

NATIONAL UNIVERSITY OF SINGAPORE
Department of Computer Science, School of Computing
IT5001 – *Software Development Fundamentals*
Academic Year 2025/2026, Semester 1

Mid-Term Assessment
SOLUTIONS MANUAL

8 October 2025

Time allowed: 1 hour

MULTIPLE CHOICE QUESTIONS [75 marks]

1. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print(True and False and True and True)
```

- A. True
- B. False**
- C. None
- D. 0
- E. Running the code results in an error

2. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print((True + 1) * 25 // 4)
```

- A. 12**
- B. 12.5
- C. 6.25
- D. False
- E. Running the code results in an error

3. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print(not ((1 < 3) + 1) - 2)
```

- A. True**
- B. False
- C. -2
- D. -1
- E. Running the code results in an error

4. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print(len("Cult of the Lamb"))
```

- A. 13
- B. 14
- C. 15
- D. 16**
- E. 17

5. Which of the following will produce the following string? **[3 marks]**

```
'Rounding Adj.: -0.01'
```

- A. "Rounding Adj.: {}".format(-0.01)
- B. "Rounding Adj.: " + str(-0.01)
- C. "Rounding Adj.: {}".format(-0.01)
- D. All of the above**
- E. None of the above

6. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
e, w, f = "earth", "wind", "fire"
topic = "file io"
old_macdonald_farm = "-".join([e[0], w[1], f[-1], topic[::-1][:2][::-1]])

print(old_macdonald_farm)
```

- A. e-i-e-io**
- B. e-i-e-i-o
- C. e-i-e-o-i
- D. e-i-e-oi
- E. None of the above

7. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print([18, 28, 38, 48, 58][-1])
```

- A. 18
- B. 28
- C. 38
- D. 48
- E. 58**

8. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
x = list(1, 2, 3, 4)
print(x)
```

- A. [1, 2, 3, 4]
- B. [1]
- C. [] # Empty list
- D. [(1, 2, 3, 4)]

E. Running the code results in an error

9. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
lst1 = ["Mario"]
lst1.append("Luigi")
lst1.extend("Bowser")
lst2 = lst1
lst2.pop()
lst1.pop(0)
print(lst1)
```

- A. ['Mario']
- B. ['Luigi']
- C. ['Luigi', 'B', 'o', 'w', 's', 'e']**
- D. ['Luigi', 'B', 'o', 'w', 's', 'e', 'r']
- E. Running the code results in an error

10. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print(tuple("bravo")[1])
```

- A. ('b', 'r', 'a', 'v', 'o')
- B. 'r'**
- C. ('r',)
- D. Running the code results in an error
- E. None of the above

11. The following tuple was defined in a standalone Python file.

```
my_tuple = ('a', 1, [2], (3,))
```

Which of the following expressions/statements would cause an error? **[3 marks]**

- A. 1 in my tuple
- B. my_tuple[2][0] = 4
- C. len(my_tuple[3])
- D. my_tuple[0] = my_tuple[0] + 'b'**
- E. None of the above

12. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
tup1 = tuple(i for i in range(0, 4, -1))
for i in range(10, 51, 10):
    if i % 4 == 0:
        tup1 = tup1 + (i,)
print(tup1)
```

- A. (20, 40)
- B. (10, 20, 30, 40)
- C. (10, 20, 30, 40, 50)
- D. (0, 1, 2, 3, 20, 40)
- E. (0, 1, 2, 3, 10, 20, 30, 40)

13. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
print(set("heffalump"))
```

- A. ('a', 'e', 'f', 'h', 'l', 'm', 'p', 'u')
- B. {'a', 'e', 'f', 'h', 'l', 'm', 'p', 'u'}
- C. ('a', 'e', 'f', 'f', 'h', 'l', 'm', 'p', 'u')
- D. {'a', 'e', 'f', 'f', 'h', 'l', 'm', 'p', 'u'}
- E. None of the above

14. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
pooh = set("pooh")
piglet = set("piglet")
tigger = set("tigger")

print(((pooh & piglet) - tigger) ^ pooh | pooh)
```

- A. set() # Empty set
- B. {'h', 'o', 'p'}
- C. {'h', 'l', 'o', 'p'}
- D. {'h', 'o', 'o', 'p'}
- E. Running the code results in an error

15. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
print({k: k*2 for k in range(4)})
```

- A. {1:2, 2:4, 3:6}
- B. {1:2, 2:4, 3:6, 4:8}
- C. {0:0, 1:2, 2:4, 3:6, 4:8}
- D. {0:0, 1:2, 2:4, 3:6}
- E. Running the code results in an error

16. Suppose we have a dictionary defined as follows:

