

THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2015

Campus: City

COMPUTER SCIENCE

TEST

Principles of Computer Science

(Time Allowed: 75 MINUTES)

Note:

- The use of calculators is NOT permitted.
- Compare the exam version number on the Teleform sheet supplied with the version number above. If they do not match, ask the exam 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.

CONTINUED

Question 1

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

```
my_list = [x+3 for x in range(10) if x%2 == 0]
print(my_list)
```

- (a) [3, 5, 7, 9, 11]
- (b) [3, 6, 9, 12, 15, 18]
- (c) [0, 1, 3, 5, 7, 9]
- (d) [3, 4, 5, 6, 7, 8, 9]
- (e) None of the above.

Question 2

[3 marks] What is the output of the following code? Note that the use of the backslash symbol '\ ' here is for line continuation purpose.

```
words = "Fine and Cloudy"
vowels = 'aeiouAEIOU'
my_words = [words.index(c) for c in words if c not in vowels and \
            words.count(c) == 1]
print(my_words)
```

- (a) ['F', 'n', 'C', ' ', 'd']
- (b) [0, 9, 10, 14]
- (c) [2, 4, 2, 7, 4, 7]
- (d) ['F', 'c', 'l', 'y']
- (e) None of the above.

Question 3

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

```
x = [ 22, 45, 11, 68]
y = x
x += [5]
print (x == y, x is y)
```

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

Question 4

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

```
a = [[2, 4, 6], [1, 3, 7] , [11], [5, 8]]
b = a.copy()
print (a == b, a is b, a[0] is b[0])
```

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

Question 5

[3 marks] When defining a class in Python, which of the following statements is true about the constructor method?

- (a) The constructor method is named as `__init__` with its first parameter usually named as `self`.**
- (b) The constructor method is named as `__init__` with no parameters at all.
- (c) The constructor method is named the same as the class name with no parameters.
- (d) The constructor method is named as `__class__` with its second parameter named as `self`.
- (e) None of the above.

Question 6

[3 marks] Suppose that the definition of the Point class is saved in a file named 'Geometry.py', what statement we should include before creating Point objects in another Python file, e.g., 'test.py'? You can assume that the 'test.py' and 'Geometry.py' are in the same folder.

- (a) `from Point import Geometry`
- (b) `from Geometry import Geometry`
- (c) `from Geometry import Point`**
- (d) `from Point import Point`
- (e) None of the above.

Question 7

[3 marks] Which of the following statements are true in Python?

- I. A private data field of a class can be defined as a variable with two leading underscores.
- II. The value of a private data field can be directly accessed outside of the class.
- III. A class with private data fields must define the accessor (`get`) and mutator (`set`) methods.
- IV. Python will report an error if a private data field of a class is directly accessed by objects outside the class.

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

The partial definition of the Point and Circle classes is used by the following 5 questions.

```
import math
class Point:
    def __init__(self, loc_x, loc_y):
        self.__x = loc_x
        self.__y = loc_y

    def getX(self):
        return self.__x

    def setX(self, loc_x):
        self.__x = loc_x

    def getY(self):
        return self.__y

    def setY(self, loc_y):
        self.__y = loc_y

    def __repr__(self):
        return 'Point({0}, {1})'.format(self.__x, self.__y)

class Circle:
    def __init__(self, x, y, r):
        self.__center = Point(x,y)
        self.__radius = r

    def getCenter(self):
        return self.__center

    def getRadius(self):
        return self.__radius

    def setRadius(self, r):
        self.__radius = r
```

Question 8

[3 marks] Which of the following code segment gives a correct implementation of the method `setCenter` that changes the x, y location of the center of a `Circle` object?

- (a)

```
def setCenter(self, x, y):
    self.center = Point(x, y)
```
- (b)

```
def setCenter(self, x, y):
    self.__center.__x = x
    self.__center.__y = y
```
- (c)

```
def setCenter(x, y):
    self.__center = Point(x, y)
```
- (d)

```
def setCenter(self, x, y):
    self.__center.setX(x)
    self.__center.setY(y)
```
- (e) None of the above.

Question 9

[4 marks] Which of the following code segment gives a correct implementation of the method `__str__` that defines a string representation when printing out a `Circle` object? The format example with a `Circle(10, 20, 6)` object is “Circle with radius 6 at Point(10, 20)”. Note that the use of the backslash symbol ‘\’ here is for line continuation purpose.

- (a) `def __str__(self):`
`return 'Circle with radius {0} at {1}'.format(self.__radius,`
`str(self.__center))`
- (b) `def __str__(self):`
`return 'Circle with radius {0} at {1}'.format(self.__radius, \`
`self.__center.__x, self.__center.__y)`
- (c) `def __str__(self):`
`return 'Circle with radius {0} at Point({1},`
`{2})'.format(self.__radius, \`
`self.__center)`
- (d) `def __str__(self):`
`return 'Circle with radius {0} at`
`Point({1})'.format(self.__radius, \`
`self.__center.getX(), self.__center.getY())`
- (e) None of the above.

