

THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2016
Campus: City

COMPUTER SCIENCE

Principles of Computer Science

(Time Allowed: 75 minutes)

Note:

- The use of calculators is NOT permitted.
- Compare the test version number on the Teleform sheet supplied with the version number above. If they do not match, ask the test 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.
- 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.
- Choose the BEST answer for each question based on the material covered in the course.
- There are 25 questions in this test, worth 100 marks in total. Allocate your time appropriately.

Question 1

[4 marks] What is the output of the following code?

```
def foo(n):  
    return 2 * n + 1  
  
my_list = [foo(x) for x in range(1, 10) if x % 2 != 0]  
print(my_list)
```

- (a) [1, 3, 5, 7, 9]
- (b) [3, 7, 11, 15, 19]**
- (c) [2, 4, 6, 8]
- (d) [5, 9, 13, 17]
- (e) None of the above.

Question 2

[4 marks] What is the output of the following code?

```
my_list = [[1, 2], [[3, 4], [5, 6, 7]]]  
for x in my_list:  
    for y in x:  
        print(y, end=' ')
```

- (a) 1 2 3 4 5 6 7
- (b) [1, 2, 3, 4, 5, 6, 7]
- (c) 1 2 [3, 4] [5, 6, 7]**
- (d) [1, 2, [[3, 4], [5, 6, 7]]]
- (e) None of the above.

Question 3

[4 marks] What is the output of the following code?

```
x = [1, 2, 3]  
y = [4, 5, 6]  
z = x + y  
w = x  
x += y  
print(x == z, x is z, w == z, w is z, sep=' ')
```

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

Question 4

[4 marks] What is the output of the following code?

```
x = [1, 2, 3]
y = x
y *= 2
x.reverse()
print(y)
```

- (a) [1, 2, 3]
- (b) [3, 2, 1, 3, 2, 1]
- (c) [1, 2, 3, 3, 2, 1]
- (d) [3, 2, 1]
- (e) None of the above.

Question 5

[4 marks] What is the output of the following code?

```
import copy
a = [ [11], [22], [33] ]
b = a.copy()
c = copy.deepcopy(a)
print (a is b, a[0] is b[0], b is c, b[0] is c[0], sep = ' ')
```

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

Question 6

[4 marks] Which of the following statements are true?

- I. A software object's state is represented by its variables, called data fields. A software object implements its behavior with functions.
- II. Every object is a bundle of variables and related functions. We make an object perform its actions by invoking the functions on that object.
- III. Two objects of the same kind would have the same set of behaviors and same state information.
- IV. The definition of a particular kind of objects is called a class. Once created, an object is an instance of the class.

- (a) I, II and IV
- (b) II, III and IV
- (c) I and II
- (d) I, II, III and IV
- (e) None of the above.

Question 7

[4 marks] Which of the following code segments define the constructor of a `Point` class with all the data fields hidden?

- (a) `class Point:`
 `def init(loc_x, loc_y):`
 `__x = loc_x`
 `__y = loc_y`
- (b) `class Point:`
 `def __init__(self, loc_x, loc_y):`
 `self.x = loc_x`
 `self.y = loc_y`
- (c) `class Point:`
 `def Point(self, loc_x, loc_y):`
 `self.__x = loc_x`
 `self.__y = loc_y`
- (d) `class Point:`
 `def __init__(self, loc_x, loc_y):`
 `self.__x = loc_x`
 `self.__y = loc_y`
- (e) None of the above.

Question 8

[4 marks] Which of the following functions define a string representation of the `Point` object? For example, the following statement `print(Point(10,20))` produces the output of `'(10,20)'`. You can assume that the `Point` class has two hidden data fields, which represent the `x` and `y` coordinates.

- (a) `def __repr__(self):`
 `return 'Point({0},{1})'.format(self.__x,self.__y)`
- (b) `def __str__(self):`
 `return '({0},{1})'.format(self.__x,self.__y)`
- (c) `def __str__():`
 `return '(' + self.__x + ',' + self.__y + ')'`
- (d) `def __repr__():`
 `return '({0},{1})'.format(__x,__y)`
- (e) None of the above.

Question 9

[4 marks] Which of the following functions overwrite the '=' operator that compares the value of two Point objects? Two Point objects are considered to be value equivalent if and only if their corresponding x and y coordinates are the same.

- (a)

```
def __eq__(self, other):  
    return self.__x == other.__x and self.__y == other.__y
```
- (b)

```
def _==(self, other):  
    return self.__x == other.__x and self.__y == other.__y
```
- (c)

```
def __eq__(other):  
    return self.x == other.x or self.y == other.y
```
- (d)

```
def _==(self, other):  
    return self.__x * other.__y == self.__y * other.__x
```
- (e) None of the above.

