THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2014
Campus: City

COMPUTER SCIENCE

Computer Science Fundamentals – SOLUTIONS

(Time allowed: 50 minutes)

NOTE

- Attempt all questions.
- All questions are worth equal marks
- Use of calculators is NOT permitted.
- You may detach the answer sheet from the questions and keep the questions.
- **Use the answer sheet to give your answers.**

SECTION A
MULTIPLE-CHOICE QUESTIONS

Circle the preferred choice on the **answer sheet** provided.

1. What does the following expression evaluate to?

```
2 + 3 - 4 // 3 * 3 % 4
```

(a). 5
(b). 4
(c). 3
(d). 2
(e). 1

2. Consider the block of code below, where variables athlete, beautician and carpenter each have integer values. Under which condition will the value in the variable athlete be printed?

```
if athlete < beautician:
      if beautician < carpenter:
            print(carpenter)
      else:
            print(beautician)
elif athlete < carpenter:
      print(carpenter)
else:
      print(athlete)
```

(a). When athlete is 3, beautician is 2 and carpenter is 1
(b). When athlete is 1, beautician is 2 and carpenter is 3
(c). When athlete is 1, beautician is 3 and carpenter is 2
(d). When athlete is 2, beautician is 1 and carpenter is 3
(e). Under no circumstances, because variable athlete's value can never be printed by this code

3. What is the output of the following code?

```
list_1 = [1, -5, 2, 0, -2, -3]
list_2 = [0, -2, 2, 4, -5, -3]

result = 0
for i in range(0, len(list_1)):
      if list_1[i] < list_2[i]:
            result += 1
print(result)
```

(a). 0
(b). 1

(c). 2

(d). 3

(e). 4

4. What is the output of the following code?

```
n = 5
for x in range(0, n):
      s = ''
      for y in range(0, x + 1):
            if (x == 0 or x == n - 1 or
               y == 0 or y == n - 1 or
               x == y):
                  s += 'x'
            else:
                  s += '-'
      print(s)
```

(a).

```
xxxxx
xx--x
x-x-x
x--xx
xxxxx
```

(b).

```
xxxx
xx--
x-x-
x--x
xxxx
```

(c).

```
xxxxx
x---x
x---x
x---x
xxxxx
```

(d).

```
xxxxx
x--x
x-x
xx
x
```

(e).

```
x
xx
x-x
x--x
xxxxx
```

5. What is the output of the following code?

```
def my_function_a(x):
      i = 0
      while len(x) > i + 1:
            if x[i] > x[i + 1]:
                  return i + 1
            i += 1
      return -1

def my_function_b(x):
      p = x[0]
      for e in x:
            if p > e:
                  return False
            p = e
      return True

y = [4, 6, 7, 4, 8, 5]
print(my_function_b(y), my_function_a(y))
```

(a). `False 3`

(b). `True 3`

(c). `False 4`

(d). `True 4`

(e). `True 5`

6. Which one of the options could be the output of the following code?

```
dictionary = {'a' : 23, 'b' : 12}
dictionary['a'] = 20
dictionary[23] = 'a'
dictionary['c'] = 7
print(dictionary)
```

(a). `{'b': 12, 'a': 23, 'c': 7}`

(b). A runtime error will occur before the print function is executed.

(c). `{'a': 20, 'a': 23, 'b': 12, 'c': 7}`

(d). `{7 : 'c', 23 : 'a', 20 : 'a', 12 : 'b'}`

(e). `{'c': 7, 'a': 20, 'b': 12, 23 : 'a'}`

7. The following code is supposed to calculate the frequency of characters in the sentence, but it has an error. What is the problem?

```
sentence = "hello"
dictionary = {}
for letter in sentence:
      if dictionary.get(letter, 0) == 0:
            dictionary[letter] = dictionary[letter] + 1
print(dictionary)
```

(a). The sentence contains duplicate letters and the letter is used as the key, but each key in a dictionary must be unique, so the code will generate a runtime error.

(b). The code `dictionary.get(letter, 0) == 0` performs the wrong comparison, so the code contains a logic error that will cause it to print the wrong result.

