed to.
- There are 30 multiple choice questions worth 1 point each.
- Each multiple choice question has only **one** correct answer.
- You must not communicate with other students during this test.
- No books, notes, or electronic devices are permitted. In other words, you are not allowed to use a dictionary on your mobile phone or other electronic devices. However, if you don't understand the meaning of a particular English word in this exam, please raise your hand and the instructor will explain the meaning of the English word to you.

Fill in your information:

Full Name: _____

Student ID: _____

1. (1 point) Consider the following program:

```
a = 1
def f( a ):
    a = 2 * a
    return a
    a = 3
x = f( a ) + a
```

What is the **value** of **x** after this program is executed?

- A. 1
- B. 2
- C. ★ 3
- D. 4
- E. None of the other answers are correct.

2. (1 point) Evaluate the following expression:

`(True or False) or not (False and True)`

What value is produced?

- A. False
- B. ★ True

3. (1 point) Consider the following incomplete program.

```
a = "ROBIN"
b = "HOOD " # note the space
d = { }
for x,y in zip( a,b ):
    ???
s = ""
for c in a:
    s += d[ c ]
print( s )
```

What should replace the three question marks to cause this program to print out 'HOOD '?

- A. ★ `d[x] = y`
- B. `d[y] = x`
- C. `d[a] = b`
- D. `d[b] = a`
- E. `d[a] = x`

4. (1 point) Consider the following program.

```
s="".join(["2","0","1"])
x=0
for a,b in enumerate(s):
    x+=a+int(b)
```

After it is run, what is the final **value** of **x**?

- A. 5
- B. 9
- C. 3
- D. 201
- E. ★ None of the other answers are correct.

5. (1 point) Consider the following program.

```
x='@'
x+='@' # x+='@'
#x+='@'
'''
'''
x+='''@'''
'''
x+='@'
'''
```

After it is run, what is the final **value** of **x**?

- A. '@'
- B. '@@'
- C. ★ '@@@'
- D. '@@@@'
- E. '@@@@@'

6. (1 point) Consider the following program.

```
d = { }  
for a,b in enumerate( '0123456' ):  
    d[ b ] = a  
x = d[ '2' ] + d[ '1' ]
```

After it is run, what is the final **value** of **x**?

- A. 1
- B. 4
- C. 2
- D. 5
- E. ★ 3

7. (1 point) Evaluate the following expression:

```
len(",3,4,5,6,7,".split(','))
```

- A. 5
- B. 6
- C. ★ 7
- D. 25
- E. '34567'

8. (1 point) Consider the following program.

```
a,b = "KING","JOHN"  
def f( a,b ):  
    return b,a  
x = ','.join( f( a,b ) )
```

After it is run, what is the final **value** of **x**?

- A. "KING", "JOHN"
- B. "JOHN", "KING"
- C. "KING,JOHN"
- D. ★ "JOHN,KING"
- E. None of the other answers are correct

9. (1 point) Consider the following program.

```
a,b = "MAID","MARIAN"
def f( a,b ):
    return "%sx%d" % (a , len(b))
    b,a = a,b
x = f( a,b )
```

After it is run, what is the final **value** of **x**?

- A. "MAID",6
- B. "MARIAN",4
- C. "MAIDxMARIAN"
- D. ★ "MAIDx6"
- E. None of the other answers are correct

10. (1 point) Consider the following program.

```
import numpy as np
x=np.array([ [1,2] ]*2)+1
```

After it is run, what is the final **value** of **x**?

A. ★ $\begin{bmatrix} 2 & 3 \\ 2 & 3 \end{bmatrix}$

B. $\begin{bmatrix} 3 & 5 \end{bmatrix}$

C. $\begin{bmatrix} 3 \\ 5 \end{bmatrix}$

D. $\begin{bmatrix} 2 & 3 & 2 & 3 \end{bmatrix}$

- E. None of the other answers are correct

11. (1 point) Consider the following 2-dimensional numpy array:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

Assuming it is stored in a variable named `a`, how can we index and retrieve the value 6?

- A. ★ `a[1][2]`
- B. `a[2][1]`
- C. `a[2][3]`
- D. `a[3][2]`

12. (1 point) Consider the following 2-dimensional array:

$$\begin{bmatrix} 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

Which of the following expressions will generate this array?

- A. ★ `np.array([[6,5,4], [3,2,1]])`
- B. `np.array([[6,3], [5,2], [4,1]])`
- C. `np.array([[6,5], [4,3], [2,1]])`
- D. `np.array([[6,5,4,3,2,1]])`

13. (1 point) Consider the following program.

```
d = { "T":2,"U":1,"C":4,"K":1 }
for c in "FRIAR":
    print( d[ c ] + 3 )
```

What kind of exception will this program throw?

