Name	Student Number

THE UNIVERSITY OF MANITOBA
COMP1012: Computer Programming for Scientists and Engineers
Midterm In Class Exam (A01—Andres)
2014 October 21, 8:30 am

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 2 - 4**2 / 3 ?`	-3 Test 3 operators, all in
B.	What is printed by `print tuple(range(-2))   ?`	() ½ if []
C.	What is printed by `print 4 or 5 != 5 / 2 ?`	4 Definition of or
D.	What is printed by `print 1 < 2 >= 3 / 3 ?`	True

Work space:

Part 2: Write a program [Total 8 marks]

2A. [4 marks] Write a complete program that generates and prints random lowercase letters between 'a' to 'z' until the last 4 characters printed form a valid word. The details are shown below, followed by two sample outputs, with different random characters:

- Include any imports needed by your program.
- Print the heading and final line of output as shown. Be sure to print the final period.
- Obtain a tuple of valid words by calling `getWords()`. Assume this function is already written; you do ***not*** have to write it.

```
GENERATE RANDOM STREAM

g j a i v y s a z e h r u q p i m m z l k h c y v y c h w j v v g n h i p s

The final word is hips.
```

```
GENERATE RANDOM STREAM

x g z z g y z d a i m u c n j k p m n y n s k x q n m n a j b k a t j q h z n n p j
i q g i i p s y k e y q s m n z s v f d o c k

The final word is dock.
```

```
import random
def getWords():
    …
print "GENERATE RANDOM STREAM\n"
words = getWords()
text = "    "
while not text[-4:] in words :
    char = chr(ord('a') + random.randrange(26))
    print char,
    text += char
print "\n"
print "The final word is %s." % text[-4:]
```

A. Heading / import

B. while loop / getWords

C. random letter / word

D. Output line, print letters

E. bad indentation, poor variables

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

Part 2: Write a program [Total 8 marks]

2B. [4 marks] The following series evaluates the function shown, for any x.

$$\frac{e^x - 1}{x} = 1 + \frac{x}{2!} + \frac{x^2}{3!} + \frac{x^3}{4!} + \frac{x^4}{5!} + \frac{x^5}{6!} + \ldots$$

Write a function `reducedExp` to evaluate the infinite sum above, using the standard approach taught in this course (NO FACTORIAL FUNCTIONS).

Details:

- `xx` is the value corresponding to x; `xx` could be positive or negative or 0.
- Assume `small` is a small positive number; add all the terms in the series that are greater than `small` 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 reducedExp(xx,small) :
    term = product = 1.0
    total = 0.0
    count = 0
    while abs(term) > small :
        total += term
        count += 1
        product = product * xx / (count + 1.)
        term = product
    return total
```

.

A. 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 missing :, or …).

B. “total” initialization and update; deduct ½ if updated after computing term, since term hasn’t been tested yet; also check initial values to ensure first term is included

C. computation of term (initialization, update, use of count; deduct ½ for power of xx or if there is a chance of an int divide)

D. return with correct value, ½ for return, ½ for value, unless it is completely ridiculous (like return “Done”).

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

THE UNIVERSITY OF MANITOBA
COMP1012: Computer Programming for Scientists and Engineers
Midterm In Class Exam (A01—Andres)
2014 October 21, 8:30 am

Name	**Student Number**

Time: 45 minutes

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

A. Given the following lines have just been executed, which of the options below creates a list of just the positive even numbers in seq1?

```
seq1 = [-3,  4, -4, -5,  4,  0,  1, -4,  3, -3]
seq2 = []
```

a) `for jj in seq1:  seq2.append(jj * (jj % 2 == 0) * (jj > 0))`
b) `for jj in seq1:  seq2  = [jj] * (jj % 2 == 0) * (jj > 0)`
c) `for jj in seq1:  seq2 += [jj] * (jj % 2 == 0) * (jj > 0)`
d) `for jj in seq1:  seq2 += [jj] * ((jj % 2 == 0) or  (jj > 0))`
e) `for jj in seq1:  seq2  = [jj] * ((jj % 2 == 0) and (jj > 0))`

B. Given the following function definitions, which of the given choices produces the value 2 ?

```
def f(xx) : return xx - 1
def g(xx) : return xx * 2
def h(yy) : return yy // 2
```

a) `h(f(g(5)))`
b) `f(h(g(5)))`
c) `f(g(h(5)))`
d) `g(f(h(5)))`
e) `g(h(f(5)))`

C. Which of the following statements about tuples, lists and strings is false, assuming `seq` is a tuple, a list or a string that is large enough to make the expressions valid?

a) `seq[3]` is the third item if `seq` is any these data types.
b) `seq[-2]` is the second last item if `seq` is any of these data types.
c) `bool(seq[3:3])` is `False` for any of these data types.
d) `2 * seq` is twice the size of seq for any of these data types.
e) `seq += seq` doubles the size of seq for any of these data types.

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:

$$\left\lfloor \left\lceil \left(10 + \tan\left(4 - \pi\right)\right)^{\frac{b}{3}} \right\rceil \right\rfloor$$

Put expression here

```
math.floor(math.ceil((10. + math.tan(4. - math.pi))**(bb / 3.)))
```