NATIONAL UNIVERSITY OF SINGAPORE

Department of Computer Science, School of Computing

IT5001—*Software Development Fundamentals*

Academic Year 2022/2023, Semester 1

Mid-Term Assessment

QUESTION BOOKLET

24 September 2022 **Time allowed:** 1 hour

INSTRUCTIONS TO CANDIDATES (please read carefully):

1. This is a **closed-book assessment**. You are allowed **ONE (1)** A4-sized reference sheet, double-sided, printed or written, and **ONE (1)** blank A4 paper for scratch.
2. You may use a non-programmable, NUS-approved calculator.
3. Use of any other electronic devices, including smart watches, is **NOT** allowed.
4. The assessment consists of three (3) documents—the 'QUESTION BOOKLET' (this document), the 'ANSWER BOOKLET', and an 'OCR Answer Sheet'. Do **NOT** open these documents until you are told to do so.
5. The 'QUESTION BOOKLET' (this document) comprises **THIRTY (30) questions** and **TEN (10) pages** including this cover page.
6. The 'ANSWER BOOKLET' comprises **FOUR (4) pages** including the cover page.
7. **Do NOT** write your name on any document you submit.
8. Write your Student Number (starting with `A`) on the 'ANSWER BOOKLET'.
9. Write and shade your Student Number (starting with `A`) on the 'OCR Answer Sheet' **using a 2B pencil**.
10. For Multiple-Choice Questions (MCQs) Q1 to Q25 (inclusive), shade your answers on the 'OCR Answer Sheet' **using a 2B pencil**.
11. For the remaining questions, write your answers in the space provided in the 'ANSWER BOOKLET'; no extra sheets will be accepted as answers. You may write in blue or black with a pencil or pen.
12. You are required to only submit the 'ANSWER BOOKLET' and the 'OCR Answer Sheet' at the end of the assessment. You may use the 'QUESTION BOOKLET' (this document) as scratch paper.
13. The total attainable score for this assessment is **100 marks**. You must complete all questions to score full marks. This assessment counts towards **20%** of your final grade.
14. You **cannot** communicate with anyone other than the invigilators throughout the exam.
15. **You must attempt the assessment on your own**. The University takes a zero-tolerance approach towards plagiarism and cheating.

Expression Evaluation [24 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each question, one or more Python expressions are entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the final expression entered and shade the correct option in the 'OCR Answer Sheet'.

Question 1) [2 marks]

```
1(2 + 3) % 4
```

Options:
A. `0`
B. `1`
C. `5`
D. `None`
E. Evaluating this expression yields an error

Question 2) [2 marks]

```
int(13 / 4)
```

Options:
A. `3`
B. `4`
C. `5`
D. `None`
E. Evaluating this expression yields an error

Question 3) [2 marks]

```
'IT' * (5 * 0 ** 0 * 1)
```

Options:
A. `'ITITITITIT'`
B. `''`
C. `'IT'`
D. `None`
E. Evaluating this expression yields an error

Question 4) [2 marks]

```
(1 != 2) * 3
```

Options:
A. `[False, False, False]`
B. `0`
C. `3`
D. `None`
E. Evaluating this expression yields an error

Question 5) [2 marks]

```
[1, 2] + (3, 4)
```

Options:
A. `[1, 2, 3, 4]`
B. `[1, 2, (3, 4)]`
C. `(1, 2, 3, 4)`
D. `None`
E. Evaluating this expression yields an error

Question 6) [2 marks]

```
sum([1, 2, 3, 4])
```

Options:
A. `10`
B. `15`
C. `''`
D. `None`
E. Evaluating this expression yields an error

(The **Expression Evaluation** section continues in the next page...)

