Name:____

Section:______________________ **Date:**__________________

INSTRUCTIONS:

(1) DO NOT OPEN YOUR EXAM BOOKLET UNTIL YOU HAVE BEEN TOLD TO BEGIN.

(2) This exam booklet contains 30 questions, each of which will be weighted equally at 5 points each. The total points for the exam is 150 points.

(3) You may use one 8.5" x 11" note sheet during the exam. No other reference materials or electronic devices may be used during the examination, i.e. no calculators, cell phones, mp3 players, etc. Paper dictionaries are allowed.

(4) Questions will not be interpreted during the examination.

(5) You should choose the *single best alternative* for each question, even if you believe that a question is ambiguous or contains a typographical error. If a question has more than one best answer, credit will be given for any of the correct answers. Provide only one answer.

(6) Please fill in the requested information at the top of this exam booklet.

(7) Use a #2 pencil to encode answers on the OMR form.

(8) Please encode the following on the OMR form:
– Last name and first initial
– MSU PID
– Exam form (2 A)

(9) Please sign the OMR form.

(10) Only answers recorded on your OMR form will be counted for credit. Completely erase any responses on the OMR form that you wish to delete.

(11) You must turn in this exam booklet, the OMR form, your note sheet, and your scrap paper when you have completed the exam. Be sure your name is on any of these items that you would like to have returned.

(12) The exam ends at 2:00 pm. When leaving, please be courteous to those still taking the exam.

1	2	3	4	5	6	7	8	9	10
D	A	C	C	A	D	A	D	A	B

11	12	13	14	15	16	17	18	19	20
C	A	E	C	D	C	A	B	E	A

21	22	23	24	25	26	27	28	29	30
C	B	B	A	C	A	D	*	B	D

Error in Q13: extra comma before the assignment was not intended.
Either (d) or (e) given full credit (but correct answer is (e)).

Error in Q19: print instruction removed from the program, but the output it produced was mistakenly left in the answer. Either (b) or (e) given full credit (but correct answer is (e)).

Q24: partial credit (2 pts) for marking (b); for (b) to be correct, the function would need a return instruction, e.g. return par

Q28: this question was thrown out because of poor wording. (c) was poorly worded: in fact, assignment does not return a value --- that is subtlely different from returning None. So none of the alternatives is technically correct. Any answer was accepted for this question. (Including no answer.)

```
a = [1] * 5
print( a )     # Line 1

a = list( range( 2 ) )
print( a )     # Line 2

a *= 3
print( a )     # Line 3
```

Figure 1

(1) In Fig. 1, what is displayed by the line labeled `Line 1`?

(a) `[5, 5, 5, 5, 5]` (b) `5` (c) `[5]`
(d) `[1, 1, 1, 1, 1]` (e) None of (a)–(e)

(2) In Fig. 1, what is displayed by the line labeled `Line 2`?

(a) `[0, 1]` (b) `[1, 2]` (c) `[0, 1, 0, 1, 0, 1]`
(d) `[1, 2, 1, 2, 1, 2]` (e) None of (a)–(e)

(3) In Fig. 1, what is displayed by the line labeled `Line 3`?

(a) `[1, 2]` (b) `[1, 2, 1, 2, 1, 2]` (c) `[0, 1, 0, 1, 0, 1]`

(d) `[0, 3]` (e) None of (a)–(e)

```
c = []
for n in range( 3 ):
    c.append( 10 * n - 2 )
print( c )
```

Figure 2

(4) In Fig. 2, what is displayed?

(a) `24` (b) `[8, 18, 28]` (c) `[-2, 8, 18]`
(d) `[0.01, 0.1, 1]` (e) None of (a)–(e)

```
b = ['let', 'bygones', 'be']
print( 'b' in b )                  # Line 1
print( ' '.join(b) )               # Line 2
print( b )                         # Line 3
print( b[1:-1] )                   # Line 4
print( b[1][-1] )                  # Line 5
b.sort()
print( b )                         # Line 6
```

FIGURE 3

(5) In Fig. 3, what is displayed by the line labeled `Line 1`?

(a) `False` (b) `'b' in ['let', 'bygones', 'be']`
(c) `True` (d) None of (a)–(c)

(6) In Fig. 3, what is displayed by the line labeled `Line 2`?

(a) `['let', 'bygones', 'be']` (b) `['be', 'bygones', 'let']`
(c) `letbygonesbe` (d) `let bygones be`
(e) None of (a)–(d)

(7) In Fig. 3, what is displayed by the line labeled `Line 3`?

(a) `['let', 'bygones', 'be']` (b) `let bygones be`
(c) `letbygonesbe` (d) `['be', 'bygones', 'let']`
(e) None of (a)–(d)

(8) In Fig. 3, what is displayed by the line labeled `Line 4`?

