

THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2017
Campus: City

COMPUTER SCIENCE

TEST

Principles of Computer Science

(Time Allowed: ONE hour)

Note:

- The use of calculators is NOT permitted.
- Compare the term test version number on the Teleform sheet supplied with the version number above. If they do not match, ask the supervisor for a new sheet.
- Enter your name and student ID on the Teleform sheet. Your name should be entered left aligned. If your name is longer than the number of boxes provided, truncate it.
- Answer all **Multiple-choice** questions on the Teleform answer sheet provided. Answer Section **B** in the space provided in this booklet. Attempt all questions.
- Use a dark pencil to mark your answers in the multiple choice answer boxes on the Teleform sheet. Check that the question number on the sheet corresponds to the question number in this question/answer book. If you spoil your sheet, ask the supervisor for a replacement.
- Write your answers in the space provided in the short answer section. Write as clearly as possible. The space provided will generally be sufficient but is not necessarily an indication of the expected length. Extra space is provided at the end of this exam book.

Surname:	
First Name(s):	
Student ID:	
Lab Time:	

MARKERS ONLY

	Question	Mark	Out Of
1 – 25	Multiple Choice		34
26 – 27	Written		16
	TOTAL		50

Question 1

[1 mark] If you would like to add a value of 2 to every element in `list1`, which of the following code should you use?

- (a)

```
for x in list1:  
    list1[x] += 2
```
- (b)

```
for x in list1:  
    x += 2
```
- (c)

```
for i in range(len(list1)):  
    list1[i] += 2
```
- (d)

```
for x in list1:  
    x = x + 2
```
- (e) None of the above

Question 2

[1.5 marks] What output would be generated by the following code fragment?

```
x = [1, 2]  
y = [9, 8]  
for item in x:  
    item += 3  
for index in range(len(y)):  
    y[index] = x[index] + y[index]  
print('Result =', x[0], y[0])
```

- (a) Result = 13 22
- (b) Result = 1 10
- (c) Result = 13 13
- (d) Result = 10 10
- (e) Result = 4 13

Question 3

[1.5 marks] What output would be generated by the following code fragment?

```
x = [[1], [1, 2]]  
x[0].append([3, 4])  
x[1].extend([5, 6])  
print(x)
```

- (a) `[[1, [3, 4]], [1, 2, 5, 6]]`
- (b) `[[1], [1, 2], [3, 4], [5], [6]]`
- (c) `[[1], [1, 2], [3, 4], [5, 6]]`
- (d) `[[3, 4], [1], [5], [6], [1, 2]]`
- (e) None of the above

Question 4

[1 mark] What output would be generated by the following code fragment?

```
values = ['a', 'aba', [1], [1, 3], 'abcd']
my_list = [ x * 2 for x in values if len(x) > 1]
print(my_list)
```

- (a) ['aa', 'abaaba', [1, 1], [1, 3, 1, 3], 'abcdabcd']
- (b) ['abaaba', [1, 3, 1, 3], 'abcdabcd']
- (c) ['abaaba', 'abcdabcd']
- (d) ['abaaba', [2, 6], 'abcdabcd']
- (e) None of the above

Question 5

[1.5 marks] What output would be generated by the following code fragment?

```
phrase = 'happy'
vowels = 'aeiou'
my_list = [phrase.find(c) for c in phrase if c not in vowels]
print('Result =', my_list)
```

- (a) Result = [0, 1, 2, 2, 4]
- (b) Result = [1, 2, 2, 1]
- (c) Result = [1, 1, 1, 1]
- (d) Result = [0, 2, 2, 4]
- (e) Result = [0, 2, 2, 5]

Question 6

[1.5 marks] What output would be generated by the following code fragment?

```
values = ['hello', 'world']
my_list = [values[x][::-1] for x in range(len(values) - 1, -1, -1)]
print(my_list)
```

- (a) ['dlrow', 'olleh']
- (b) ['hello', 'world']
- (c) ['world', 'hello',]
- (d) ['olleh', 'dlrow']
- (e) None of the above