Question 7) [2 marks]

```
x = [5, 0, 0, 1]
x += 'IT'
x
```

Options:

A. `['5', '0', '0', '1', 'IT']`

B. `[5, 0, 0, 1, 'IT']`

C. `[5, 0, 0, 1, 'I', 'T']`

D. `None`

E. Evaluating at least one of these expressions yields an error

Question 8) [2 marks]

```
[1, 2, 3][4:5] and 'IT5001!'
```

Options:

A. `'IT5001!'`

B. `True`

C. `[]`

D. `False`

E. Evaluating this expression yields an error

Question 9) [2 marks]

```
[[1, 2], [3, 4]][1][0]
```

Options:

A. `1`

B. `2`

C. `3`

D. `None`

E. Evaluating this expression yields an error

Question 10) [2 marks]

```
list(filter(bool, [0, 1, 2]))
```

Options:

A. `[1, 2]`

B. `[0, 1, 2]`

C. `[False, True, True]`

D. `None`

E. Evaluating this expression yields an error

Question 11) [2 marks]

```
dict('24 Sept 2022')
```

Options:

A. `{'2': '2', ' ': 'S', 'e': 'p', 't': ' '}`

B. `{'2': '4'}`

C. `{'2': '4 Sept 2022'}`

D. `None`

E. Evaluating this expression yields an error

Question 12) [2 marks]

```
sum(map(lambda z: z(1), map(lambda x: lambda y: x + y, range(5))))
```

Options:

A. `5`

B. `10`

C. `15`

D. `None`

E. Evaluating this expression yields an error

(The next section begins in the next page...)

True or False Questions [16 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each of these questions you are given either a statement or a Python expression. For each of these, determine if the statement is true or false, or if the expression evaluates to `True` or `False`, and shade the correct option in the 'OCR Answer Sheet'.

Question 13) [2 marks]. The expression `1 + 2 > 3 - 4` evaluates to `True`.
Options:
A. True
B. False

Question 14) [2 marks]. The expression `'abcdE' > 'abcDe'` evaluates to `True`.
Options:
A. True
B. False

Question 15) [2 marks]. The list `[i + 1 for i in range(0, 10, 2)]` has 6 elements.
Options:
A. True
B. False

Question 16) [2 marks]. You cannot add elements via `append` to a tuple like `([1, 2], [3, 4])` because tuples are immutable objects.
Options:
A. True
B. False

Question 17) [2 marks]. Lists cannot be added to dictionaries as values.
Options:
A. True
B. False

Question 18) [2 marks]. For some expression `map(f, ls)`, `ls` cannot be a two-dimensional list because `f` must be a function that receives only one parameter.
Options:
A. True
B. False

(The **True or False Questions** section continues in the next page...)

Question 19) [2 marks]. We say 'x contains itself' to mean that `x in x` gives `True`.
As we have seen, a list `ls` can contain itself, such as when we do `ls.append(ls)`. However, a set/tuple can never contain itself.
Options:
A. True
B. False

Question 20) [2 marks]. If we do not define functions using `def` and we only define functions using lambda expressions, we cannot write recursive functions.
Options:
A. True
B. False

Program Tracing [20 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each of the following questions in this section, you are given a complete Python program stored in a `.py` file. Determine the output (if any) of the program upon execution, and shade the correct option in the 'OCR Answer Sheet'.

Question 21) [4 marks]

```
count = 0
for char in 'IT5001!':
  if char in 'IT':
    count += 1
print(count)
```

Options:
A. 0
B. 1
C. 2
D. No output will be produced
E. An error message will be produced

(The **Program Tracing** section continues in the next page...)

Question 22) [4 marks]

```
def f(s, c):
  if not s:
    return 0
  rec = f(s[1:], c)
  if s[0] == c:
    return 1 + rec
  return rec
print(f('IT5001 is the best!', 'i'))
```

Options:
A. 0
B. 1
C. 2
D. No output will be produced
E. An error message will be produced

Question 23) [4 marks]

```
def f(d, name):
  for k, v in d.items():
    if name == v:
      return k
d = {123: 'Alice', 456: 'Bob', 789: 'Charlie'}
print(f(d, 'Bob'))
```

Options:
A. Bob
B. 456
C. None
D. No output will be produced
E. An error message will be produced

(The **Program Tracing** section continues in the next page...)