```
my_dict = {1:10, 2:15, 3:1, 4:3,
           5:6, 6:4, 10:5, 15:2}
```

Which of the following expressions will produce an output of 10? **[3 marks]**

- A. `my_dict[my_dict[1]] - my_dict.get(4,6)`
- B. `my_dict[my_dict[5]] - my_dict.get(3,7)`
- C. `my_dict[my_dict[15]] - my_dict.get(10,5)`
- D. All of the above
- E. None of the above

17. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
d1 = {i: [_ for _ in range(i+4)] for i in range(3)}
d2 = d1.copy()
d1[0] = len(d1)
d1[1].append(-1)
print(d2)
```

- A. `{0: [0, 1, 2, 3],`
 `1: [0, 1, 2, 3, 4, -1],`
 `2: [0, 1, 2, 3, 4, 5]}`
- B. `{0: [0, 1, 2, 3],`
 `1: [0, 1, 2, 3, 4],`
 `2: [0, 1, 2, 3, 4, 5]}`
- C. `{0: 3,`
 `1: [0, 1, 2, 3, 4, -1],`
 `2: [0, 1, 2, 3, 4, 5]}`
- D. `{0: 3,`
 `1: [0, 1, 2, 3, 4],`
 `2: [0, 1, 2, 3, 4, 5]}`
- E. Running the code results in an error

18. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
mine = False
yours = mine
stars = 0
if mine:
    stars = 5
elif yours:
    stars = 4
if not stars:
    stars = 2
print(stars)
```

- A. 0
- B. 2**
- C. 4
- D. 5
- E. None of the above

19. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
def weekend(posn, dest):
    if posn != dest:
        if dest == 'JB':
            print('take bus')
        elif dest == 'KL':
            print('take train')
        print('go shopping')
    else:
        print('stay home')
```

weekend('SG', 'KL' or 'JB')

- A. stay home
- B. go shopping
- C. take bus
go shopping
- D. take train
go shopping**
- E. take bus
take train
go shopping

20. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
val = 4
for k in range(10):
    val += 5
    if k > 5:
        val += 1
print(val)
```

- A. 60
- B. 59
- C. 58**
- D. 55
- E. 54

21. The following code is in a standalone Python file. How many lines are printed when the code is run? **[3 marks]**

```
def combo(moves, target):
    for i in range(len(moves)):
        (move1, pt1) = moves[i]
        gg = False
        j = i+1

        while (not gg) and j < len(moves):
            (move2, pt2) = moves[j]
            pts = pt1 + pt2
            print(f'{move1}-{move2} combo: {pts} pts')
            if pts >= target:
                gg = True
            j += 1

    if gg:
        print('Good game!')

moves = [('run',100), ('jump',250), ('kick',300), ('shoot',350)]
combo(moves, 500)
```

- A. 5
- B. 6
- C. 7**
- D. 8
- E. 9

22. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
print((lambda a,b,c: [a+b, b+c])('q','b','9'))
```

- A. 'qbb9'
- B. 'qb, b9'
- C. ['q', 'b', '9']
- D. ['qb', 'b9']**
- E. Running the code results in an error

23. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
def alpha(f,c):  
    return lambda x,y: f(x,y) + c  
def beta(n,e):  
    return n ** e  
print(alpha(beta,13)(5,3))
```

- A. 28
- B. 125
- C. 138**
- D. 256
- E. Running the code results in an error

24. The following code is in a standalone Python file. What is the output of the code when it is run? **[3 marks]**

```
def boo():  
    global d  
    d = 'halloween'  
  
def foo(d):  
    d = 'thanksgiving'  
    boo()  
    goo(d)  
  
def goo(d):  
    print(d)  
    d = 'christmas'  
  
d = 'holiday'  
foo(d)  
print(d)
```

A. halloween
christmas

B. halloween
holiday

**C. thanksgiving
halloween**

D. thanksgiving
holiday

E. Running the code results in an error

25. The following code is in a standalone Python file. What is the output of the code when it is run? [3 marks]

```
d = {'I': '1',
     'Z': '2',
     'E': '3',
     'H': '4',
     'S': '5',
     'G': '6',
     'T': '7',
     'B': '8',
     'Q': '9',
     'O': '0'}

def f(s):
    if s == '':
        return ''
    if s[0] in d:
        return d[s[0]] + f(s[1:])
    return s[0] + f(s[1:])

def g(s):
    out = ''
    for c in s:
        if c in '1234567890':
            out = out + str(9-int(c))
        else:
            out = out + c
    return out

print(g(f('HELLO THERE')))
```

- A. 6R652 9LL96
- B. 56LL9 256R6**
- C. 43LL0 743R3
- D. 3R347 0LL34
- E. Running the code results in an error

1. B	2. A	3. A	4. D	5. D
6. A	7. E	8. E	9. C	10. B
11. D	12. A	13. B	14. B	15. D
16. C	17. A	18. B	19. D	20. C
21. C	22. D	23. C	24. C	25. B

FILL-IN-THE-BLANKS QUESTIONS [25 marks]

26. The following function will take in a string and output a string with every character doubled up.

Sample output:

