THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2014
Campus: City

COMPUTER SCIENCE

Principles of Programming

(Time Allowed: TWO HOURS)

SAMPLE SOLUTIONS

Note:

- The use of calculators is NOT permitted.
- You should separate the Section A Question Booklet from the Section B Question/Answer Booklet. You may keep the Section A Question Booklet. You must hand in the Section B Question/Answer booklet and the Teleform sheet.
- Compare the exam 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 and Student ID should be entered left aligned. If your name is longer than the number of boxes provided, truncate it.
- Answer **Section A** on the Teleform answer sheet provided. For Section A, use a dark pencil to mark your answers in the answer boxes on the Teleform sheet. Check that the question number on the sheet corresponds to the question number in this question/answer book. Do not cross out answers on the Teleform sheet if you change your mind. You must completely erase one answer before you choose another one. If you spoil your sheet, ask the supervisor for a replacement. There is one correct answer per question.
- Answer Section B in the space provided in the Section B Question/Answer Booklet.
- Attempt all questions. 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.

SECTION A

MULTIPLE CHOICE QUESTIONS

There is only **one** correct answer for each question. For each question, choose the **best** answer according to the information presented in lectures. Select your preferred answer on the Teleform answer sheet provided by shading in the appropriate box.

Question 1

[2 marks] What is the output of this code?

```
x = 5
y = 10
temp = y
x = y
y = x
print("x =", x, "y =", y)
```

(a) **x = 10 y = 10**

(b) x = 10 y = 5

(c) x = 5 y = 5

(d) x = 5 y = 10

(e) None of the above.

Question 2

[2 marks] What is the output of this code?

```
my_string = "Happy Birthday to you!"
print(my_string[3:12:2])
```

(a) pitd

(b) yBrh

(c) pyBrh

(d) **p itd**

(e) None of the above.

Question 3

[2 marks] What is the output of this code?

```
my_string = "How are you feeling?"
print(my_string[5::-1])
```

(a) a woH

(b) **ra woH**

(c) gnile

(d) gnilee

(e) None of the above.

Question 4

[1.5 marks] What possible values could the following command print?

```
print(random.randint(0,6))
```

(a) **0, 1, 2, 3, 4, 5, 6**

(b) 1, 2, 3, 4, 5

(c) 0, 1, 2, 3, 4, 5

(d) 1, 2, 3, 4, 5, 6

(e) None of the above.

Question 5

[2 marks] What will the following function print?

```
def temp2():
    num_list = []
    for num in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
        if num <= 4 or num >= 8 and num % 2 == 0:
            num_list += [num]
    print(num_list)
```

(a) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

(b) **[1, 2, 3, 4, 8, 10]**

(c) [2, 4, 8, 10]

(d) [2, 4, 6, 8, 10]

(e) None of the above.

Question 6

[2 marks] What will the following function print?

```
def temp3():
    sum = 0
    for i in range(0,5):
        for j in range (i,5):
            sum += j
    print(sum)
```

(a) 40

(b) 70

(c) 80

(d) 50

(e) None of the above.

Question 7

[2 marks] What will the following code assign to the variable `var1`?

```
var1 = input("This is a word")
```

(a) The string “`input("This is a word")`”.

(b) **Whatever the user types before pressing the `<Enter>` key.**

(c) The first character the user types.

(d) The string “`This is a word`”.

(e) None of the above.

Question 8

[2 marks] When `my_list = [1, 2, 3, 4]` what will the following function print out?

```
def mystery(my_list):
    out_list = [my_list[0]]
    for i in range(1, len(my_list)):
        out_list += [out_list[len(out_list)-1] + my_list[i]]
    print(out_list)
```

(a) **[1, 3, 6, 10]**

(b) [1, 12, 123, 1234]

(c) [10, 20, 30, 40]

(d) [1, 4, 10, 20]

(e) None of the above.

Question 9

[2 marks] Which of the following is the correct notation to indicate the fourth element of a sequence?

(a) `a(4)`

(b) `a{4}`

(c) **`a[3]`**

(d) `a[4]`

(e) None of the above.

Question 10

[2 marks] Which of the following lines of code creates a one-element tuple?

(a) `one_tuple = [7]`

(b) **`one_tuple = 7,`**

(c) `one_tuple = {7}`

(d) `one_tuple = (7)`

(e) None of the above.

Question 11

[2 marks] What will the following code print?

```
print("@" * 3)
```

(a) `@@@`

(b) `333`

(c) `@3`

(d) A syntax error

(e) None of the above.

Question 12

[2 marks] Which of the following function calls produces the same sequence as `range(10)`?

(a) `range(1, 10)`

(b) `range(10, 0, 0)`

(c) **`range(0, 10, 1)`**

(d) `range(10, 1)`

(e) None of the above.