Question 7

[1.5 marks] What output would be generated by the following code fragment?

```
x = [1, 2]
x1 = [1, 2]
x2 = [1, 2]
y = x
x += [3]
y1 = x1
x1 = x1 + [3]
y2 = x2
x2.append(3)
print(x is y, x1 is y1, x2 is y2)
```

- (a) False True False
- (b) True True False
- (c) True False True
- (d) False True True
- (e) None of the above

Question 8

[1.5 marks] What output would be generated by the following code fragment?

```
x = [1, 3]
y = x
y += [5]
x = y[::-1]
print(x == y, x is y, x, y)
```

- (a) False False [3, 1] [1, 3, 5]
- (b) False False [5, 3, 1] [1, 3, 5]
- (c) True True [1, 3, 5] [1, 3, 5]
- (d) True False [1, 3, 5] [1, 3, 5]
- (e) True False [5, 3, 1] [5, 3, 1]

Question 9

[1.5 marks] The class definition of the `Book` class is given in page 6. Consider the following code:

```
json_dict = json.loads(????)
result = Book(json_dict['__title'], json_dict['__pages'] )
print(result)
```

and the output:

```
The Martian(150 pages)
```

The above code is incomplete. Which of the following alternatives is the correct code to replace the `????` in order to produce the output as shown above?

- (a) `{"__class__": "Book", "__pages": 150, "__title": "The Martian"}`
- (b) `{"__class__": "Book", "title": "The Martian", "pages": 150}`
- (c) `{"title": "The Martian", "pages": 150}`
- (d) `{"__class__": "Book"}`
- (e) None of the above

The function below is used by the following 3 questions:

```
def fun(value, count = 2):
    return value * count
```

Question 10

[1 mark] What output would be generated by the following code fragment?

```
print(fun('a', 3))
```

- (a) 3a
- (b) a3
- (c) aaa
- (d) The program has a runtime error
- (e) 333

Question 11

[1 mark] What output would be generated by the following code fragment?

```
print(fun(count = 4, value = 'a'))
```

- (a) 4444
- (b) 4a
- (c) 4
- (d) aaaa
- (e) The program has a runtime error

Question 12

[1.5 marks] What output would be generated by the following code fragment?

```
print (fun (fun (2, 3)))
```

- (a) 12
- (b) 222
- (c) 6
- (d) 23
- (e) None of the above

The definition of the `Book` class is used by the following 5 questions.

```
class Book:
    def __init__(self, title='', pages = 0):
        self.__title = title
        self.__pages = pages

    def __add__(self, other_book):
        return self.__pages + other_book.__pages

    def __radd__(self, other):
        return self.__pages + other * 2

    def __iadd__(self, other):
        self.__pages += other.__pages
        self.__pages += 1
        return self

    def set_pages(self, value):
        if value > 0:
            self.__pages = value
```

Question 13

[1.5 marks] Which of the following code segments defines a string representation of the `Book` object?

For example, if `book1 = Book('Fluency', 321)`, the statement `print(book1)` prints out `'Fluency(321 pages)'`.

- (a)

```
def __str__(self):
    return '{0}({1} pages)'.format(self.__title, self.__pages)
```
- (b)

```
def __str__():
    return '{0}({1} pages)'.format(__title, __pages)
```
- (c)

```
def __str__(self):
    return self.__title + '(' + self.__pages + ' pages)'
```
- (d)

```
def __str__(self):
    return self.__title + self.__pages
```
- (e) None of the above

Question 14

[1.5 marks] What output would be generated by the following code fragment?

```
book3 = Book('Ready Player One', 100)
book1 = Book('Fluency', 321)
book3 += book1
print(book3)
```

- (a) Ready Player One (742 pages)
- (b) TypeError: unsupported operand type(s) for +: 'Book' and 'Book'
- (c) Ready Player One (100 pages)
- (d) Ready Player One (422 pages)
- (e) Ready Player One (421 pages)