```
>>> print(double_up('abc'))
aabbcc
>>> print(double_up('IT5001'))
IITT55000011
```

Complete the function by filling in the blanks. [5 marks]


```
def double_up(s):
    output = ''
    for c in ___BLANK_1___:
        output += ___BLANK_2___
    return output
```

ANSWER

```
def double_up(s):
    output = ''
    for c in s:          # BLANK 1
        output += c*2    # BLANK 2
    return output
```

27. In many countries, beef is graded into different quality levels. One of the predominant grading systems is by the Beef Marble Score (BMS) in Japan. Here is how the wagyu beef (in Japanese: 和牛) is graded in Japan.

GRADE	BMS
5	8-12
4	5-7
3	3-4
2	2
1	1



Complete the following function which outputs the grade of the beef from the input BMS (as integer). You can assume that BMS is always an integer such that $0 < \text{BMS} < 13$. **[5 marks]**

```
def beef_grade(BMS):
    if ___BLANK_1___:
        return ___BLANK_2___
    elif ___BLANK_3___:
        return ___BLANK_4___
    elif ___BLANK_5___:
        return ___BLANK_6___
    elif ___BLANK_7___:
        return ___BLANK_8___
    else:
        return ___BLANK_9___
```

Sample output:

```
>>> print(beef_grade(12))
5
>>> print(beef_grade(2))
2
```

ANSWER

```
def beef_grade(BMS):
    if BMS >= 8:           # BLANK 1
        return 5          # BLANK 2
    elif BMS >= 5:         # BLANK 3
        return 4          # BLANK 4
    elif BMS >= 3:         # BLANK 5
        return 3          # BLANK 6
    elif BMS >= 2:         # BLANK 7
        return 2          # BLANK 8
    else:
        return 1          # BLANK 9
```

28. Secret agents always use encryption to pass messages. For example, the following message does not make much sense:

```
"ill the nest to our madterm"
```

But if you exchange 'i' and 'a', and change 'n' to 'b', it says:

```
"all the best to our midterm"
```

The given function deciphers a message with two inputs (an encrypted message string *S* and a decryption key *d* in a dictionary form), and outputs the deciphered message. The function will return a new string from *S*, where every character from *S* that are keys of the dictionary is replaced with the corresponding value in the dictionary.

Sample output:

```
>>> keys = {'i': 'a', 'n': 'b', 'a': 'i'}
>>> message = 'ill the nest to our madterm'
>>> print(decipher(message,keys))
all the best to our midterm
```

Complete the following function to achieve the above functionality. **[5 marks]**

```
def decipher(s,d):
    output = ''
    for ___BLANK_1___:
        if ___BLANK_2___:
            output += ___BLANK_3___
        else:
            output += ___BLANK_4___
    return output
```

ANSWER

```
def decipher(s,d):
    output = ''
    for c in s:
        if c in d:
            output += d[c]
        else:
            output += c
    return output
```

29. Given an array of integers and a number n, return True if the array contains two consecutive n in it.

Sample output:

```
>>> print(has_nn([1,2,3,1,2,3,3,2,1],1))
False
>>> print(has_nn([1,2,3,1,2,3,3,2,1],3))
True
```

Complete the function by filling in the blanks. **[5 marks]**

```
def has_nn(nums,n):
    for i in __BLANK_1__:
        if __BLANK_2__:
            return True
    return False
```

ANSWER

```
def has_nn(nums,n):
    for i in range(len(nums)-1):           # BLANK 1
        if nums[i] == n and nums[i+1] == n: # BLANK 2
            return True
    return False
```

30. There is a secret way to pass a secret message just by looking at the first letter of every word in a sentence. For example, if the sentence is

"Cute animals that sleep",

the answer is "Cats".

Complete a one-liner function `acrostics()` which carries out this process. By "one-liner", it means that your function body should be contained in just one line (and without any semicolons). Also, your function **CANNOT** be recursive.

Fill up the blank to complete the function. **[5 marks]**

```
def acrostics(secret):  
    return ___BLANK_1___
```

Sample output:

```
>>> print(acrostics('Cute animals that sleep'))
```

Cats

```
>>> print(acrostics('Since everyone can read encoding text in normal sentences  
is doubtfully effective'))
```

Secretinside

ANSWER

```
def acrostics(secret):  
    return ''.join(map(lambda x:x[0], secret.split(' '))) # BLANK 1
```

-- End of Solutions Manual --