UNIVERSITY OF ESSEX

Undergraduate Examinations 2014

INTRODUCTION TO PROGRAMMING

Time allowed: **TWO** hours

Candidates are permitted to bring into the examination room:
Hand-held, non-programmable calculators (containing no textual information)

Candidates must answer **ALL** questions.

The paper consists of **SIX** questions.

The questions are **NOT** of equal weight.

The percentages shown in brackets provide an indication of the proportion of the total marks for the **PAPER** which will be allocated.

Please do not leave your seat unless you are given permission by an invigilator.
Do not communicate in any way with any other candidate in the examination room.
Do not open the question paper until told to do so.
All answers must be written in the answer book(s) provided.
All rough work must be written in the answer book(s) provided. A line should be drawn through any rough work to indicate to the examiner that it is not part of the work to be marked.
At the end of the examination, remain seated until your answer book(s) have been collected and you have been told you may leave.

*Candidates must answer **ALL** questions.*

Question 1

(a) Briefly describe how exceptions can be used in Python to handle errors. [7%]

(b) The following Python program should read a sequence of integers from the keyboard and terminate when the total of the numbers that have been read is greater than 500, outputting the average value so far after each input. However it contains two errors that will be detected by the IDLE syntax-checker, one more that will cause failure at runtime, and another one that will cause the program to behave incorrectly. For each of the four errors, state [8%]

- whether it will be detected by the syntax checker, cause failure at runtime or cause the program to behave incorrectly
- the number of the line of code where it occurs, and if it causes runtime failure, the number of the line at which the failure will occur
- how the error should be corrected

```
01   total = 0
02   count = 0
03   running = True
04   while running
05       number = input("Enter a number (0 to finish):")
06       total += number
07       count + = 1
08       print("Current average is", total/count)
09   if total > 500 : running = False
```

Question 2

(a) Provide a full definition for a Python function called `isRoman`, which takes a string as an argument and returns `True` if its argument is a single-character string with a value that is used in Roman numerals (i.e. `′I′`, `′V′`, `′X′`, `′L′`, `′C′` , `′D′` or `′M′`) and returns `False` otherwise. [4%]

(b) Write a function that takes a string as an argument and makes several calls to the `isRoman` function in order to calculate and return the number of Roman-numeral characters in the string. [4%]

(c) Write two separate versions of a function called `compare`, which takes as arguments two lists and a positive integer `n`, and returns `True` if the last `n` items in the first list are the same as the last `n` items in the second list (in the same order), and returns `False` otherwise. You may assume that both lists contain at least `n` items. [7%]

The first version should use a loop to compare the items individually; the second must not use a loop.

Question 3

(a) What is the value of each of the following Python 3 expressions? (If any of the expressions has no value since it would be rejected by the syntax checker you should give the answer "no value".)

(i) `9 / 3` [1%]

(ii) `9 // 2` [1%]

(iii) `4 * "hi"` [1%]

(iv) `[3, 4] + 2` [1%]

(v) `"hello"[:-2]` [1%]

(b) What is the value of `sum` after the following code has been executed? [3%]

```
i = 10
sum = 0

while i > 1 :
  if i%3 == 0 :
    sum = sum + i
  else:
    sum = sum + 1
  i = i - 1
```

(c) Briefly describe the syntax and behaviour of a `for` loop in Python. [3%]

(d) Rewrite the code fragment from part (b) as concisely as possible using a `for` loop. [4%]

Question 4

(a) Explain what is meant by a *recursive* function. [3%]

(b) The binomial coefficient of two positive numbers, *n* and *m*, can be calculated using the following definition: [7%]

> If *m*>*n* the binomial coefficient of *n* and *m* is 0.
> If *m* is equal to *n* the binomial coefficient of *n* and *m* is 1.
> If *m* is equal to 1 the binomial coefficient of *n* and *m* is equal to *n*.
> Otherwise the binomial coefficient of *n* and *m* is equal to the sum of the binomial coefficients of *n*-1 and *m*-1 and of *n*-1 and *m*.

Use this definition to produce a recursive function that takes as arguments two integers `m` and `n` and returns the binomial coefficient of `m` and `n`. You may assume that the arguments supplied to the function will always be positive.

(c) The following program runs without error, but does not perform any useful computation.

```
def myFunc(x) :
  c = x + a

a = 0
myFunc(1)
b = a + 1
myFunc(b)
a = b
```

The program contains three variables: `a`, `b`, and `c`.

(i) What is the scope of the variable `a`? [1%]

(ii) What is the scope of the variable `c`? [1%]

(iii) Which variables (if any) have a lifetime that lasts until the end of the execution of the program? [1%]

(iv) Briefly explain why, if an extra line `a = 5` were to be added immediately beneath the line `c = x + a` (at the same level of indentation), the program would fail at runtime. [2%]

Question 5

(a) An outline for a `Lecturer` class is defined as follows. [10%]

```
class Lecturer(object) :
    def __init__(self, name, dept, salary) :
        ...............

    def __str__(self) :
        ...............
```

Provide complete bodies for the two methods. The initialisation method should store in attribute variables within the class object the name (string), department (string) and salary (integer) supplied as arguments, and should raise an exception if the salary supplied as an argument is less than 0, The `__str__` method should return a string that contains details of the lecturer in the format

```
Name: Fred Smith; Department: CSEE; Salary: £30000
```

(b) Write a full definition for a Python function which takes as an argument a list of objects of type `Lecturer` and a salary value and displays the details of all of the lecturers in the list who have salaries of less than the value supplied as the second argument, one lecturer per line. [5%]

Question 6

A file containing data about a collection of people has the following format. [25%]

```
Fred Smith 23 programmer
Jim Brown 48 programmer
Ann Simpson 33 doctor
```

Each line contains a first name, a second name, an age and a job title.

Write a complete Python program that will input the contents of the file into a list of tuples, and output each of the following

- full details of the people sorted by age (in a neatly formatted table)
- the names of all people with an age of 30 or less
- the number of people with each job title

The output for the data shown above should be

```
Fred Smith          23  programmer
Ann Simpson         33  doctor
Jim Brown           48  programmer

Fred Smith

doctor 1
programmer 2
```

You may assume that no name occupies a total of more than 20 characters.

The filename should be obtained from the user; if the file does not exist the program should terminate after displaying an appropriate message.

END OF EXAM PAPER CE151-4-AU