Question 13

[2 marks] Consider the following function named `mystery`:

```
def mystery(number):
    counter = 0
    while number > 1:
        number = number // 2
        counter = counter + 1
    print(counter)
```

What is the output of the following call to the `mystery` function above?

```
mystery(11)
```

(a) **3**

(b) 11

(c) 5

(d) 1

(e) None of the above.

Note: Use the following code fragment to answer the next three questions:

```
a = input("Enter a: ")
b = input("Enter b: ")
c = input("Enter c: ")
if a > b:
    print("Who?")
    if b > c:
        print("Up")
    else:
        print("Down")
elif b > c:
    print("What?")
    if a >= c:
        print("1")
    elif a < b:
        print("2")
    elif c == b:
        print("3")
else:
    print("Where?")
    if a == b:
        print("apple")
    else:
        print("banana")
print("DONE")
```

Question 14
[1.5 marks] What will be printed when a =3, b=4, c=5?

(a) Who?
Up
DONE

(b) What?
1
Done

(c) **Where?**
banana
DONE

(d) Where?
apple
DONE

(e) None of the above.

Question 15

[1.5 marks] What will be printed when a =5, b=4, c=3?

(a) Where?
banana
DONE

(b) Who?
Up
DONE

(c) Where?
apple
DONE

(d) Who?
Down
DONE

(e) None of the above.

Question 16

[1.5 marks] What will be printed when a =3, b=3, c=5?

(a) Where?
apple
DONE

(b) Who?
Up
DONE

(c) Where?
banana
DONE

(d) Where?
banana
DONE

(e) None of the above.

Question 17

[2 marks] Consider the following function named `func`:

```
def func(x):
    total = 0
    for i in range(x):
        total = total + i * (i-1)
    return total
```

What is the output of the following code fragment?

```
print(func(2))
```

(a) 2

(b) -1

(c) 1

(d) **0**

(e) None of the above.

Question 18

[2 marks] Are we able to change the value of x[1] as follows?

```
x = [1, (2, 3), 4]
x[1] = (5, 6)
```

(a) **Yes, because we are changing the contents of a list.**

(b) No we cannot modify the contents of a tuple.

(c) Yes, because we are changing the contents of a tuple.

(d) No, we cannot modify the contents of a list.

(e) None of the above.

Question 19

[2 marks] Consider the following two functions:

```
def f1(a):
    return 2 * a - 1

def f2(b):
    return f1(b) - 3
```

What is the output of the following code fragment?

```
print(f2(5))
```

(a) 5

(b) **6**

(c) 2

(d) 9

(e) None of the above.

Question 20

[2 marks] Which of the following is the correct boolean expression to test for the following condition assuming that x and y are int variables?

x is a value between, but not including, 25 and 65, or y is not equal to 50.

(a) `x >= 25 and x <= 65 and y != 50`

(b) `x < 25 and x > 65 or !(y == 50)`

(c) **`x > 25 and x < 65 or y != 50`**

(d) `x > 25 and x < 65 or y not = 50`

(e) None of the above.

Question 21

[2 marks] What is the resulting value and type of evaluating each of the following expressions?

```
x = 4.0 / 10.0 + 2.0 * 3
y = not 4.0 < 10.0 or 5 > 1.2345
```

(a) x is 6.0 (float), y is False (bool)

(b) x is 6.4 (float), y is False (bool)

(c) x is 6 (int), y is True (bool)

(d) **x is 6.4 (float), y is True (bool)**

(e) None of the above.

Question 22

[2 marks] Consider the following code fragment that takes a test score (an integer between 0 and 100 inclusive) as input and prints out a grade. What would the following code fragment print out if the user types in 89 at the prompt?

```
testscore = int(input("Enter a test score: "))
if testscore >= 90:
    grade = "A"
if testscore >= 80:
    grade= "B"
if testscore >= 70:
    grade= "C"
if testscore >= 60:
    grade= "D"
else:
    grade = "F"
print(grade)
```

(a) **D**

(b) B

(c) C

(d) A

(e) None of the above.

Question 23

[2 marks] What is the output of the following code fragment?

```
t = (1, 2, 4, 3)
print(t[1 : -1])
```

(a) `(1, 2, 4)`

(b) **`(2, 4)`**

(c) `(1, 2)`

(d) `(2, 4, 3)`

(e) None of the above.

Question 24

[2 marks] Consider the following tuple named `t`:

```
    t = (1, 2, 4, 3)
```

Which of the following statements will cause an error to occur?

(a) `print(max(t))`

(b) `print(len(t))`

(c) `print(t[3])`

(d) **`t[3] = 45`**

(e) None of the above.

Question 25

[2 marks] Consider the following dictionary **d**:

```
d = {"john":40, "peter":45}
```

What will happen when retrieving a value using `d["susan"]`?

(a) It is executed fine, and it returns -1.

(b) **Since "susan" is not a key in the dictionary, Python raises an error.**

(c) It is executed fine, and it returns None.