(c). The use of `dictionary[letter] =` causes a problem because a value cannot be assigned an entry in the dictionary when the key doesn't exist, so a runtime error will be generated .

(d). The `dictionary[letter] + 1` expression causes a problem because the key doesn't exist, so accessing `dictionary[letter]` will result in a runtime error.

(e). The dictionary.get(letter, 0) causes a problem because the key doesn't exist, so the get() function will generate a runtime error.

8. Assume that a text file called `sample.txt` contains 3 lines of text as indicated below:

```
This is
a small
text file.
```

What is the output produced by the following program?

```
f = open('sample.txt')
contents = f.readline()
print(len(contents))
```

(a). 8

(b). 3

(c). 2

(d). 24

(e). 26

9. Assume that a text file called `sample.txt` contains 3 lines of text as indicated below:

```
This is
a small
text file.
```

Which one of the following programs would print out:

```
['This', 'is', 'a', 'small', 'text', 'file.']
```

(a).
```
f = open('sample.txt')
contents = [f.read()]
f.close()
print(contents)
```

(b).
```
f = open('sample.txt')
contents = f.readlines()
f.close()
print(contents)
```

(c).
```
f = open('sample.txt')
line = f.readline()
contents = []
while line is not '':
   contents += [line]
   line = f.readline()
f.close()
print(contents)
```

(d).
```
f = open('sample.txt')
contents = []
for word in f:
   contents += [word]
f.close()
print(contents)
```

(e).
```
f = open('sample.txt')
contents = f.read()
contents = contents.split()
f.close()
print(contents)
```

10. What is the output produced by the following program?

```
import json
my_list = [1, 2, 3]
my_dict = {'a' : my_list}
my_json = json.dumps(my_dict)
my_data = json.loads(my_json)
print(type(my_data))
```

(a). `<class 'list'>`

(b). `<class 'module'>`

(c). `<class 'dict'>`

(d). `<class 'json'>`

(e). `<class 'str'>`

11. What is the output produced by the following program?

```
list_a = [1, 2, 3]
list_b = list_a
list_b = list_b + [4]
print(list_a == list_b, list_a is list_b)
```

(a). `False False`

(b). `True False`

(c). `False True`

(d). `True True`

(e). `No output because a runtime error will be generated`

12. What is the output produced by the following program?

```
list_a = [1, 2, 3]
list_b = list_a
list_c = [list_a, list_a]
list_d = [list_b] + [list_b]
list_b.append(4)
print(list_d[0] is list_c[0], list_c == list_d, list_c is list_d)
```

(a). `False False False`

(b). `True True False`

(c). `False True False`

(d). `False False True`

(e). `True False False`

13. What value is stored in `result` after executing the following code?

```
result = ''
try:
      num = 100 / 0
      result += 'a'
except ZeroDivisionError:
      result += 'b'
except:
      result += 'c'
finally:
      result += 'd'
```

(a). `'bd'`

(b). `'bcd'`

(c). `'b'`

(d). `'bc'`

(e). `'cd'`

14. What value is stored in `result` after executing the following code?

```
result = ''
try:
      num = int('Hello')
      result += 'a'
      try:
            num = 200 / 0
            result += 'b'
      except ValueError:
            result += 'c'
```

CONTINUED

```
      except:
            result += 'd'
      finally:
            result += 'e'
except ZeroDivisionError:
      result += 'f'
finally:
      result += 'g'
```

(a). `'cefg'`

(b). `'eg'`

(c). `'g'`

(d). `'deg'`

(e). `'ceg'`

15. Consider the following definition of the `test` class.

```
class test:
      def __init__(self):
            self.questions = {}
            self.counter = 1

      def add_question(self, question, marks):
            self.questions[self.counter] = (question, marks)
            self.counter += 1

      def foo(self):
            bar = 0
            for item in self.questions.values():
                  bar += item[1]
            return bar
```

What is the output of the following code?

```
t = test()
t.add_question('What does 2 + 3 evaluate to?', 1)
t.add_question('What does 3 * 4 evaluate to?', 3)
t.add_question('What does 3 // 2 evaluate to?', 3)
print(t.foo())
```