Question 10

[4 marks] Which of the following functions overwrite the '+=' operator that performs the addition of a Point object to an existing one? The addition of two Point objects is the sum of the two corresponding x and y coordinates. Note that the '+=' represents the in-place addition operator, which does not result in creating a new object after the addition.

- (a)

```
def __add__(self, other):  
    new_x = self.__x + other.__x  
    new_y = self.__y + other.__y  
    return Point(new_x, new_y)
```
- (b)

```
def __iadd__(self, other):  
    self.__x = self.__x + other.__x  
    self.__y = self.__y + other.__y  
    return self
```
- (c)

```
def __+=__(other):  
    self.x = self.x + other.x  
    self.y = self.y + other.y  
    return self
```
- (d)

```
def __iadd__(other):  
    return Point(self.x + other.x, self.y + other.y)
```
- (e) None of the above.

Question 11

[4 marks] Which of the following defines the function 'isNext' that examines whether two Point objects are next to each other? The two Point objects are considered to be in a next relationship, if and only if either x (or y) coordinates of the two are the same and the corresponding y (or x) coordinates differ by 1.

- (a) `def isNext(other):`
 `return self.__x - other.__x == 1 or self.__y - other.__y == 1`
- (b) `def isNext(self, other):`
 `cond = self.__x == other.__x and abs(self.__y - other.__y) == 1`
 `return cond and self.__x - other.__y == 1`
- (c) `def isNext(self, other):`
 `cond1 = self.__x == other.__x and abs(self.__y - other.__y) == 1`
 `cond2 = self.__y == other.__y and abs(self.__x - other.__x) == 1`
 `return cond1 or cond2`
- (d) `def isNext(other):`
 `return abs(self - other) == 1`
- (e) None of the above.

Question 12

[4 marks] Which of the following defines the function 'distance' that calculates the distance between two Point objects? To find the distance between two points (x1, y1) and (x2, y2), use the formula below. Note that the result should be rounded to 2 decimal places. You can assume that the math module has already been imported.

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- (a) `def distance(self, other):`
 `result = math.sqrt((self.__x - other.__x)**2 \`
 `+ (self.__y - other.__y)**2)`
 `return round(result,2)`
- (b) `def __distance(other):`
 `result = math.sqrt((__x2 - other.__x1)**2 \`
 `+ (__y2 - other.__y1)**2)`
 `return round(result,2)`
- (c) `def __dist__(other):`
 `return math.sqrt((__x - other.__x)**2 + (__y - other.__y)**2)`
- (d) `def distance(self, other):`
 `result = (self.__x - other.__x) + (self.__y - other.__y)`
 `return round(result,2)`
- (e) None of the above.

Question 13

[4 marks] Which of the following statements are true?

- I. An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions at runtime. Creating an exception object and handing it to the runtime system is called throwing an exception.
- II. If an exception is not caught in the current function, it is passed to its caller. The process is repeated until the exception is caught or passed to the main function. In other words the exception is handled or the program is disrupted and terminates without further execution.
- III. It is always good to separate the detection of an error (done in a called function) from the handling of an error (done in the calling function).
- IV. Exceptions can alter the flow of control of a program, i.e., when an exception is raised, the program always stops.

- (a) I, II, and IV
- (b) I, II, III, and IV
- (c) I, II, and III**
- (d) II, III and IV
- (e) None of the above.

Question 14

[4 marks] What is the output of the following code?

```
def divide(a, b):
    try:
        result = a / b
        print ("Try-block")
    except:
        result = 'Error in input data'
    except TypeError:
        result = 'Type of operands is incorrect'
    except ZeroDivisionError:
        result = 'Divided by zero'
    return result

x = divide('abc', 0)
print ("Program can continue to run ...")
print(x)
```

- (a) Program can continue to run ...
Type of operands is incorrect
- (b) Try-block
Error in input data
- (c) Program can continue to run ...
Error in input data
- (d) Program can continue to run ...
Divided by zero
- (e) None of these.**

The function divide below is used by the following 2 questions.

```
def divide(a, b):
    try:
        result = a / b
        print ("Try-block finishes")
    except TypeError:
        result = 'Type of operands is incorrect'
    except ZeroDivisionError:
        result = 'Divided by zero'
    except:
        result = 'Other types of errors'
    else:
        print("Result is", result)
    finally:
        print("Executing finally clause")
    return result
```

Question 15

[4 marks] What is the output if the following statements are executed?

```
x = divide(2, 1)
print(x)
```

- (a) Executing finally clause
Divided by zero
Result is 2.0
- (b) Executing finally clause
Type of operands is incorrect
2.0
- (c) Try-block finishes
Result is 2.0
Executing finally clause
2.0
- (d) Try-block finishes
Executing finally clause
2.0
- (e) None of the above.

