THE UNIVERSITY OF MANITOBA
COMP1012: Computer Programming for Scientists and Engineers
Midterm In Class Exam (A01—Andres)
2013 Oct-29, 8:30 am

Name	Student Number

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.
5. Marks total to 16. Marks for each question are shown in the heading.

Marks for Part 1	Part 2A	Part 2B	Part 3	Total
/ 4	/ 5	/ 3	/ 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 3 // 2 + 3. // 1  ?`	`4.0`
B.	What is printed by `print [0, 1, 2, 3][-3:4]  ? ?`	`[1, 2, 3]`
C.	What is printed by `print range(1,-3,-2)  ?`	`[1, -1]`
D.	What is printed by `print "Tue, 2013-Oct-29".split('-')[0] ?`	`'Tue, 2013'`

Work space:

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

COMP1012: Computer Programming for Scientists and Engineers

Midterm In Class Exam (A01—Andres)

2013 Oct-29, 8:30 am — Time: 45 minutes

Part 2: Write a program [8 marks]

Write a program in two parts. In part A, define a function that sums an infinite series to evaluate a function. In part B, write statements to use this function to print a table.

2A. [5 marks] Define a function `sinh(xx,eps)` to evaluate the *hyperbolic sine function* sinh(x) by approximately summing this infinite series:

$$\sinh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \ldots$$

Details:

- Use the `def` statement below.
- `xx` is the value corresponding to x; convert it to a `float` in case it isn't one already.
- `eps` is a small positive number. Use an `assert` statement to cause your function to fail if `eps` is not positive.
- 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.
- Make sure you ***do*** fill in a doc string for the function.

```
def sinh(xx,eps) :
    """Evaluate hyperbolic sine at xx, using terms bigger than
    eps"""
    xx = float(xx)
    assert eps > 0, "eps argument must be positive float"
    count = 0
    term = xx
    total = 0.0
    xsq = xx * xx

    while abs(term) > eps :
        total += term
        count += 1
        term = term * xsq / (2. * count) / (2. * count + 1.)

    return total
```

A. Doc string
B. Checks/conversion on xx, eps
C. while loop
D. total initialization, update and return
E. computation of term

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

2B. [3 marks] Write a script to print a table of sinh values, as shown below. Call the function you wrote in Question 2A to evaluate sinh.

- Use 10^{-10} as the second argument in calling sinh.
- Use a `for` loop to print the six rows.
- Use a tab character to line up columns.
- Show the sinh value to 11 decimal places.

```
x       sinh(x)
  0     0.00000000000
0.2     0.20133600254
0.4     0.41075232580
0.6     0.63665358206
0.8     0.88810598218
  1     1.17520119364
```

Put your script here:

```
print "x  \tsinh(x)"
for count in range(6) :
    xx = count * 0.2
    sinhx = sinh(xx, 1.e-10)
    print "%3g \t%.11f" % (xx,sinhx)
```

```
A. Heading with tab
B. for loop and xx
C. sinh and print
```

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 [2 x 1 mark + 2 marks]

A. Which of the choices below is *NOT* a valid way of printing the following line?

```
She said, "I can't."
```

a) `print '''She said, "I can't." '''`
b) `print """She said, "I can't." """`
c) `print 'She said, "I ' "can't." '"'`
d) `print 'She said, "I can't"'`
e) `q1, q2 = "'", '"'; print 'She said, I %scan%st.%s' % (q2, q1, q2)`

B. Which of the following functions evaluates $n!$ (that is, n factorial)? Assume `math` has been imported.

a)
```
def fac(nn) :
    fac = nn
    while nn > 0 :
        fac *= nn
        nn = nn - 1
    return fac
```

b)
```
def fac(nn) :
    fac = 1
    while nn > 1 :
        fac *= nn
        nn = nn - 1
    return fac
```

c)
```
def fac(nn) :
    fac = 1
    for jj in range(nn) :
        fac *= jj
    return fac
```

d)
```
def fac(nn) :
    fac = math.factorial(nn)
    return
```

C. [2 marks] Using good coding practices, and the same rules as QuizMaster write a Python expression to evaluate this mathematical expression, assuming `math` has already been imported:

$$\left\lfloor \left(\frac{10 - \pi}{\cos(2)} \right) \cdot (5 - \tan(x)) \right\rfloor$$

Put expression here

```
math.floor((10. - math.pi) / math.cos(2.) * (5. - math.tan(xx)))
```

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