(a) `['be', 'bygones']` (b) `['be', 'bygones', 'let']`
(c) `['bygones', 'be']` (d) `['bygones']`
(e) None of (a)–(d)

(9) In Fig. 3, what is displayed by the line labeled `Line 5`?

(a) `s` (b) `['s']`
(c) `t` (d) `['bygones']`
(e) None of (a)–(d)

(10) In Fig. 3, what is displayed by the line labeled `Line 6`?

(a) `['let', 'bygones', 'be']` (b) `['be', 'bygones', 'let']`
(c) `letbygonesbe` (d) `let bygones be`
(e) None of (a)–(d)

```
def foo( x = 4, y = -1 ):
    x = x - y
    y = 2 * x
    return x + y

x, y = 1, 2
z = foo( x, y )
print( x, y, z )     # Line 1

x, y = 3, 2
z = foo( y )
print( x, y, z )     # Line 2

x, y, = 2, 5
z = foo( y = x )
print( x, y, z )     # Line 3
```

Figure 4

(11) In Fig. 4, what is displayed by the line labeled `Line 1`?

(a) `5 10 15` (b) `-1 -2 -3` (c) `1 2 -3`
(d) `1 2 15` (e) None of (a)–(e)

(12) In Fig. 4, what is displayed by the line labeled `Line 2`?

(a) `3 2 9` (b) `3 2 -3` (c) `3 2 6`
(d) `-1 -2 -3` (e) None of (a)–(e)

(13) In Fig. 4, what is displayed by the line labeled `Line 3`?

(a) `2 5 9` (b) `2 4 9` (c) `3 6 6`
(d) `2 5 6` (e) None of (a)–(e)

(14) In Fig. 4, which name is not in scope in the body (suite) of function `foo`?

(a) `x` (b) `y` (c) `z`
(d) `foo` (e) None of (a)–(e)

```
s0 = set()
for i in [2, 3, 9, 5]:
    s0 = s0 | { i%3 }
print(s0)                              # Line 1

s1 = set( range( 1, 6, 2 ) )
s2 = set( [ 3*n - 1 for n in range(3) ] )
print(s1 & s2)                         # Line 2
print(s2 - s1)                         # Line 3
print ( (s1 & s2) < (s1 | s2) )        # Line 4
```

FIGURE 5

(15) In Fig. 5, what is displayed by the line labeled `Line 1`?

(a) `{0, 1, 3}`
(b) `{2, 3, 9, 5}`
(c) `{}`
(d) `{0, 2}`
(e) None of (a)–(d)

(16) In Fig. 5, what is displayed by the line labeled `Line 2`?

(a) `{}`
(b) `{-1, 2, 3, 5}`
(c) `{5}`
(d) `{-1, 2, 3}`
(e) None of (a)–(d)

(17) In Fig. 5, what is displayed by the line labeled `Line 3`?

(a) `{2, -1}`
(b) `{3, 1}`
(c) `{}`
(d) `{3, -1, 2}`
(e) None of (a)–(d)

(18) In Fig. 5, what is displayed by the line labeled `Line 4`?

(a) `False`
(b) `True`
(c) None of (a)–(b)

```
def get_value( prompt ):
    value_str = input( prompt )
    try:
        value = float( value_str )
    except ValueError:
        print ( "** Invalid input **" )
        value = 0
    return value

def main():
    try:
        val1 = get_value( "Enter the 1st value:  " )
        val2 = get_value( "Enter the 2nd value:  " )
        print( round(val1/val2, 1) )
    except ZeroDivisionError:
        print( "** Invalid **" )

    print( "Done" )

main()
```

FIGURE 6

(19) In Fig. 6, what is displayed if the user enters 2 at the first prompt and 3 at the second?

(a)
```
Enter the 1st value:  2
Enter the 2nd value:  3
2.0 / 3.0:  0.7
```

(b)
```
Enter the 1st value:  2
Enter the 2nd value:  3
2.0 / 3.0:  0.7
Done
```

(c)
```
Enter the 1st value:  2
Enter the 2nd value:  3
2 / 3:  0.7
```

(d)
```
Enter the 1st value:  2
Enter the 2nd value:  3
2 / 3:  0.6
Done
```

(e) None of (a)–(d)

(20) In Fig. 6, what is displayed if the user enters `’2’` at the first prompt and `’3’` at the second (with the quotes)?

(a)
```
Enter the 1st value:  ’2’
** Invalid input **
Enter the 2nd value:  ’3’
** Invalid input **
** Invalid **
Done
```

(b)
```
Enter the 1st value:  ’2’
Enter the 2nd value:  ’3’
2.0 / 3.0:  0.7
Done
```

(c)
```
Enter the 1st value:  ’2’
** Invalid input **
Enter the 2nd value:  ’3’
** Invalid input **
Done
```

(d)
```
Enter the 1st value:  ’2’
** Invalid input **
Enter the 2nd value:  ’3’
** Invalid input **
Invalid
```

(e) None of (a)–(d)

(21) In Fig. 6, what is displayed if the user enters `2.3` at the first prompt and `0` at the second?