- A. ★ `KeyError: 'F'`
- B. `TypeError: list indices must be integers, not str`
- C. `SyntaxError: invalid syntax`
- D. `TypeError: unsupported operand type(s) for +: 'str' and 'int'`

14. (1 point) Consider the following program.

```
import numpy as np
x = np.zeros( ( 3,3 ) )
for i in range( 3 ):
    for j in range( 3 ):
        x[ i ][ j ] = i*i + j
```

After it is run, what is the final **value** of **x**?

A. $\begin{bmatrix} 0 & 1 & 4 \\ 1 & 2 & 5 \\ 2 & 3 & 6 \end{bmatrix}$

B. ★ $\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

C. $\begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 3 & 6 \end{bmatrix}$

D. $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$

E. None of the other answers are correct

15. (1 point) Consider the following program.

```
import numpy as np
x = np.zeros( ( 3,3 ) )
for i in range( 3 ):
    x[ i,i ] = 1
    for j in range( 3 ):
        if i < j:
            continue
        x[ i,j ] = 2
```

After it is run, what is the final **value** of **x**?

A. ★ $\begin{bmatrix} 2 & 0 & 0 \\ 2 & 2 & 0 \\ 2 & 2 & 2 \end{bmatrix}$

B. $\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}$

C. $\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$

D. $\begin{bmatrix} 1 & 2 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$

E. $\begin{bmatrix} 2 & 2 & 2 \\ 0 & 2 & 2 \\ 0 & 0 & 2 \end{bmatrix}$

16. (1 point) Consider the following program.

```
a = [ "1",2,"3",0 ]
x = 0
for e in a:
    try:
        x += e
    except:
        x += 1
```

After it is run, what is the final **value** of **x**?

A. 1

B. 2

C. 3

D. ★ 4

E. 5

17. (1 point) Consider the following program.

```
x = "1 2 3".split()
x = ','.join( x )
try:
    print( x.append( 4 ) )
except:
    print( type( x ) )
```

After it is run, what is printed by this program?

- A. `TypeError`
- B. `[1,2,3,4]`
- C. `<class 'list'>` (output from calling the `type` function on a list)
- D. ★ `<class 'str'>` (output from calling the `type` function on a string)

18. (1 point) Consider the following incomplete program:

```
x = "TUCK"
y = "CKTU"
z = ""
???
print( z )
```

Replacing the three question marks with which of the following will result in an output of "TUCK"?

- A. ★

```
for a in x:
    for b in y:
        if a == b:
            z += b
```
- B.

```
for a in y:
    for b in x:
        if a == b:
            z += b
```
- C.

```
for a in x:
    for b in y:
        if a != b:
            z += b
```
- D.

```
for a in y:
    for b in x:
        if a != b:
            z += b
```

19. (1 point) For this problem, your job is to put the lines of code below in the proper order to create a function that accomplishes a task. We will completely ignore indentation.

```
1 except:
2 def is_close( a,b,rtol=1e-3 ):
3 try:
4 return None
5 def is_close( a,b,rtol )
6 rtol = 1e-3
7 return ( abs(a-b)/abs(b) <= rtol )
8 return ( (a-b)/b <= rtol )
```

The function you should write is called `is_close`, and it should accept a two numbers, `a` and `b`. An optional third argument is the relative tolerance `rtol` with default value `1e-3`. `is_close` returns `True` or `False` depending on whether the numbers are closer than `rtol`:

$$\frac{|a-b|}{|b|} \leq \text{rtol} \rightarrow \text{True} \qquad \frac{|a-b|}{|b|} > \text{rtol} \rightarrow \text{False}$$

The code should return `None` if the calculation fails (for instance, if the parameters `a` or `b` are non-numeric).

What is the proper selection and ordering of the given lines of code?

- A. ★ 2, 3, 7, 1, 4
- B. 2, 3, 8, 1, 4
- C. 2, 7
- D. 5, 6, 3, 8, 1, 4
- E. 5, 6, 3, 7, 1, 4

20. (1 point) Consider the following program.

```
def gisbourne( guy ):  
    guy.append( "arrow" )  
    guy.reverse()  
    guy = guy.append( "arrow" )  
    return guy  
  
sheriff = "nottingham forest".split(" ")  
gisbourne( sheriff )
```

After it is run, what is the final **value** of **sheriff**?

- A. ★ ['arrow', 'forest', 'nottingham', 'arrow']
- B. ['arrow', 'arrow', 'forest', 'nottingham']
- C. None
- D. ['forest', 'nottingham']
- E. ['nottingham', 'forest']

21. (1 point) Consider the following Python program.

```
d = { 0:0,1:0,2:0 }  
for i in range( 0,5 ):  
    d[ i%3 ] += i  
x = d[ 1 ]
```

After it is run, what is the final **value** of **x**?

- A. 6
- B. ★ 5
- C. 4
- D. 3
- E. 2

22. (1 point) Consider the following program.

```
def richard( s ):
    return s * 3
    return s
```

```
s = richard( "LIONHEART" )
```

After it is run, what is the final **value** of **s**?