Question 10

[3 marks] Which of the following code segment gives a correct implementation of the method `area` that compute the area of a `Circle` object?

- (a) `def area(self):`
`return (math.pi * math.pow(self.radius, 2))`
- (b) `def area(self):`
`return (math.pi * self.__radius * self.__radius)`
- (c) `def __area():`
`return (math.pi * self.radius * self.radius)`
- (d) `def area(self):`
`return (math.pi * self.__radius * 2)`
- (e) None of the above.

Question 11

[3 marks] Which of the following code segment gives a correct implementation of the method `__eq__`, which compares two `Circle` objects? That is, two `Circle` objects are considered to be equal if they have the same area.

- (a) `def __eq__(self, other):`
`return self.area() == other.area()`
- (b) `def __eq__(other):`
`return self.area() - other.area() == 0`
- (c) `def __eq__(other):`
`return self.__area() == other.__area()`
- (d) `def __eq__(self, other):`
`return self.radius == other.radius`
- (e) None of the above.

Question 12

[4 marks] Suppose that the above methods of the `Circle` class have been correctly implemented, what is the output of the following code?

```
a = Circle(10, 20, 6)
b = Circle(200, 345, 6)
c = Circle(24, 56, 20)
b.setCenter(10, 20)
c.setRadius(6)
b.setRadius(12)
print(a is b, a == b, b == c)
```

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

The method `mean` below is used by the following 3 questions.

```
def mean(data):
    try:
        sum = 0
        for element in data:
            sum += element
        mean = sum / len(data)
    except ZeroDivisionError:
        print('Divided by Zero')
    except TypeError:
        print('Invalid Type')
    except:
        print('Error Occurred')
    else:
        print('Result = ' + str(mean))
    finally:
        print('Finalization')
```

Question 13

[3 marks] What is the output if the statement `mean ([])` is executed?

- (a) Divided by Zero**
Finalization
- (b) Divided by Zero
Result = 0
- (c) Invalid Type
Result = Error Occurred
- (d) Result = 0
Finalization
- (e) None of the above.

Question 14

[3 marks] What is the output if the statement `mean([12, 34, 57, 33, 88])` is executed?

- (a) Result = 44.80
- (b) Result = 44.8
Finalization
- (c) Invalid Type
Finalization
- (d) Result is 28
No Error
- (e) None of the above.

Question 15

[3 marks] What is the output if the statement `mean([[11,24,55], [23,56], [2,9,5]])` is executed?

- (a) Divided by Zero
Result = 0
- (b) Error Occurred
Result = Error
- (c) Result = 12
Finalization
- (d) Invalid Type
Finalization
- (e) None of the above.

Question 16

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

```
import json
import io

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

my_dict = {'Jing': '82286', 'Mike': '86133', 'Burkhard': '83705'}
file_name = 'test_contact.txt'
write(my_dict, file_name)
```

- (a) {'Jing': '82286', 'Mike': '86133', 'Burkhard': '83705'}
- (b) {"Mike": "86133", "Jing": "82286", "Burkhard": "83705"}
- (c) {"Burkhard": "83705", "Jing": "82286", "Mike": "86133"}
- (d) {"Burkhard": 83705, "Jing": 82286, "Mike": 86133}
- (e) None of the above.

Question 17

[4 marks] Which of the following would have the greatest complexity?

- (a) $n * (\log 25 * n)$
- (b) $n * \log n + 53$
- (c) $n * (\log n + 25 * n)$
- (d) $n * (\log n * (12 + n) + 5)$
- (e) $n * (n + 53) + 34$

Question 18

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

```
i = 1
while i < n:
    for j in range(1, n):
        print(i, j)
    i = i * 2
```

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

Question 19

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

```
i = 1
for i in range(1, n):
    for j in range(1, i):
        print(i, j)
    i = i * 2
```

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

Question 20

[3 marks] Which of the following is FALSE about linear structures?

- (a) They differ by which end items are added/deleted.
- (b) They have a front end and a back end.
- (c) They are ordered collections of items.
- (d) In Queues, all the adds and deletes occur at the same end.
- (e) None of the above.

Question 21

[3 marks] Which of the following is FALSE about Abstract Data Types?

- (a) They hide the implementation details.
- (b) They are considered different than data structures.
- (c) Their interface methods have a well defined complexity.
- (d) They provide a well defined interface.
- (e) None of the above.