(d) Since "`susan`" is not a value in the dictionary, Python raises an error.

(e) None of the above.

Question 26

[2 marks] Consider the following function named **`f`**:

```
def f(number_list):
    return max(number_list), min(number_list)
```

What is the output of the following code fragment?

```
x , y = f([4, 2, 6, 3, 9])
print(x,y)
```

(a) **9 2**

(b) 2 9

(c) 9

(d) 18

(e) None of the above.

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK.

THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2014
Campus: City

Computer Science

Principles of Programming

(Time Allowed: TWO HOURS)

SECTION B Question/Answer Booklet

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.

Surname:	
First Name(s):	
Student ID:	
Login Name (UPI):	

MARKERS ONLY

Q1 – Q26 (/50)	**Q29** (/10)	**TOTAL** (/100)
Q27 (/10)	**Q30** (/10)	
Q28 (/10)	**Q31** (/10)	

Question 27 [10 marks]

Complete the function named `days_till_end()` that takes 3 parameters (`initial_rabbits`, `percent_integer`, and `max_number`) and returns the number of days until rabbits take over Albert Park. We are given the number of initial rabbits at the beginning of the first day in the parameter "`initial_rabbits`". The rabbits increase by `percent_integer` each day (for instance 2 means 2%) and Albert Park can only hold "`max_number`" of rabbits.

```
def days_till_end(initial_rabbits, percent_integer, max_number):
    """
    >>> days_till_end(100, 2, 200)
    36
    """


    days = 0
    rabbits = initial_rabbits
    while rabbits < max_number:
        rabbits = rabbits * (1 + percent_integer/100)
        days += 1
    return days
```

(10 marks)

Question 28 [10 marks]

Write a function named `set_up_translators()` which inputs a list of English and Spanish tuples, e.g. [("one", "uno"), ("two", "duo") …] and returns two dictionaries. The first dictionary is named `spanish_to_english` (where the index is the Spanish word) and the second dictionary is named `english_to_spanish` (where the index is the English word).

```
def set_up_translators(word_pair_list):
    """
    >>> set_up_translators([("one", "uno"), ("two", "duo")])
    ({'uno': 'one', 'dos': 'two'}, {'one': 'uno', 'two': 'dos'})

    spanish_to_english = {}
    english_to_spanish = {}

    for element in word_pair_list:
        e, s = element
        spanish_to_english[s] = e
        english_to_spanish[e] = s
    return spanish_to_english, english_to_spanish
```

(10 marks)

Question 29 [10 marks]

Complete the function named `give_me_half()` which takes two parameters – a boolean and a list of strings and returns a list that contains either every even element of the parameter list or every odd element of the list. If the boolean parameter is True, the function returns a list containing all the elements with an even index, otherwise it returns a list containing all the elements with an odd index.

```
def give_me_half(even, my_list):
    """
    >>> give_me_half(True,["pig","ape","dog","cat"])
    ['pig', 'dog']
    """
    if even:
        return my_list[0:len(my_list):2]
    return my_list[1:len(my_list):2]
```

(10 marks)

Question 30 [10 marks]

Complete the following program which allows the user to enter two integer numbers, calculates the sum of all integers between these two numbers inclusive (the sum should include the user-entered numbers themselves), and prints the sum as in the example below. You may assume that the first number is smaller than the second number. A sample interaction would look like:

```
This program sums all the integers between two integers.
Enter the first integer: 3
Enter the second integer: 5
The sum of integers between 3 and 5 is 12
```

```
print ("This program sums all the integers between two integers.")
sum = 0

first = int(input("Enter the first integer: "))
second = int(input("Enter the second integer: "))

for number in range(first, second + 1):
    sum += number

print("The sum of integers between", first, "and", second,
"is", sum)
```

(10 marks)

Question 31 [10 marks]

Complete the function `count_frequency()` below which prompts the user to enter a filename and counts the occurrences of words in the text file then displays the result. Assume the words are case-sensitive.

Sample contents of input file (words.txt):

```
The woods are lovely dark and deep
But I have promises to keep
And miles to go before I sleep
And miles to go before I sleep
```

Sample Call to function:

```
>>> count_frequency()
Enter a filename: sample.txt
The 1
miles 2
deep 1
I 3
are 1
...
```

```
def count_frequency():

    input_filename = input("Enter a filename: ")
    input_file = open(input_filename, "r")
    contents = input_file.read()

    input_file.close()
    word_list = contents.split()
    freq = {}

    for word in word_list:
        if word in freq:
            freq[word] += 1
        else:
            freq[word] = 1

    for key, value in freq.items():
        print (key, value)
```

(10 marks)

OVERFLOW PAGE

(If you have used this page, please indicate clearly under the relevant question that you have overflowed to this page)

OVERFLOW PAGE

(If you have used this page, please indicate clearly under the relevant question that you have overflowed to this page)