(a)
```
Enter the 1st value:  2.3
** Invalid input **
Enter the 2nd value:  0
** Invalid input **
** Invalid **
Done
```

(b)
```
Enter the 1st value:  2.3
Enter the 2nd value:  0
** Invalid input **
0.0 / 2.3:  0.0
Done
```

(c)
```
Enter the 1st value:  2.3
Enter the 2nd value:  0
** Invalid **
Done
```

(d)
```
Enter the 1st value:  2.3
Enter the 2nd value:  0
** Invalid **
```

(e) None of (a)–(d)

(22) In Fig. 6, what is displayed if the user enters `0` at the first prompt and `two` at the second?

(a)
```
Enter the 1st value:  0
** Invalid input **
Enter the 2nd value:  two
** Invalid input **
** Invalid **
Done
```

(b)
```
Enter the 1st value:  0
Enter the 2nd value:  two
** Invalid input **
** Invalid **
Done
```

(c)
```
Enter the 1st value:  0
Enter the 2nd value:  two
0.0 / 2.0:  0.0
Done
```

(d)
```
Enter the 1st value:  0
Enter the 2nd value:  two
0 / 2:  0
Done
```

(e) None of (a)–(d)

```
# remember mutable vs.  immutable!
def fun1( par, num ):
    par[0] = num

a1 = [’a’, ’b’, ’c’]
a2 = fun1( a1, 2 )

print(a1)                                # Line 1
print(a2)                                # Line 2

def fun2 (p1,p2):
    p1 = p1[::-1]
    p2 = max(p2) + min(p2)
    return p1, p2

t1 = (’a’, ’b’)
t2 = (’A’, ’B’)
a,b = fun2(t1,t2)

print(a)                                 # Line 3
print(b)                                 # Line 4
print(t2)                                # Line 5
```

Figure 7

(23) In Fig. 7, what is displayed by the line labeled `Line 1`?

(a) `None` (b) `[2, ’b’, ’c’]` (c) `[’a’, ’b’, ’c’]`
(d) `[’a’]` (e) None of (a)–(d)

(24) In Fig. 7, what is displayed by the line labeled `Line 2`?

(a) `None` (b) `[2, ’b’, ’c’]` (c) `[’a’, ’b’, ’c’]`
(d) `[’a’]` (e) None of (a)–(d)

(25) In Fig. 7, what is displayed by the line labeled `Line 3`?

(a) `(’a’, ’b’)` (b) `[’b’, ’a’]` (c) `(’b’, ’a’)`
(d) `[b]` (e) None of (a)–(d)

(26) In Fig. 7, what is displayed by the line labeled `Line 4`?

(a) `BA` (b) `B` (c) `A`
(d) `[ba]` (e) None of (a)–(d)

(27) In Fig. 7, what is displayed by the line labeled `Line 5`?

(a) `AB` (b) `(’B’, ’A’)` (c) `BA`
(d) `(’A’, ’B’)` (e) None of (a)–(d)

(28) Which of the following is correct?
(a) Python is an interpreted language.
(b) The Python assignment instruction produces a side effect.
(c) The Python assignment instruction returns `None`.
(d) All of (a)–(c) are correct.
(e) None of (a)–(c) are correct.

```
import string

def dissemble(d, w):
    for c in w:
        if c in d:
            if w not in d[c]:
                d[c].append(w)
        else:
            d[c] = [w]

s_str = ’A man.  A plan.’
s_lst = s_str.split()
w_lst = [w.strip(string.punctuation).lower() for w in s_lst]

data = {}
for w in w_lst:
    dissemble(data, w)

if ’n’ in data:          # first if
    print( sorted( data[’n’] ) )
else:
    print( "Not there." )

if ’.’  in data:          # second if
    print( sorted( data[’.’]  )  )
else:
    print( "Not there." )
```

Figure 8

(29) In Fig. 8, what is displayed by the first `if` statement (labeled `first if`)?

(a) `[’plan’, ’man’]` (b) `[’man’, ’plan’]`
(c) `[’A’, ’a’, ’man’, ’plan’]` (d) `Not there.`
(e) None of (a)–(d)

(30) In Fig. 8, what is displayed by the second `if` statement (labeled `second if`)?

(a) `[’plan.’, ’man.’]` (b) `[’man.’, ’plan.’]`
(c) `[’A’, ’a’, ’man’, ’plan’]` (d) `Not there.`
(e) None of (a)–(d)

Scratch