(a). `4`

(b). `6`

(c). `1`

(d). `7`

(e). `3`

16. Consider the following definition of the `test` class.

```
class test:
      def __init__(self):
            self.questions = {}
            self.counter = 1

      def add_question(self, question, marks):
            self.questions[self.counter] = (question, marks)
            self.counter += 1

      def __str__(self):
            result = ''
            for key in sorted(self.questions.keys()):
                  item = self.questions[key]
                  result += 'Q{0}: {1}'.format(key, item[0])
            return result
```

What is the output of the following code?

```
t = test()
t.add_question('What does 2 + 3 evaluate to?', 5)
print(t)
```

(a). `Q1: (What does 2 + 3 evaluate to?, 5)`

(b). `Q0: What does 2 + 3 evaluate to?`

(c). `Q1: What does 2 + 3 evaluate to?`

(d). `Q0: ('What does 2 + 3 evaluate to?', 5)`

(e). `{1: ('What does 2 + 3 evaluate to?', 5)}`

CONTINUED

SECTION B
SHORT-ANSWER QUESTIONS

Answer in the gap in the **answer sheet** provided.

17. Give the Big-O complexity of the `bigO1()` function below:

```
def bigO1(my_list1):
            n = len(my_list1)
            i = 5
            while i < n:
                  helperO1(my_list1, i)
                  i += 5
            return my_list1

def helperO1(my_list1, position2):
            value = my_list1[position2]
            i = position2 - 3
            while i > -1 and i < position2:
                  average = int((value + my_list1[i]) / 2)
                  my_list1[i] = average
                  i += 1
```

SOLUTION: O(n)

18. Give the Big-O complexity of the `bigO2()` function below:

```
      def bigO2(my_list1):
            my_list2 = []
            n = len(my_list1)

            start = 0
            halfway = n // 2
            end = n
            for i in range(halfway):
                  element_to_check = my_list1[i]
                  if helperO2(my_list1, element_to_check, halfway, end):
                        my_list2.append(my_list1[i])

            return my_list2

      def helperO2(my_list1, element_to_check, start, end):
            i = start
            while i < end:
                  if element_to_check == my_list1[i]:
                        return True
                  i += 1
            return False
```

SOLUTION: O(n^2)

19. Give the Big-O complexity of the bigO3() function below:

```
def bigO3(my_list1):
    my_list2 = []
    for i in range(0, len(my_list1)):
        my_list2.append(my_list1[i])

    my_list2.sort()

    number = 0
    count = 0
    for i in my_list1:
        if i == my_list2[number]:
            count += 1
        number += 1
    return count
```

SOLUTION: O($nlog(n)$)

20. Give the Big-O complexity of the bigO4() function below:

```
def bigO4(my_list1):
    my_list2 = []
    n = len(my_list1)
    for i in range(n):
        j = i
        while j > 0:
            my_list2.append(my_list1[j])
            j = j // 2

    return my_list2
```

SOLUTION: O($nlog(n)$)

SPARE PAGE FOR ROUGH WORKING
(Will not be marked)

ANSWER SHEET FOLLOWS

ANSWER SHEET

ID: ....

SURNAME: ____

FORENAME(S): ____________________________________

UPI: ____

SECTION A: Circle your chosen response. If you need to change your answer, mark a cross through your wrong answer and circle your final choice.

1.	a	b	c	d	e
2.	a	b	c	d	e
3.	a	b	c	d	e
4.	a	b	c	d	e
5.	a	b	c	d	e
6.	a	b	c	d	e
7.	a	b	c	d	e
8.	a	b	c	d	e
9.	a	b	c	d	e
10.	a	b	c	d	e
11.	a	b	c	d	e
12.	a	b	c	d	e
13.	a	b	c	d	e
14.	a	b	c	d	e
15.	a	b	c	d	e
16.	a	b	c	d	e

CONTINUED

ANSWER SHEET

ID: ....

SECTION B: Write your answer in the gap provided

17.

18.

19.

20.