THE UNIVERSITY OF MANITOBA | **Name** | **Student Number**

COMP1012: Computer Programming for Scientists and Engineers

Midterm In Class Exam (A02—Andres)

2014 Mar-5, 11:30 pm — Time: 45 minutes

Instructions:

1. Answer all questions on this paper. For multiple choice questions, circle the letter of the ***best*** or most complete choice. For short answer questions, write your answer in the space provided.
2. Extra work space is available on the last page.
3. You will find a Python Guide along with your midterm; ask if you don't have one. You may ***not*** use your own copy. No other aids (such as calculators or cell phones) are permitted.
4. You have 45 minutes to complete the exam.

Marks for Part 1	Part 2A	Part 2B	Part 3	Total
/ 4	/ 4	/ 4	/ 4	/16

Part 1: Predict the output [4 x 1 mark]

In each row of the table below, mentally execute the code on the left and enter the expected output in the box on the right. Each table row is separate. Use the space below for scrap work.

	Code Fragment	*Expected output*

A.	What is printed by
	`print 'COMP1012'[-7 : 3]` ?
		`OM or 'OM'`
		
		`½ if slightly off`

B.	What is printed by
	`print range(5,3,-1)` ?
		`[5, 4]`
		
		`½ if brackets are missing or ()`

C.	What is printed by `print 0 < 0 - 2 % 3` ?	`False`
D.	What is printed by `print 5 or 4 or 5 / 3` ?	`5`

Work space:

Part 2: Write a program [Total 8 marks]

2A. [4 marks] A friend claims that the following equation is correct, but you are not sure.

$$-\log(1-x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \frac{x^5}{5} \ldots \quad \text{if } |x| < 1$$

You write a Python script to check the claim by printing out the following table. Your script calls the function `calcSeries` to evaluate the series. (You will write **`calcSeries`** ***on the next page.***) Show the code to produce this table.

```
TABLE TO COMPARE TWO FUNCTIONS
x      -log(1-x)       series sum
 0.1  0.10536051566  0.10536051566
 0.2  0.22314355131  0.22314355131
 0.3  0.35667494394  0.35667494394
 0.4  0.51082562377  0.51082562377
 0.5  0.69314718056  0.69314718056
 0.6  0.91629073187  0.91629073187
 0.7  1.20397280433  1.20397280433
 0.8  1.60943791243  1.60943791243
 0.9  2.30258509299  2.30258509299
```

Details:

- 11 decimal places are shown.

```
def calcSeries(xx,eps) :  # write it on next page
…
# put script to produce table here
import math  # not assigned mark
print "TABLE TO COMPARE TWO FUNCTIONS"
print "   x      -log(1-x)       series sum"
for num in range(1,10) :
    xx = num * 0.1
    print "%4g %14.11f %14.11f" % (xx, -math.log(1 - xx),
                                   calcSeries(xx,1.e-13))
```

A. Heading—approximate layout (lose 1/2 for missing line, or printed after loop, or /t instead of \t; exact spacing not checked)
B. for loop with 9 entries (lose 1/2 for using while loop or having float counter for xx or range(0.1, 1, 0.1); warning if loop variable is changed in loop)
C. function, series call (lose ½ if one missing or wrong, or bad value of eps—should be from 1.e-20 to 1.e-11—or non-existent function like math.ln, or wrong arguments)
D. format line of output (lose ½ for r or s or d instead of f, or wrong number of decimals, or too small width like %2.11f though %.11f OK)
E. no positive mark; deduct ½ for at least two lines badly indented

For marker use only	
Item	Mark
A	
B	
C	
D	
E	
Sum	

Part 2: Write a program [Total 8 marks]

2B. [4 marks] A friend claims that the following equation is correct, but you are not sure.

$$-\log(1-x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \frac{x^5}{5} \ldots \quad \text{if } |x| < 1$$

Write a function `calcSeries` to evaluate the infinite sum above, using the standard approach taught in this course.

Details:

- `xx` is the value corresponding to x.
- Assume `eps` is a small positive number; add all the terms in the series that are greater than `eps` in absolute value, and only those terms.
- Return the value of the series sum to the calling code.
- Do ***not*** print any output from this function.
- You do ***not*** need to fill in a doc string and you do ***not*** need to check parameter values.

```
def calcSeries(xx,eps) :
    assert abs(xx) < 1 # assertion not marked
    count = 0
    total = 0.0
    product = xx
    term = xx
    while abs(term) > eps :
        count += 1
        total += term
        product *= xx
        term = product / (count + 1)
    return total
```

A. Correct use of count, total, product, term and parameters (deduct ½ for no product or vastly different names or redefined xx or eps).
B. while loop (deduct ½ for missing or misplaced abs because xx could be negative, or wrong comparison, or term initialized to 0 so it fails first time, or eps changed in this function).
C. "total" initialization, update and return (needs 2 of the 3 for ½ mark).
D. computation of term (deduct ½ if term defined from previous term without getting rid of previous divisor, or use of ** with power)
E. no positive mark; deduct 1 if work is incorrectly divided between the main script and this function.

For marker use only	
Item	Mark
A	
B	
C	
D	
E	
Sum	

Part 3: Circle the letter of the *best* answer, or provide the required answer [4 x 1 mark]

A. Which of the following formatted prints would produce the given output?

```
Output: '-1.251000      '
```

a) `print "Output: '%-15e'" % (-1.251)`
b) `print "Output: '%-15f'" % (-1.251)`
c) `print "Output: '%-15g'" % (-1.251)`
d) `print "Output: '%15.3f'" % (-1.251)`
e) `print "Output: '%15f'" % (-1.251)`

B. After the following three lines are executed, what is the value of `list2[-2]` ?

```
list1 = ['5' , [''] , () , (1,)]
list2 = []
for xx in list1:
    if bool(xx):
        list2 += [xx]
```

a) `' 5 '`
b) `['']`
c) `()`
d) `(1,)`
e) None of these because `list2[-2]` is not a valid expression.

C. Which of the following statements about tuples and lists is true?

a) Tuples and lists are very similar, but a list can be changed and a tuple cannot.
b) Tuples and lists are very similar, but a tuple can be changed and a list cannot.
c) A tuple cannot contain a list as an entry.
d) A list cannot contain a tuple as an entry.
e) Tuples and lists differ only in the type of brackets they use, "( )" for tuples, "[ ]" for lists.

D. Using good coding practices and the same rules as QuizMaster, write a Python expression to evaluate this math expression, assuming math has already been imported:

$$\log_{10}\left(\frac{10^{\sin(4)+\pi}}{5+\ln|-a|}\right)$$

Put expression here

```
math.log10(10**(math.sin(4) + math.pi )/(5 + math.log(abs(-aa)) ) )
```

This page is intentionally left blank except for the header and this message.