Question 24) [4 marks]

```
def c(f, g):
  return lambda x: f(g(x))
def f(x):
  return x + 1
def g(x):
  return x * 2
print(c(f, g)(3))
```

Options:
A. 3
B. 7
C. 8
D. No output will be produced
E. An error message will be produced

Question 25) [4 marks]

```
def f(g, x):
  if isinstance(x, int):
    return [g(x)]
  result = []
  for i in x:
    result.extend(f(g, i))
  return result
print(f(lambda x: x * x, [[[1, [2]], [3], 4], 5, [6]]))
```

Options:
A. `[[[1, [2]], [3], 4], 5, [6]]`
B. `[[[1, [4]], [9], 16], 25, [36]]`
C. `[1, 4, 9, 16, 25, 36]`
D. No output will be produced
E. An error message will be produced

(The next section begins in the next page...)

Program Comprehension [28 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each of the following questions in this section, you are given a complete Python program stored in a `.py` file. Answer the questions posed to you and write your answers in the 'ANSWER BOOKLET'. You can only obtain full marks for a question if you answer accurately, concisely, and write legibly.

Question 26) [7 marks]. Observe the following program fragment.

```
def f26(a, b, c):
  if a > b:
    if a > c:
      return a
    return c
  if b > c:
    return b
  return c
```

Assuming the arguments to `f26` are all integers, **describe function `f26`; or in other words, what does `f26` do?**

Question 27) [7 marks]. Observe the following program fragment.

```
def f27(n):
    x = n
    while True:
        if x ** 2 <= n:
            return x
        x -= 1
```

Example uses of `f27` follow:

```
>>> f27(100)
10
>>> f27(99)
9
```

Assuming the argument to `f27` is a nonnegative integer, **describe function `f27`; or in other words, what does `f27` do?**

(The **Program Comprehension** section continues in the next page...)

Question 28) [7 marks]. Observe the following program fragment.

```
def f28(s):
    return ''.join(map(lambda x: x[0].upper(), s.split()))
```

Assuming the argument to `f28` is always a string, **describe function `f28`; or in other words, what does `f28` do?**

Question 29) [7 marks]. Observe the following program fragment.

```
def f29(ls):
    d = {}
    res = set()
    m = -1
    for i in ls:
        d[i] = d.get(i, 0) + 1
    for k, v in d.items():
      if v > m:
        res = set()
        m = v
      if v == m:
        res.add(k)
    return res
```

Example uses of `f29` follow:

```
>>> f29([1, 'T', 5, 0, 0, 1])
{0, 1}
>>> f29('The quick brown fox jumps over the lazy dog')
{' '}
```

Assuming the argument to `f29` is always a nonempty sequence of elements, and the elements themselves can be added to dictionaries as keys, and added to sets. **Describe function `f29`; or in other words, what does `f29` do?**

(the next section begins in the next page...)

Programming [12 marks]

In this section, you are given an incomplete Python program stored in a `.py` file. Answer the questions posed to you and write your answers in the 'ANSWER BOOKLET', by replacing each blank with a syntactically correct Python expression/statement. You can only obtain full marks for this question if you answer accurately, concisely, and write legibly.

Question 30) [20 marks]. A table is represented as a two-dimensional list whose elements are the rows of the table, where the first row is its header.

The `get_col_index` function receives a table and some column name, and determines the column index of that column:

```
>>> table = [['Name',  'Score'],
             ['Alice', 100],
             ['Bob',   90]]
>>> get_col_index(table, 'Name')
0
>>> get_col_index(table, 'Score')
1
```

The `query` function receives a table, a key column name, and a key; it returns the first row whose value at the key column is equal to the key:

```
>>> table = [['Name',  'Score'],
             ['Alice', 100],
             ['Bob',   90]]
>>> query(table, 'Name', 'Bob')
['Bob', 90]
>>> query(table, 'Score', 100)
['Alice', 100]
```