Question 16

[4 marks] What is the output if the following statements are executed?

```
x = divide([2], 1)
print(x)
```

- (a) Executing finally clause
Other types of errors
- (b) Result is other types of errors
- (c) Try-block finishes
Executing finally clause
Type of operands is incorrect
- (d) Executing finally clause
Type of operands is incorrect**
- (e) None of the above.

Question 17

[4 marks] What is the output of the following code?

```
def checkLevel(level):
    if level < 1:
        raise ValueError('Invalid level:'+ str(level)+'!')
    else:
        print (level)

try:
    checkLevel(-2)
    print ('Can this print statement be reached?')
except ValueError as x:
    print ('Problem: {0}'.format(x))
```

- (a) Problem: Invalid level:-2!**
- (b) -2
Can this print statement be reached?
- (c) Invalid level:-2!
- (d) -2
Problem: Invalid level!
- (e) None of the above.

Question 18

[4 marks] Which of the following statements are true?

- I. If both except and finally blocks are defined, the except block must precede the finally block.
- II. A try block must always contain at least one else or finally block.
- III. For each try block there must be at least one except block defined.
- IV. A try block may be followed by any number of finally blocks.

- (a) I, II, and IV
- (b) I, II, and III
- (c) II, III and IV
- (d) I and III**
- (e) None of the above.

Question 19

[4 marks] What is the content of the 'test_point.txt' file after the following code has been executed?

```
import json
def write(data, filename):
    file = open(filename, 'w')
    str_out = json.dumps(data, indent = 4, sort_keys = True)
    file.write(str_out)
    file.close()

my_data = {'__class__': 'Point', 'y' : 3, 'x' : 2}
file_name = 'test_point.txt'
write(my_data, file_name)
```

- (a) {"__class__": "Point", "y": 3, "x": 2}
- (b) {**
 "__class__": "Point",
 "x": 2,
 "y": 3
 }
- (c) {"__class__": "Point", "x": "2", "y": "3"}
- (d) {
 "__class__": "Point",
 "y": 3,
 "x": 2
}
- (e) None of the above.

Question 20

[4 marks] What is the big-O complexity of the following function?

```
def complexity_01(my_list1):
    n = len(my_list1)
    count = 0
    while n > 0:
        count += 1
        n = n // 2
    return count
```

- (a) $O(n)$
- (b) $O(n^2)$
- (c) $O(n \log(n))$
- (d) $O(n^3)$
- (e) $O(\log(n))$

Question 21

[4 marks] What is the big-O complexity of the following function?

```
def complexity_02(my_list1):
    count = 0
    n = len(my_list1)
    for i in range(1, n):
        for j in range(1, n):
            for k in range(1, n):
                count += my_list1[i]
    return count
```

- (a) $O(n \log(n))$
- (b) $O(\log(n))$
- (c) $O(n^3)$
- (d) $O(n)$
- (e) $O(n^2)$

Question 22

[4 marks] What is the big-O complexity of the following function?

```
def complexity_03(my_list):
    n = len(my_list)
    j = 3
    while j < n:
        print(my_list[j])
        j = j + 1
```

- (a) $O(\log(n))$
- (b) $O(n^3)$
- (c) $O(n^2)$
- (d) $O(n)$
- (e) $O(n \log(n))$

Question 23

[4 marks] What is the big-O complexity of the following function?

```
def complexity_04(my_list1):  
    n = len(my_list1)  
    for i in range(0, n):  
        my_list1[i] *= 2  
    for j in range(0, n):  
        my_list1[j] += 1  
    return my_list1
```

- (a) $O(n^2)$
- (b) $O(n^3)$
- (c) $O(n)$
- (d) $O(n \log(n))$
- (e) $O(\log(n))$

Question 24

[4 marks] The following sets of functions have different Big-O's:

- I) $f(n) = 5n^3 + 10n^2 \log(n) + n^2$
- II) $f(n) = n^4 + 5$
- III) $f(n) = n^2 \log(n) + 10n^5 + 5$
- IV) $f(n) = 1000 + 3 \log(n)$
- V) $f(n) = n^2 + 2n + 100 \log(n)$

Which of the following orderings represents the correct ascending ranking of the above functions' complexities, from smallest Big-O (on the left) to largest Big-O (on the right)?

- (a) V, IV, III, II, I
- (b) IV, V, III, I, II
- (c) II, III, I, V, IV
- (d) IV, V, I, II, III
- (e) II, I, III, V, IV

Question 25

[4 marks] If a is a list, which of the following operations is not $O(1)$?

- (a) `a.pop(0)`
- (b) `a[10]`
- (c) `a.append(0)`
- (d) `a.pop()`
- (e) `a[10] = 6`