Question 15

[1.5 marks] Which of the following code segment gives a correct implementation to add a **POSITIVE** page number to a `Book` instance? For example, the following code fragment:

```
book1 = Book('Fluency', 321)
book2 = Book('The Martian', 150)
book1.add_pages(-10) #ignored
print(book1)
book2.add_pages(10)
print(book2)
```

will produce the output:

```
Fluency(321 pages)
The Martian(160 pages)
```

- (a)

```
def add_pages(number):
    if number > 0:
        set_pages(__pages + number)
```
- (b)

```
def add_pages(self, number):
    self.set_pages(self.__pages + number)
```
- (c)

```
def add_pages(self, number):
    if number > 0:
        self.set_pages(self.__pages + number)
```
- (d)

```
def add_pages(self, number):
    if number > 0:
        set_pages(self.__pages + number)
```
- (e) None of the above

Question 16

[1.5 marks] Which of the following code segment gives a correct implementation to compare two Book instances by the number of pages? For example, the following code fragment:

```
book1 = Book('Fluency', 321)
book2 = Book('The Martian', 150)
print('book1<=book2:', book1 <= book2)
```

will produce the output:

```
book1<=book2: False
```

- (a) `def __le__(self, other):`
`return self.__pages <= other.__pages`
- (b) `def __less_equal__(self, other):`
`return self.__pages <= other.__pages`
- (c) `def __le__(other):`
`return self.__pages <= other`
- (d) `def __lt__(self, other):`
`return self.__pages < other.__pages`
- (e) None of the above

Question 17

[1.5 marks] Which of the following code segment gives a correct implementation of the method `copy_book` which copy a Book instance to another object? For example, the following code fragment:

```
book1 = Book('The Martian', 150)
book2 = book1.copy_book()
print(book1, book2, book1 is book2)
```

will produce the output:

```
The Martian(150 pages) The Martian(150 pages) False
```

- (a) `def copy_book(self):`
`return (self.__title, self.__pages)`
- (b) `def copy_book(self):`
`return '{0}({1} pages)'.format(self.__title, self.__pages)`
- (c) `def copy_book(self):`
`return Book(self.__title, self.__pages)`
- (d) `def cp(self):`
`return Book(self.title, self.pages)`
- (e) None of the above

The function `exception` below is used by the following 4 questions.

```
def exception (list1, num):
    try:
        for i in range(1, num):
            number = list1[i]//list1[i-1]
            print('{0}/{1}={2}'.format(list1[i],list1[i-1],number), end=' ')
    except ZeroDivisionError:
        print('Division by 0 Error!')
    except IndexError:
        print('Index Error!')
    except:
        print('Unknown Error!')
```

Question 18

[1 mark] What output is produced when the statement `exception([1,2,3,4,5], 3)` is executed?

- (a) `2/1=2 3/2=1`
- (b) `2/1=2 3/2=1 4/3=1`
- (c) `2/1=2 3/2=1 4/3=1 5/4=1`
- (d) `2/1=2`
- (e) None of the above

Question 19

[1.5 marks] What output is produced when the statement `exception(['1',0,3,4,5], 4)` is executed?

- (a) `2/1=2 3/2=1`
- (b) `Unknown Error!`
- (c) `2/1=2 3/2=1 4/3=1`
- (d) `Index Error!`
- (e) None of the above

Question 20

[1.5 marks] What output is produced when the statement `exception([1,0,3,4,5], 4)` is executed?

- (a) `Division by 0 Error!`
- (b) `3/2=1 4/3=1`
- (c) `Division by 0 Error! 4/3=1`
- (d) `0/1=0 Division by 0 Error!`
- (e) None of the above

Question 21

[1.5 marks] What output is produced when the statement `exception([1,2,3], 4)` is executed?