An incomplete implementation of `query` is given below. Replace each blank with a valid Python expression/statement and write your answers in the 'ANSWER BOOKLET'. Assume that the table is always nonempty, and that the key column name is always valid.

```
def get_col_index(table, col_name):
  return table[0].index(col_name)

def query(table, key_col_name, key):
  key_col_index = <BLANK_1>
  for row in <BLANK_2>:
    if <BLANK_3>:
      return row[:]
```

– End of Assessment –

NATIONAL UNIVERSITY OF SINGAPORE

Department of Computer Science, School of Computing

IT5001—*Software Development Fundamentals*

Academic Year 2022/2023, Semester 1

Mid-Term Assessment

SOLUTIONS MANUAL

24 September 2022 **Time allowed:** 1 hour

Expression Evaluation [24 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each question, one or more Python expressions are entered into a fresh Python shell with no prior import statements. Determine the result from evaluating the final expression entered and shade the correct option in the 'OCR Answer Sheet'.

Question 1) [2 marks]
`1(2 + 3) % 4`
Options:
A. `0`
B. `1`
C. `5`
D. `None`
(E). Evaluating this expression yields an error

Question 2) [2 marks]
`int(13 / 4)`
Options:
(A). `3`
B. `4`
C. `5`
D. `None`
E. Evaluating this expression yields an error

Question 3) [2 marks]
`'IT' * (5 * 0 ** 0 * 1)`
Options:
(A). `'ITITITITIT'`
B. `''`
C. `'IT'`
D. `None`
E. Evaluating this expression yields an error

Question 4) [2 marks]
`(1 != 2) * 3`
Options:
A. `[False, False, False]`
B. `0`
(C). `3`
D. `None`
E. Evaluating this expression yields an error

Question 5) [2 marks]
`[1, 2] + (3, 4)`
Options:
A. `[1, 2, 3, 4]`
B. `[1, 2, (3, 4)]`
C. `(1, 2, 3, 4)`
D. `None`
(E). Evaluating this expression yields an error

Question 6) [2 marks]
`sum([1, 2, 3, 4])`
Options:
(A). `10`
B. `15`
C. `''`
D. `None`
E. Evaluating this expression yields an error

(The **Expression Evaluation** section continues in the next page...)

Question 7) [2 marks]

```
x = [5, 0, 0, 1]
x += 'IT'
x
```

Options:

A. `['5', '0', '0', '1', 'IT']`

B. `[5, 0, 0, 1, 'IT']`

(C). `[5, 0, 0, 1, 'I', 'T']`

D. `None`

E. Evaluating at least one of these expressions yields an error

Question 8) [2 marks]

```
[1, 2, 3][4:5] and 'IT5001!'
```

Options:

A. `'IT5001!'`

B. `True`

(C). `[]`

D. `False`

E. Evaluating this expression yields an error

Question 9) [2 marks]

```
[[1, 2], [3, 4]][1][0]
```

Options:

A. `1`

B. `2`

(C). `3`

D. `None`

E. Evaluating this expression yields an error

Question 10) [2 marks]

```
list(filter(bool, [0, 1, 2]))
```

Options:

(A). `[1, 2]`

B. `[0, 1, 2]`

C. `[False, True, True]`

D. `None`

E. Evaluating this expression yields an error

Question 11) [2 marks]

```
dict('24 Sept 2022')
```

Options:

A. `{'2': '2', ' ': 'S', 'e': 'p', 't': ' '}`

B. `{'2': '4'}`

C. `{'2': '4 Sept 2022'}`

D. `None`

(E). Evaluating this expression yields an error

Question 12) [2 marks]

```
sum(map(lambda z: z(1), map(lambda x: lambda y: x + y, range(5))))
```

Options:

A. `5`

B. `10`

(C). `15`

D. `None`

E. Evaluating this expression yields an error

(The next section begins in the next page...)

True or False Questions [16 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each of these questions you are given either a statement or a Python expression. For each of these, determine if the statement is true or false, or if the expression evaluates to `True` or `False`, and shade the correct option in the 'OCR Answer Sheet'.

