Please fill in your Student Number and Name.

Student Number : ___________________________________

Name:

____

____

Student Number:

____

University of Cape Town ~ Department of Computer Science

Computer Science 1018F ~ 2009

June Exam

Question	Max	Internal	External	Question	Max	Internal	External
1	10			7	6		
2	8			8	14		
3	8			9	20		
4	12						
5	12						
6	10						
				TOTAL	**100**		

Marks : 100

Time : 180 minutes

Instructions:

a) Answer all questions.

b) Write your answers in pen in the spaces provided.

c) Show all calculations where applicable.

Question 1 [10]

a) Python has many strengths as a programming language but it does have some weaknesses. List (and explain) one of these weaknesses. [1]

Efficiency – a Python program is usually about 10x slower than the equivalent C++ or Java program [1]

b) Describe three major ways in which Python is different from Java and also three ways in which it is similar. [6]

Different:

Python uses whitespace layout as a means of demarcating code blocks (unlike {} in Java) [1]

Python datatypes are dynamically typed (unlike Java which has static typing) [1]

Python has more powerful native data-types, such as Dictionary and List which support elements of different type. [1]

Similar:

Both are fundamentally object-oriented procedural languages [1]

They are both strongly typed [1]

Neither language is typically compiled directly from high level to machine code [1]

c) What is the difference between mutable and immutable datatypes? Provide two examples of each. [3]

Mutable datatypes allow in-place change of their contents, while immutable structures do not (a new variable must be created) [1] . mutable – list, dictionary, set.[1] immutable – tuple, string and basic datatypes (float, integer etc) [1]

Question 2 [8]

a) Given a list:

```
captains = ['Kirk', 'Picard', 'Sisko', 'Janeway', 'Archer',
'Kirk'].
```

What would the `newcaptains` list look like after each of the following operations: [5]

```
i. newcaptains = captains[1:-1]
```

newcaptains = [‘Picard’, ‘Sisko’, ‘Janeway’, ‘Archer’]

```
ii.newcaptains = captains
   newcaptains.delete('Kirk')
```

newcaptains = [‘Picard’, ‘Sisko’, ‘Janeway’, ‘Archer’, ‘Kirk’]

```
iii.newcaptains = [captains.index(c)  for c in captains]
```

newcaptains = [0, 1, 2, 3, 4, 0]

```
iv.newcaptains = ', '.join(captains)
```

newcaptains = ‘Kirk, Picard, Sikco, Janeway, Archer, Kirk’

```
v. newcaptains = captains.insert(2, 'Pike')
```

newcaptains = [‘Kirk’, ‘Picard’, ‘Pike’, ‘Sisko’, ‘Janeway’, Archer’, ‘Kirk’]

b) Given a list of strings, called `inlist` (e.g., `inlist = ['I', 'you', 'we', 'she', 'them']`), write a code snippet that will check if each element is less than three characters in length and return a list, called `outlist`, with true (len < 3) or false (len >= 3) in the corresponding positions (e.g., `outlist = [True, False, True, False, False]`). Code this, using: [3]

i. a `for` loop

```
outlist = []
for s in inlist:
   outlist.append(len(s) < 3)
```

ii. list comprehension

```
outlist = [len(s) < 3 for s in inlist]
```

c) Write Python code to convert a nested list, called `inlist`, of the form: `inlist = [[key1, value1], [key2, value2], [key3, value3], …]` into a dictionary, `outdict`, where the first element of each sublist is the key and the second element is the corresponding value. If a duplicate key occurs later in the list, its value should overwrite the existing value. [2]

```
outdict = {} [1/2]
for (key, val) in inlist: [1]
   outdict[key] = val [1/2]
```

Question 3 [12]

The following is the declaration for a class that implements a matrix:

```
Class Matrix():
      """A standard two-dimensional array of floating point
      numbers, as used in Linear  Algebra."""
```

a) Write an initialisation function that creates a matrix of size n x m (where n and m are input parameters, both of which default to 1). All the elements of the matrix should be set to zero, initially. [4]

```
def __init__(self, n = 1, m = 1):  [1/2]
  """Initialise a zero-filled matrix to size n x m""" [1/2]

  self.rows = n                  [1/2]
  self.cols = m                  [1/2]
  self.mat = []                  [1/2]
  for i in xrange(n):            [1/2]
     self.mat.append([0.0]*m)  [1]
```

b) Write a method for the Matrix class that will determine if the matrix is symmetric. This means that the element at position (x,y) is equal to the element at position (y,x). Your method should return a boolean value of True if the matrix is symmetric and False otherwise. Note: in order to be symmetric a matrix must first be square (n == m). [4]

```
def symmetric(self):  [1/2]
   """Test the matrix for symmetry""" [1/2]

   if not(self.rows == self.cols):          [1/2]
      return False
  for x in xrange(self.rows):              [1/2]
    for y in xrange(self.cols):          [1/2]
      if not(self.mat[x][y] == self.mat[y][x]): [1/2]
         return False                      [1/2]
return True                                [1/2]
```

