THE UNIVERSITY OF MANITOBA
COMP1012: Computer Programming for Scientists and Engineers
Midterm In Class Exam (A01/A02—Andres/Amiri)
2013 February 28, 2:30 pm

Name	Student Number

Time: 65 minutes

Instructions:

1. Answer all questions on this paper. For multiple choice questions, circle the letter of the ***best*** choice. For short answer questions, write your answer in the space provided.
2. Extra work space is available on the last page.
3. No aids (such as calculators or cell phones) are permitted.
4. You have 65 minutes to complete the exam.
5. Marks total to 20. Marks for each question are shown in the heading.
6. You should find a Python Guide along with your midterm; ask for one if you don't have one. You may ***not*** use a copy you printed yourself.

Marks for Part 1	Part 2A	Part 2B	Part 3	Total
/ 5	/ 5	/ 5	/ 5	/20

Part 1: Predict the output [5 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**3 % 4 + 2  ?`	`5`
B.	What is printed by `print 2 and 1 - 2 != 3  ?`	`True`
C.	What is printed by `print range(-2,4,2)  ?`	`[-2, 0, 2]`
D.	What is printed by `print "alphabet"[-6:4]  ?`	`"ph"`

E.	What is printed by
	`print [jj + 2 * 2`
	`  for jj in [1, 3, 4, 5] ]  ?`
		`[5, 7, 8, 9]`

Work space:

Part 2: Write a program [10 marks]

Write a program in two parts. In part A, define a function that sums an infinite series to evaluate a cosine function. In part B write statements to use this function to find the height of a cliff.

2A. [5 marks] Define a function `cos(xx,eps)` to evaluate the following infinite series for the trigonometric cosine function:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \ldots$$

Details:

- Use the def statement below.
- `xx` is the angle in radians at which you will evaluate the function. Assume it is a `float`.
- `eps` is a small number—add all the terms in the series that are greater than `eps` in absolute value, and only those terms. Assume it is a positive `float`.
- Return the value of the cosine function to the calling code.
- Do ***not*** print any output from this function.
- You do ***not*** have to have a doc string and comments.

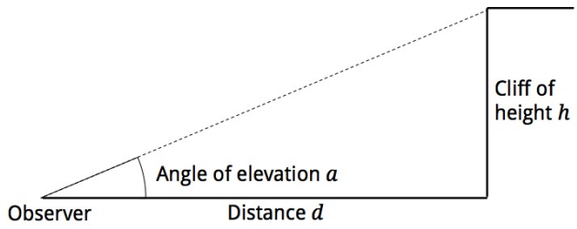
```
def cos(xx,eps) :
    count = 0
    term = 1.0
    total = 0.0
    xsq = xx * xx

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

    return total
```

2B. [5 marks] Imagine a situation where you look up at a cliff and try to estimate its height from its angle of elevation, as in the diagram at right. Using trigonometry,

$$h = d\frac{\sqrt{1-(\cos a)^2}}{\cos a}$$

Write a series of Python instructions to evaluate this formula for several angles and produce a table of heights that looks as close as possible to the following output:

```
ESTIMATING CLIFF HEIGHTS AT 1000 m
Angle [deg]   Angle [rad]   Height [m]
      5           0.087          87.5
     10           0.175         176.3
     15           0.262         267.9
     20           0.349         364.0
     25           0.436         466.3
     30           0.524         577.4
     35           0.611         700.2
     40           0.698         839.1
     45           0.785        1000.0
```



Details:

- Use a for loop to produce angles ranging from 5 to 45 degrees as shown. Convert each angle to radians using the transformation $a_{rad} = a_{deg} \cdot \pi/180$.
- You may import `math` and use `math.pi` and `math.sqrt`, but not `math.cos`.
- Use the formula for h above, and call the `cos` function you defined in 2A. Use 10^{-10} as a second parameter when calling `cos`.
- Follow programming standards for variable names and units.
- You do not need any other output besides what is shown.

```
dd = 1000. # [m] distance from cliff
print "ESTIMATING CLIFF HEIGHTS AT %g m" % dd
print "Angle [deg]   Angle [rad]   Height [m]"

for aDeg in range(5,46,5) : # [deg] angle of elevatioon
    aRad = aDeg * math.pi / 180. # [rad] angle of elevation
    cosine = cos(aRad, 1.e-10) # [-] cos of angle
    hh = dd * math.sqrt(1.0 - cosine**2) / cosine
    print "%6g %14.3f %14.1f" % (aDeg, aRad, hh)
```

Part 3: Circle the letter of the *best* answer, or provide the required answer [5 x 1 mark]

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

```
Output: |-0.142857142857     |
```

a) `print "Output: |%-20e|" % (-1./7)`
b) `print "Output: |%20f|"  % (-1./7)`
c) `print "Output: |%-20g|" % (-1./7)`
d) `print "Output: |%20r|"  % (-1./7)`
e) `print "Output: |%-20s|" % (-1./7)`

B. After the following three lines are executed, what is the value of `list2[3]`?

```
list1 = ['Picasso', (2, 5), 3, 17.2, True]
list2 = list1
list1[1], list1[3] = list1[3], list1[1]
```

a) `3`
b) `17.2`
c) `'Picasso'`
d) `(2, 5)`
e) none of these because `list2[3]` is not a valid expression

C. Which of the following **`for`**-loops would print the *smallest* number of asterisks?

a) `for jj in range(40) :  print "*",`
b) `for jj in range(-40) :  print "*",`
c) `for jj in range(0,80,2) :  print "*",`
d) `for jj in range(80,0,-2) :  print "*",`
e) `for jj in range(100,140) :  print "*",`

D. Which of the following Python expressions acts as `True` when a boolean value is needed?

a) `1 < 50 < 50.0`
b) `[]`
c) `bool('False')`
d) `0.0e20`
e) `list("tabulate")[-5:3]`

E. Using good coding practices, write a Python expression to evaluate this mathematical expression, assuming `math` has already been imported:

$$y \cdot (10^5) - e^x - (\lceil x \rceil - b)$$

Put expression here

```
yy * 1.0e5 - math.exp(xx) - (math.ceil(xx) - bb)
```

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