Question 13) [2 marks]. The expression `1 + 2 > 3 - 4` evaluates to `True`.
Options:
(A). True
B. False

Question 14) [2 marks]. The expression `'abcdE' > 'abcDe'` evaluates to `True`.
Options:
(A). True
B. False

Question 15) [2 marks]. The list `[i + 1 for i in range(0, 10, 2)]` has 6 elements.
Options:
A. True
(B). False (it has five elements)

Question 16) [2 marks]. You cannot add elements via `append` to a tuple like `([1, 2], [3, 4])` because tuples are immutable objects.
Options:
(A). True (tuples are immutable)
B. False

Question 17) [2 marks]. Lists cannot be added to dictionaries as values.
Options:
A. True
(B). False (they can)

Question 18) [2 marks]. For some expression `map(f, ls)`, `ls` cannot be a two-dimensional list because `f` must be a function that receives only one parameter.
Options:
A. True
(B). False (no reason why it can't)

(The **True or False Questions** section continues in the next page...)

Question 19) [2 marks]. We say 'x contains itself' to mean that `x in x` gives `True`.
As we have seen, a list `ls` can contain itself, such as when we do `ls.append(ls)`. However, a set/tuple can never contain itself.
Options:
(A). True (unless you manually assign the reference stored in the tuple, but that breaks abstraction barrier (also depends on reference implementation); sets are unhashable therefore cannot be contained in sets)
B. False

Question 20) [2 marks]. If we do not define functions using `def` and we only define functions using lambda expressions, we cannot write recursive functions.
Options:
A. True
(B). False (use Z combinator)

Program Tracing [20 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each of the following questions in this section, you are given a complete Python program stored in a `.py` file. Determine the output (if any) of the program upon execution, and shade the correct option in the 'OCR Answer Sheet'.

Question 21) [4 marks]

```
count = 0
for char in 'IT5001!':
  if char in 'IT':
    count += 1
print(count)
```

Options:
A. `0`
B. `1`
(C). `2`
D. No output will be produced
E. An error message will be produced

(The **Program Tracing** section continues in the next page...)

Question 22) [4 marks]

```
def f(s, c):
  if not s:
    return 0
  rec = f(s[1:], c)
  if s[0] == c:
    return 1 + rec
  return rec
print(f('IT5001 is the best!', 'i'))
```

Options:
A. 0
(B). 1
C. 2
D. No output will be produced
E. An error message will be produced

Question 23) [4 marks]

```
def f(d, name):
  for k, v in d.items():
    if name == v:
      return k
d = {123: 'Alice', 456: 'Bob', 789: 'Charlie'}
print(f(d, 'Bob'))
```

Options:
A. Bob
(B). 456
C. None
D. No output will be produced
E. An error message will be produced

(The **Program Tracing** section continues in the next page...)

Question 24) [4 marks]

```
def c(f, g):
  return lambda x: f(g(x))
def f(x):
  return x + 1
def g(x):
  return x * 2
print(c(f, g)(3))
```

Options:
A. 3
(B). 7
C. 8
D. No output will be produced
E. An error message will be produced

Question 25) [4 marks]

```
def f(g, x):
  if isinstance(x, int):
    return [g(x)]
  result = []
  for i in x:
    result.extend(f(g, i))
  return result
print(f(lambda x: x * x, [[[1, [2]], [3], 4], 5, [6]]))
```

Options:
A. `[[[1, [2]], [3], 4], 5, [6]]`
B. `[[[1, [4]], [9], 16], 25, [36]]`
(C). `[1, 4, 9, 16, 25, 36]`
D. No output will be produced
E. An error message will be produced

(The next section begins in the next page...)

Program Comprehension [28 marks]

There are several questions in this section. Answer each question **independently and separately**.

In each of the following questions in this section, you are given a complete Python program stored in a `.py` file. Answer the questions posed to you and write your answers in the 'ANSWER BOOKLET'. You can only obtain full marks for a question if you answer accurately, concisely, and write legibly.

