THE UNIVERSITY OF WESTERN AUSTRALIA

Computer Science and Software Engineering

SEMESTER 1, 2015 EXAMINATIONS

CITS1401
PROBLEM SOLVING AND PROGRAMING

FAMILY NAME:____________________________ GIVEN NAMES: ______________________

STUDENT ID:									SIGNATURE:________________________

This Paper Contains: **11** pages **(including title page)**
Time allowed: **2:10 hours (including reading time)**

INSTRUCTIONS:

Answer all questions.
Write your answers in the spaces provided on this paper.
No other paper will be accepted for the submission of answers.
All questions have equal marks.
Total marks = 90.

PLEASE NOTE

Supervisors Only - Student left at:

This page has been left intentionally blank

Q1. Arithmetic

A point on a plane can be represented as a pair of numbers denoting its Cartesian coordinates. A triangle whose base is aligned with the x-axis can be represented by three points (x1,0), (x2,0) and (x3,y3).

(a) Define the following function: **3 marks**

```
def area(x1, x2, x3, y3):
# returns the area of the triangle (x1, 0), (x2, 0) and (x3, y3).
# e.g. area(0, 10, 0, 10) = 50
```

(b) Define the following function. **5 marks**

```
def inner(x1, x2, x3, y3):
# returns the coordinates of another triangle whose vertices are at the mid-points of the
edges of the input triangle. The coordinates are returned as two tuples;  the first
contains the three x values and the second contains the corresponding y values
# e.g. inner(0,10, 0, 10) = ((5.0, 0.0, 5.0), (0.0, 5.0, 5.0))
```

(c) Define the following function: **2 marks**

```
def flip(x1, x2, x3, y3):
# returns the coordinates of another triangle flipped on the horizontal axis. The
coordinates are returned similar to the input arguments in a tuple.
# e.g. quart(0, 10, 0, 10) = (0, 10, 0, -10)
```

Q2. Boolean and testing

(a) Define the following function **3 marks**

def inside(x1, y1, r1, x2, y2, r2):
# tests if circle with center x1,y1 and radius r1 is fully inside the second circle with center x2,y2 and radius r2 e.g. inside(2, 2, 4, 3, 3, 5) = False.

(b) Define the following function **4 marks**

def one(b1, b2, b3):
# one takes three Booleans and returns True if and only if exactly one of them is False. Otherwise it returns False, e.g. one(True, False, False) = False.

(c) Define the following function **3 marks**

def fits(b1, b2, n):
# Takes two Booleans and an integer. Returns True if there are exactly n Trues e.g. fits(True, False, 1) = True.

Q 3. String processing

(a) Define the following function **6 marks**

```
def countWords(str):
# takes a parameter of type string and returns a list of unique words in the list followed
# by their number of occurrences
# e.g. count(“the best of the best”) = [‘the’, 2, ‘best’, 2, ‘of’, 1]
```

(b) Define the following function **4 marks**

```
def nameDomain(email):
# takes a parameter string containing an email address of the form
# firstName.lastName@domain and returns the first name, last name and domain in a
# list, e.g. nameDomain(‘ajmal.mian@uwa.edu.au’) =  [‘ajmal’, ‘mian’, ‘uwa.edu.au’]
```

Q 4. List iteration

(a) Define the following function **4 marks**

def accum(xlist):
# takes a parameter xlist (a list containing integers) and returns another list containing the cumulative sum of the elements of xlist, e.g. accum([1, 2, 3, 4, 5]) = [1, 3, 6, 10, 15]

(b) Define the following function that performs the opposite operation **6 marks**

def decum(ylist):
# takes a list of integers calculated by the above function accum (in Q 4 (a)) and returns the original list
# e.g. decum([1, 3, 6, 10, 15]) = [1, 2, 3, 4, 5]

Q 5. List iteration

(a) Define the following function **4 marks**

```
def mix(listA, listB):
# takes two lists of equal length and returns a list of lists of the corresponding elements
# of listA and listB, e.g. mix([‘a’, ‘b’, ‘c’], [1, 2, 3]) = [ [‘a’, 1], [‘b’, 2], [‘c’, 3] ]
```

(b) Define the following function that does the opposite of mix **6 marks**

```
def unmix(listC)
# takes a list of lists produced by the above mix function and returns the two original
# lists e.g. unmix([ [‘a’, 1], [‘b’, 2], [‘c’, 3] ]) = ([‘a’, ‘b’, ‘c’], [1, 2, 3])
```

Q 6. Dictionaries and decision structures

(a) Define the following function **8 marks**

```
def marksDistribution(dict):
# takes a dictionary of the form dict = {'student-1 name': marks1, 'student-2 name:
# marks2, ……} where marks are integers and returns a dictionary of the form
# {'0-20': n1, '21-40': n2,… '81-100:n5} where n1 … n5 are the number of students
# in the respective ranges.
```

(b) What happens if a student's name appears twice in the input dictionary? **1 mark**

(c) Will the returned dictionary be ordered as shown in (a)? **1 mark**

Q 7. Recursion **10 marks**

Define a recursive function that performs quick sort on a list of integers. In quick sort a pivot point is chosen and the list elements less than the pivot are moved ahead (to the left) of it and those that are greater than the pivot are moved to the right. This process is then recursively repeated with the two sub-lists.

e.g. def quick_sort(lst, start, end):
# this function performs quick sort recursively i.e. using recursion.

Q 8. Difference-Reduction Method

(a) Describe the basic principles of the difference-reduction method of problem solving.
5 marks

(b) Illustrate your answer to (a) with a problem that is amenable to this technique, and sketch a solution to this problem that uses the technique. **5 marks**

Q 9. Divide and Conquer

(a) Describe the basic principles and efficient operation of the problem solving technique “divide-and-conquer”. **5 marks**

(b) Illustrate your answer to (a) with a problem that is amenable to this technique, and sketch a solution to this problem that uses the technique. **5 marks**

END OF PAPER