Question 22

[3 marks] Given the following sequence of stack operations, what is top item on the stack when the sequence is complete?

```
M = Stack()
m.push('x')
m.push('y')
m.pop()
m.push('z')
m.peak()
m.push('w')
m.pop()
m.pop()
```

- (a) 'x'
- (b) 'y'
- (c) 'w'
- (d) 'z'
- (e) The stack is empty.

Question 23

[4 marks] Consider the Stack ADT implemented using a Python list (called items) such that push() and pop() are defined as follows:

```
def push(self, item):
    self.items.insert(0, item)

def pop(self):
    return self.items.pop(0)
```

What is the big-O complexity of push() and pop()?

- (a) both push() and pop() are $O(n^2)$
- (b) pop() is $O(1)$ but push() is $O(n)$
- (c) push() is $O(1)$ but pop() is $O(n)$
- (d) both push() and pop() are $O(1)$
- (e) both push() and pop() are $O(n)$

Question 24

[4 marks] Evaluate the following two post-fix expressions:

$$A = 2 \ 3 \ + \ 4 \ / \ 1 \ 6 \ *$$

$$B = 8 \ 2 \ * \ 3 \ +$$

Which of the following statements is true?

- (a) A is not a valid post-fix expression
- (b) The value of A is larger than the value of B
- (c) The value of B is larger than the value of A
- (d) B is not a valid post-fix expression
- (e) The value of A is the same as the value of B

Question 25

[3 marks] If you were implementing a stack using a Python list, which of the following would be FALSE?

- (a) The complexity of Stack operations depends on which end of the list is used.
- (b) Stacks can be implemented using either end of the list.
- (c) None of these.
- (d) Push's and pop's all occur at the same end.
- (e) Stacks are linear structures.

Question 26

[3 marks] Suppose we had the following sequence of numbers: 1, 2, 3, 4, 5 and we pushed all the numbers onto an empty stack and then popped all the numbers off, printing each one as we popped it off the stack. How would the order of the numbers as they were printed compare with the order if we had used a queue instead (i.e., using enqueue for push and dequeue for pop)?

- (a) The numbers for the stack would be printed in reverse order and in normal order for the queue.
- (b) The numbers for both would be printed in normal order.
- (c) It depends on how they were implemented.
- (d) The numbers for both would be printed in reverse order.
- (e) The numbers for the stack would be printed in normal order and in reverse order for the queue.

Question 27

[4 marks] What postfix expression does the following infix expression translate to?

$$2 + 3 - 4 / 3 * 3 + 4$$

- (a) 2 3 4 / 3 3 * - 4 + +
- (b) 2 3 4 3 3 * / - + 4 +
- (c) 2 3 4 3 / 3 * - + 4 +
- (d) 2 3 4 3 / 3 * - 4 + +
- (e) None of the above.

Question 28

[4 marks] Given that the `Stack` and `Queue` classes have been implemented correctly, what is the output of the following:

```
s1 = Stack()
q = Queue()

s1.push(1)
s1.push(2)
s1.push(3)

while not s1.is_empty():
    (q.enqueue(s1.pop()))

print(q.dequeue(), end = ' ')
print(q.dequeue(), end = ' ')
print(q.dequeue(), end = ' ')
```

- (a) 2 1 3
- (b) 3 2 1
- (c) 3 1 2
- (d) 1 2 3
- (e) 1 3 2

Question 29

[3 marks] As far as our in-class discussions of possible implementations of these ADTs, which of the following statements about the Big-O complexity of our ADT operations would be FALSE.

- (a) The Big-O complexity must be the same for both pop and push stack operations.
- (b) The Big-O complexity can be different for both enqueue and dequeue queue operations.
- (c) None of these.
- (d) The Big-O complexity can be the same for both enqueue and dequeue queue operations.
- (e) The Big-O complexity is not the same for both the `add_front` and `remove_rear` dequeue operations.

Question 30

[3 marks] As far as our in-class discussion of the implementation of circular queues, which of the following statements would be FALSE. (Note: we are only talking about where the indexes point between operations, not during an operation.)

- (a) When the queue is empty, the *front* index is one greater (modulo `MAX_QUEUE`) than the *back* index.
- (b) When the queue is full, the *front* index is one greater (modulo `MAX_QUEUE`) than the *back* index.
- (c) When the queue is neither empty nor full, the *front* index points to the item to be dequeued.
- (d) When the queue is neither empty nor full, the *back* index points to where the next item is to be queued.
- (e) None of the above.

Question 31

[3 marks] If you were writing a program to check whether an HTML document had correctly matching begin and end tags, which would be the most appropriate ADT to use?

- (a) Stack
- (b) Deque
- (c) Queue
- (d) Circular Queue
- (e) All of these.

Rough Working – This page will not be marked