c) Write a method, `add`, for the Matrix class that takes two matrices as input, adds them together if they are of the same size, and places the result in the current object. This involves adding together the corresponding elements from both matrices. The method should exit without doing anything if the matrices are not of equal size. [4]

```
def add(self, A, B):  [1/2]
   """Add two matrices together""" [1/2]

   if A.rows == B.rows and A.cols == B.cols:          [1/2]
     self.__init__(A.rows, A.cols)        [1/2]
     for x in xrange(A.rows):              [1/2]
       for y in xrange(B.cols):          [1/2]
         self.mat[x][y] = A.mat[x][y] + B.mat[x][y] [1]
```

Question 4 [12]

You are given a file that has been compressed using run-length compression. In this form of data compression repeated symbols in a sequence are encoded with the number of repetitions and the symbol being repeated. For instance, the stream of numbers ‘0 0 0 0 10 10 17’ would be encoded as ‘4 0 2 10 1 17’. Note that a single symbol still needs to be preceded by a ‘1’. Assuming that you have a file that has already been compressed (encoded), your task is to uncompress (decode) it.

a) Write a method, called `DecodeLoad`, that takes a file name as parameter (with a default of ‘decode.txt’). The method should open the file and read the contents into a list of integers, called `stream`, which is then returned by the method. Hint: there is a space between each integer in the file and you can use this to split them. You should use exceptions to check for file IO errors and return an empty list if an error occurs. [5]

def DecodeLoad(filename = 'decode.txt'): [1]

```
 """Load a run-length compressed file and
 return the contents as a list of integers.""" [1/2]


 try:
    f = open(filename, 'r')             [1/2]
 except IOError:                        [1/2]
    print 'unable to open file'
    return []                           [1/2]
 buf = f.read()                         [1/2]
 stream = buf.split(' ')                [1/2]
 stream = [int(s) for s in stream]      [1/2]
 return stream                          [1/2]
```

b) Write a method, called `decode`, which takes the integer list created in part (a) as a parameter and converts it into its uncompressed form. This uncompressed version must then be returned by the method as a list of integers. Hint: if `l = ['a']` then `l * 4 = ['a', 'a', 'a', 'a']` in Python. [5]

```
def Decode(incode):  [1/2]
   """Decode a run-length compressed stream of integers.”””  [1/2]
```

```
  for pos in range(0, len(incode), 2):      [1]
    rep = incode[pos]                      [1/2]
    sym = incode[pos+1]                    [1/2]
    outcode.extend([sym]*rep)            [1 1/2]
return outcode                             [1/2]
```

c) The data compression ratio is a measure of the effectiveness of a compression scheme. It is defined as the ratio of the size of uncompressed file to the size of the compressed file. So, a scheme which reduced a file from 10Mb down to 2Mb would have a data compression ratio of 5. Write a method, called `CompRatio`, which takes two lists of integers, uncompressed and compressed, and calculates and returns the compression ratio as a floating point number. [2]

```
def CompRatio(uncomp, comp):                              [1/2]
  return float(length(uncomp)) / float(length(comp))  [1 ½]
```

Question 5 [8]

You are provided with the following Visual Python code:

```
from visual import *

ball = sphere(pos=(-5,0,0), radius=0.5, color=color.red)
wallR = box(pos=(6,0,0), size=(0.2,4,4), color=color.green)
wallL = box(pos=(-6,0,0), size=(0.2,4,4), color=color.green)
dt = 0.05
ball.velocity = vector(2,0,0)
while (1==1):
    rate(100)
    ball.pos = ball.pos + ball.velocity*dt
    if ball.x > wallR.x:
        ball.velocity.x = -ball.velocity.x
        if ball.x < wallL.x:
            ball.velocity.x = -ball.velocity.x
```

a) There is a semantic error in this algorithm. What is it and how would you fix it? [1]

The final if statement is incorrectly indented and will never return true [1/2]. Simply shift the if statement one tab stop to the left in line with the previous if statement [1/2]

b) Describe the output produced by the correct (debugged) code. [3]

The program creates two vertical green walls (boxes) [1]. A red ball appears next to the left wall [1] and then moves horizontally backwards and forwards between the two walls (as if it were bouncing between them) [1]

c) Describe two ways in which the ball could be slowed down or sped up. [2]

There are actually three ways: alter dt, alter the balls velocity or change the rate. [2]

d) How you would go about adding a trail behind the ball? [2]

Add a curve object in the setup section [1] and then append the current position of the ball to its list of vertex positions. [1]

Question 6 [10]

a) Black Box and Glass Box are two code testing methodologies. Explain the approach behind each scheme. [4]