- A. ★ "LIONHEARTLIONHEARTLIONHEART"
- B. "LIONHEART"
- C. "LLLLIIIOOONNNHHHEEEAAARRRTTT"
- D. "LIONHEART3"
- E. None

23. (1 point) Consider the following program.

```
x = [ ]
for j in range( 0,5 ):
    if ( j%4 ) == 0:
        x.append( "-" )
    if ( j%5 ) == 0:
        x.append( "*" )
```

After it is run, what is the final **value** of **x**?

- A. ["*", "-", "-"]
- B. ['*', '-', '*']
- C. ['*', '-', '-']
- D. ★ ['-', '*', '-']

24. (1 point) Consider the following program:

```
???  
sin(math.pi)+cos(math.pi)
```

What should replace the three question marks to produce a program that runs without throwing an exception? Note: `sin`, `cos`, and `pi` are all part of the `math` module.

A. ★

```
from math import sin,cos  
import math
```

B.

```
from math import *  
import sin,cos
```

C.

```
import pi,sin,cos  
import math
```

D.

```
import math as pi  
import math as sin  
import math as cos
```

25. (1 point) Consider the following program.

```
import numpy as np
x = np.zeros( ( 4,3 ) )
for i in range(1,4 ):
    for j in range( 3 ):
        x[ i ][ j ] = x[ i-1 ][ j ] + 1
```

After it is run, what is the final **value** of **x**?

A. $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 0 & 0 & 0 \end{bmatrix}$

B. $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

C. $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

D. ★ $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$

E. $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{bmatrix}$

26. (1 point) Which of the following expressions is susceptible of numerical error as discussed in class? Assume that any variables used are defined.

A. `round(a+b) == 4`

B. ★ `a + b == 0.3`

C. `abs(a + b) / c <= 1e-3`

D. `np.isclose(a,b)`

27. (1 point) Which of the following Python programs completes without raising an error? Assume the following dictionary is available:

```
mydict = { 'a':1, 'b':2, 'c':3 }
```

- A.

```
for key in mydict:
    mydict[ key ] = key.upper()
mydict[ 'D' ] = 4
```
- B.

```
for value in mydict.values(): # values is a valid method
    mydict[ value ] = value.upper()
mydict[ 'D' ] = 4
```
- C.

```
for key in mydict:
    mydict[ key ] = mydict[ key ].upper()
mydict.append( 'D':4 )
```
- D. ★

```
for key in mydict:
    mydict[ key ] = key.upper()
mydict[ 'D' ] = 4
```

28. (1 point) Consider the following incomplete Python program:

```
def fib( n ):
    if n <= 1:
        return 1
    else:
        ???
```

The function `fib` should return the n th number of the Fibonacci sequence (counting from zero), in which each number is equal to the sum of the preceding two; *i.e.*,

1, 1, 2, 3, 5, 8, 13, 21, 34, ...

What should replace the `???` block to complete the program correctly?

- A. ★

```
return fib( n-1 ) + fib( n-2 )
```
- B.

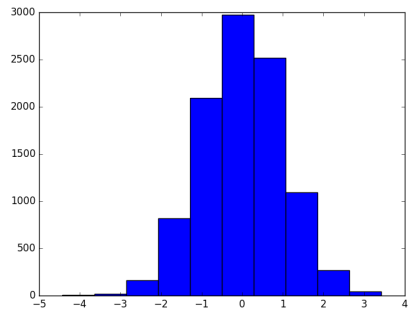
```
return fib( n ) + fib( n-1 )
```
- C.

```
return fib( n-1, n-2 )
```
- D.

```
return (n - 1) + (n - 2)
```
- E.

```
return fib[ n-1 ] + fib[ n-2 ]
```

29. (1 point) Consider the following plot, produced from 10,000 random numbers selected from an as-yet-undetermined distribution.



Which of the following programs could produce this plot? Assume that NumPy and Matplotlib have been imported as customary in class. Also assume that all programs work as written.

- A.

```
x = np.random.uniform( size=(10000,1) )  
plt.plot( x )  
plt.show()
```
- B. ★

```
x = np.random.randn( 10000,1 )  
plt.hist( x )  
plt.show()
```
- C.

```
x = np.random.randint( 0,10, size=(10000,1) )  
plt.hist( x )  
plt.show()
```
- D.

```
x = np.random.uniform( size=(10000,1) )  
plt.hist( x )  
plt.show()
```


30. (1 point) Which of the following Python programs best simulates the roll of one twenty-sided die in the variable `x`? (*I.e.*, any number from 1–20 inclusive is equally likely to result from the die roll or program code.)

A. ★

```
x = np.random.choice( np.arange( 1,21 ) )
```

B. `x = np.random.shuffle(np.arange(1,21))`

C. `x = np.random.randn(np.arange(1,21))`

D. `x = np.random.uniform(np.arange(1,21))`