- (a) `2/1=2 3/2=1 Index Error!`
- (b) `2/1=2 3/2=1 Unknown Error!`
- (c) `2/1=2 3/2=1 4/3=1`
- (d) `3/2=1 4/3=1`
- (e) None of the above

Question 22

[1.5 marks] Given the following code fragment, what is the Big-O performance with respect to n ?

```
for i in range(n):
    for j in range(n * n):
        print(i, j)
```

- (a) $O(n^2)$
- (b) $O(n \log n)$
- (c) $O(n^3)$
- (d) $O(n)$
- (e) None of the above

Question 23

[1.5 marks] Given the following code fragment, what is the Big-O performance with respect to n ?

```
sum = 0
for i in range(n):
    for j in range(i):
        sum += 1
```

- (a) $O(n^3)$
- (b) $O(n^2)$
- (c) $O(n \log n)$
- (d) $O(n)$
- (e) None of the above

Question 24

[1 mark] Given the following code fragment, what is the Big-O performance with respect to n ?

```
for i in range(n//2):
    for j in range(n):
        print(j)
    for k in range(n):
        print(k)
```

- (a) $O(n \log n)$
- (b) $O(n)$
- (c) $O(n^2)$
- (d) $O(n^3)$
- (e) None of the above

Question 25

[1 mark] Suppose we have a $O(n)$ sorting algorithm that takes 0.5 seconds to sort 100 items on our computer. About how long will it take to sort 500 items on our computer with the same code?

- (a) 15 seconds
- (b) 2.5 seconds
- (c) 0.5 seconds
- (d) 125 seconds
- (e) None of the above

SECTION B

Answer all questions in this section in the space provided. If you run out of space then please use the Overflow Sheet and indicate in the allotted space that you have used the Overflow Sheet.

Question 26:

[10 marks]

Write a Python class that behaves like the regular string class except that:

- The addition (+) operator returns the sum of the lengths of the two strings (instead of concatenation) if the left and right operands are instances of `MyString`. *For example: the statement `print(x + y)` will print '13' on the computer screen.*
- If the left operand is not an instance of `MyString`, simply add the value of the left operand to the length of the right operand `MyString` object. *For example: the statement `print(2 + x)` will print '7' on the computer screen.*
- The multiplication (*) operator returns the product of the lengths of the two strings if the left and right operands are instances of `MyString`. *For example: the statement `print(x * y)` will print '40' on the computer screen.*
- If the left operand is not an instance of `MyString`, simply multiply the value of the left operand to the length of the right operand `MyString` object. *For example: the statement `print(2 * x)` will print '10' on the computer screen.*

For example, consider the following code fragment:

```
x = MyString('hello')
y = MyString('universe')
print(x)
print(y)
print(x + y) #5 + 8 = 13
print(x * y) #5 * 8 = 40
print(2 + x) #2 + 5 = 7
print(2 * x) #2 * 5 = 10
p = MyString('happy')
q = MyString('boy')
print(p + q)
print(p * q)
print(2 + q)
print(2 * q)
```

The output is:

```
hello
universe
13
40
7
10
happy
boy
8
15
5
6
```

```
class MyString:
    def __init__(self, str=""):
        self.value = str
    def __str__(self):
        return str(self.value)
    def __add__(self, other):
        return len(self.value) + len(other.value)
    def __radd__(self, other):
        return len(self.value) + other
    def __mul__(self, other):
        return len(self.value) * len(other.value)
    def __rmul__(self, other):
        return len(self.value) * other
    def __lt__(self, other):
        return len(self.value) < len(other.value)
```

(10 marks)

Question 27:

[6 marks]

Complete the **get_word_lengths** function which takes a list as a parameter and calculates the lengths of each word in the list. The function should return a list of integers and ignore the word if the length is less than 2. For example, the following code fragment:

```
print(get_word_lengths(['hello', 'good', 'morning', 'i']))
```

will produce the output:

```
[5, 4, 7]
```

a) Note: you must use a for-loop to solve the above problem in part a).

```
def get_word_lengths(my_list):
    result = []
    for x in my_list:
        if len(x) > 1:
            result.append(len(x))
    return result
```

(3 marks)

b) Note: you must use a List Comprehension to solve the above problem in part b).

```
def get_word_lengths2(my_list):
    return [len(x) for x in my_list if len(x)>1]
```

(3 marks)

- Overflow Sheet 1 -

Write the question number and letter next to your answer. You must ALSO indicate in the allotted space that you have used the overflow sheet.

- Overflow Sheet 2 -

Write the question number and letter next to your answer. You must ALSO indicate in the allotted space that you have used the overflow sheet.