Blackbox testing: we pass known inputs with know out values into the program, If the correct outputs are produced for all selected inputs, we assume the program is free of errors. [2]

Glass Box Testing: we trace selected inputs through all possible execution paths in the program, testing for correctness of each path for each set of input. [2]

b) What is a unit test? Explain how unit tests can be combined with equivalence testing to define a (fairly robust) method for testing code. [4]

A unit is the smallest piece of code that we can execute that does something useful in our program [1]

A unit test is the process of running different inputs through our unit and seeing if the outputs are what we anticipated. [1]

Equivalence testing: break range of input values into groups with similar behaving (equivalence class)

We then choose test data for our units from these classes, choosing representative values near the boundaries and some from the interior of each equivalence class. [2]

c) Why do we need to 'test for failure' when composing test cases for our code? [2]

We want our code to fail in a predictable way for invalid output. Users can input both valid and invalid data - so we must ensure correct behaviour in either case. [2]

Question 7 [6]

Consider the following Python code:

```
import re
text = "It was a cold, blustery day. The wind whistled \
        through a crack in the attic wall, making the \
        candle gutter."
regExpression = r"\b[c|b]\w+\b"
r = re.compile(regExpression, re.MULTILINE)
```

a) What is the function of the `re.MULTILINE` flag? [1]

Allows correct matching when newlines are present in string [1]

b) What output will the following statement yield: [3]

```
print r.findall(text)
```

['cold', 'bluster', 'crack', candle']   [3]

c) How would you modify the regular expression to match words which started with a lower case vowel? [2]

regExpression = r"\b[aeiou]\w+\b" [2]

Question 8 [14]

a) Given the decimal number 212.63:

i. Convert it to binary. Show full working and assume 8 bits for the integer part and 4 bits for the fractional part. [4]

212/2 = 106 r 0

106/2 = 53 r 0

53/2 = 26 r 1

26/2 = 13 r 0

13/2 = 6 r 1

6/2 = 3 r 0

3/2 = 1 r 1

1/2 = 0 r 1

212: ---> 11010100 [2] for working/[1] for answer only.

*0.63*2 = 1.26 ---> 1*

*0.26*2 = 0.52 ---> 0*

*0.52*2 = 1.04 ---> 1*

*0.04*2 = 0.08 ---> 0 [2]*

ii. Convert it to octal. [2]

(011)(010)(100).(101)(0--)

324.50 [1] integer part/ [1] fractional part

b) What real number does the following IEEE 32-bit float represent? [3]

```
1 11011011 01000100000000000000000
```

S = -1

exponent = 11011011 = 128+64+16+8+2+1 = 219-127 = 92 [1]

mantissa = 1 + 0.25 + 1/64 = 1.265625 [1]

*final = - (1.265625*2^92) [1]*

c) Given the following Boolean algebra function F:

$$F = \overline{(A.\bar{B} + \bar{B}.C)}$$

i. Write down the Boolean truth table for the expression. [3]

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

[3]

ii. Use the rules of Boolean algebra to simplify this expression i.e reduce the number of Boolean operations required. [2]

F = comp(comp(B).[A+C])

= B + comp(A+C)

= B + comp(A).comp(C) [2]

Question 9 [20]

Bridging the Generation Gap

Teenagers communicate electronically using a language that adults often do not understand. You have been asked to investigate a solution to translate a message in SMS language into regular English. For example, “u r gr8 dude c u l8r” would be translated into “you are great dude see you later”. Assume there is no punctuation and words are separated by spaces.

You have decided that your solution will be based on the use of a translation table such as that shown below.

u	you
r	are
l8r	later
gr8	great
c	see

a) Describe the algorithm you will use to perform the translation. [4]

create a new empty string

scan through the old string word by word

for each word, check through the table for which translation candidate matches at that point

if there is a match, copy the replacement into the new string,

otherwise copy the old word into the new string

b) Describe the data type you will use to store the translation table. [2]

2 dimensional array of strings, dictionary, etc.

c) Describe 2 classes that you could use in an object-oriented solution and what the purpose of each is. [4]

driver class, to start program

interface class, to perform input and output

table class, to store translation table and do lookups

translation class, to perform translation

d) Write the code for the **translate** method that performs the translation and returns the translated string. Assume the message to be translated is passed as a string parameter and the translation table is an instance variable that already contains the table listed above. [10]

```
translate(message):
 nStr = ""
 for word in message.split():
  try:
   nStr += dict[word] + " "
  except KeyError:
   nStr += word + " "
 return nStr

or

translate(message):
 nStr = ""
 for word in message.split():
  if dict.has_key(word):
   nStr += dict[word] + " "
  else:
   nStr += word + " "
 return nStr

or

translate_ugly(message):
 nStr = ""
 for word in message.split():
  if word in dict.keys():
    nStr += dict[word] + " "
  else:
    nStr += word + " "
return nStr
```