Question 26) [7 marks]. Observe the following program fragment.

```
def f26(a, b, c):
  if a > b:
    if a > c:
      return a
    return c
  if b > c:
    return b
  return c
```

Assuming the arguments to `f26` are all integers, **describe function `f26`; or in other words, what does `f26` do?**

Question 27) [7 marks]. Observe the following program fragment.

```
def f27(n):
    x = n
    while True:
        if x ** 2 <= n:
            return x
        x -= 1
```

Example uses of `f27` follow:

```
>>> f27(100)
10
>>> f27(99)
9
```

Assuming the argument to `f27` is a nonnegative integer, **describe function `f27`; or in other words, what does `f27` do?**

(The **Program Comprehension** section continues in the next page...)

Question 28) [7 marks]. Observe the following program fragment.

```
def f28(s):
    return ''.join(map(lambda x: x[0].upper(), s.split()))
```

Assuming the argument to `f28` is always a string, **describe function `f28`; or in other words, what does `f28` do?**

Question 29) [7 marks]. Observe the following program fragment.

```
def f29(ls):
    d = {}
    res = set()
    m = -1
    for i in ls:
        d[i] = d.get(i, 0) + 1
    for k, v in d.items():
      if v > m:
        res = set()
        m = v
      if v == m:
        res.add(k)
    return res
```

Example uses of `f29` follow:

```
>>> f29([1, 'T', 5, 0, 0, 1])
{0, 1}
>>> f29('The quick brown fox jumps over the lazy dog')
{' '}
```

Assuming the argument to `f29` is always a nonempty sequence of elements, and the elements themselves can be added to dictionaries as keys, and added to sets. **Describe function `f29`; or in other words, what does `f29` do?**

(the next section begins in the next page...)

Programming [12 marks]

In this section, you are given an incomplete Python program stored in a `.py` file. Answer the questions posed to you and write your answers in the 'ANSWER BOOKLET', by replacing each blank with a syntactically correct Python expression/statement. You can only obtain full marks for this question if you answer accurately, concisely, and write legibly.

Question 30) [20 marks]. A table is represented as a two-dimensional list whose elements are the rows of the table, where the first row is its header.

The `get_col_index` function receives a table and some column name, and determines the column index of that column:

```
>>> table = [['Name',  'Score'],
             ['Alice', 100],
             ['Bob',   90]]
>>> get_col_index(table, 'Name')
0
>>> get_col_index(table, 'Score')
1
```

The `query` function receives a table, a key column name, and a key; it returns the first row whose value at the key column is equal to the key:

```
>>> table = [['Name',  'Score'],
             ['Alice', 100],
             ['Bob',   90]]
>>> query(table, 'Name', 'Bob')
['Bob', 90]
>>> query(table, 'Score', 100)
['Alice', 100]
```

An incomplete implementation of `query` is given below. Replace each blank with a valid Python expression/statement and write your answers in the 'ANSWER BOOKLET'. Assume that the table is always nonempty, and that the key column name is always valid.

```
def get_col_index(table, col_name):
  return table[0].index(col_name)

def query(table, key_col_name, key):
  key_col_index = <BLANK_1>
  for row in <BLANK_2>:
    if <BLANK_3>:
      return row[:]
```

– End of Assessment –

Question 26) [7 marks]. Describe function `f26`; or in other words, what does `f26` do?

"It returns $\max\{a, b, c\}$."

Question 27) [7 marks]. Describe function `f27`; or in other words, what does `f27` do?

"It returns $\lfloor \sqrt{n} \rfloor$."

Question 28) [7 marks]. Describe function `f28`; or in other words, what does `f28` do?

"It returns an acronym for `a`."

Question 29) [7 marks]. Describe function `f29`; or in other words, what does `f29` do?

"It returns the set of all elements who occur the most frequently in `ls`"

Question 30) [12 marks]. Replace each blank with a valid Python expression/statement and write your answers in the table provided below.

Blank	**Your Answer**
<BLANK_1>	`get_col_index(table, key_col_name)`
<BLANK_2>	`table[1:]`
<BLANK_3>	`row[key_col_index] == key`

– End of Solutions Manual –