University
of Victoria

University of Victoria
Final Examination
August 2014

Course Name & No.	SENG 265 Software Development Methods
Section(s)	A01
CRN	30704
Instructor	Michael Zastre
Duration	Three (3) hours

Name:

Student Number: V00________________

This exam has a total of 15 pages including this cover page.

Students must count the number of pages and report any discrepancy immediately to the Invigilator.

This exam is to be answered on the paper.

No materials or aids of any kind are permitted. This is a closed-book exam.

Ensure all cellphones are switched off. You must obtain permission from an invigilator to temporarily leave the examination room.

The total number of marks in this exam is 200.

Section A (40 marks): 20 questions

For each question in this section, **clearly circle ALL answers** that apply. All questions have equal weight. *Partial marks are not given for incomplete answers.*

Question 1: Arrays in C:

a. May contain a mix of types like lists do in Python.
b. Always contain within them a field storing the size of the array (i.e., number of elements).
c. Are sliced using the colon operator (e.g., `sum[1:9]`).
d. Can be declared, amongst other places, in file scope or program scope.
e. None of the above.

Question 2: A `for` -loop in C:

a. Can contain other loops within it as nested loops.
b. Is a bottom-tested loop just like the `do-while` statement.
c. Must not contain statements that dereference pointers.
d. May refer to two variables in its loop-condition expression.
e. None of the above.

Question 3: When using `malloc()`, `calloc()` or `realloc()`:

a. We are always guaranteed to obtain from the heap the amount of memory we request.
b. A wrapper function can be written that takes care of examining return values for `NULL`.
c. It is good practice to cast the result to some pointer type.
d. Code should call `free()` once memory obtained from these calls is no longer needed (i.e., in order to prevent memory leaks).
e. None of the above.

Question 4: Every C source-code file:

a. Must contain the "`main()`" function as otherwise the file cannot be part of a C program.
b. May `#include` a header file.
c. Must include function prototypes for all other functions in all other modules.
d. Can have variables that are only visible within the file and not outside of it.
e. None of the above.

Question 5: The C string function `strtok(somestring, someseparatorchars)`:

a. Must be followed at some point with `strtok(NULL, someseparatorchars)` in order to continue tokenizing the original string.
b. Can only be used when we `#include <strtok.h>`.
c. Automatically allocates memory for all tokens extracted from the string.
d. Returns the value -1 when there are no more tokens in the original string.
e. None of the above.

Question 6: Python variables:

a. May refer to functions.
b. Can be assigned values of different types within the same function.
c. Can never be global as global variables must be statically typed in all languages.
d. May be assigned the value of a tuple.
e. None of the above.

Question 7: When a description of the Python language refers to *immutable variables*:

a. It refers to the variables whose values may be frozen through the `immobilize` function.
b. The description explains how list comprehensions result in lists that can no longer be modified.
c. It is really dealing with variables that may not be garbage collected in Python.
d. It recommends when to use a `for`-statement and when to use a `while`-statement.
e. None of the above.

Question 8: Python code blocks:

a. Can be used to represent the true- and false-code of an `if-else` statement.
a. May contain other Python blocks.
b. Are indicated by indentation.
c. Are the only structures in which mutable tuples may be constructed.
d. None of the above.

Question 9: When using Python lists:

b. Slices of a list may be accessed with syntax such as `somelist[start:end]`.
c. A negative lookup such as `somelist[-3]` refers to list values positioned relative to the end of the list (i.e., here the third-to-last item).
d. A copy of list contents is made as the result of a statement such as `some_other_list = somelist`.
e. We can refer to the list from the second element to the end using `somelist[:2]`.
f. None of the above.

Question 10: To create a C-like switch statement in Python, we:

a. Should use the `case` statement.
b. Should use the `switch` statement
c. Should branch to a nested C script.
d. Can use `map` on a list.
e. None of the above.

Question 11. Considering the Python expression `(True and (x > 5) and bar)`:

a. It generates a syntax error as `bar` always represents a function and must be called with parentheses.
b. Its value may be `True` or `False`.
c. Its value may be `False` or the value of `bar`.
d. It may be used to control the iterations of a `while` loop.
e. None of the above.

Question 12: When writing the body of a function in Python:

a. You must always refer to the `self` variable.
b. The block of the function definition must be indented.
c. You must end the function with the `return` keyword.
d. Your code may call other Python functions.
e. None of the above.

Question 13: After we use `svn add` on a file in our project:

a. Every change to that file is automatically sent to the Subversion repository.
b. Every additional file added to that directory is automatically sent to the Subversion repository.
c. All other developers who have a copy of the project must perform the run `svn add` command.
d. A new file will be stored in the directory that is the concatenation of all changes made to the file.
e. None of the above.

Question 14: The UNIX "`rm`" command:

a. Can be used to delete text files.
b. Can be used to delete executable files.
c. Can be used to enlarge a text file (i.e., "more room").
d. Can be used to enlarge an executable file (i.e., "more room").
e. None of the above.

Question 15: Again considering `svn add`, after executing "`svn add categorise.c`" within an existing Subversion project, the user who issued this command:

a. Knows the file named "`categorise.c`" is part of the working copy tracked by Subversion.
b. Knows the file named "`categorise.c`" has been committed to the repository.
c. Has created a new file in their working module.
d. Has taken one of a series of steps that may allow another user to access the contents of "`categorise.c`".
e. None of the above.

Question 16: `svn checkout http://candykingdom.ooo/finnthehuman`

a. Can only be performed once for "`finnthehuman`".
b. Will return an error if some other user has already checked out "`finnthehuman`".
c. Creates possibly many subdirectories.
d. Is the opposite of "`svn checkin http://candykingdom.ooo/finnthehuman`".
e. None of the above.

Question 17: The `unittest` framework in Python:

a. can be used to associate test cases with class files we write in Python.
b. requires special scripting support from Unix to run the tests.
c. has methods for setting up and tearing down data structures needed for tests.
d. automatically executes the tests we have written.
e. none of the above.

Question 18: Regular expressions in Python:

a. May be used to identify and extract portions of strings.
b. Are only used to match strings and never for substitution into strings.
c. Make heavy use of meta-symbols.
d. Use the symbol `\d{+num}` to indicate the distance (here equal to `num`) in characters from the start of the searched string to an occurrence of the pattern itself.
e. None of the above.

Question 19: Dependencies in a "makefile" rule may:

a. Consist of a list of commands.
b. Consist of a list of files.
c. Consist of a list of targets.
d. Consist of a list of variables.
e. None of the above.

Question 20: In the Knuth-Barr categories of bugs:

a. The "algorithm" category means the code as written followed the intent of the programmer but the intent was wrong.
b. The "forgotten" category means the code reads or writes incorrect data or accesses a wrong storage location.
c. The "distracted" category means the code was written by overcommitted and exhausted developers who introduced bugs as a result of deadline pressures.
d. The "compiler" category means the bug was introduced into the source code by an incorrectly written compiler.
e. None of the above.

Section B: Python (Total marks: 60)

Question 21 (10 marks): Consider the following Python code, named "`mystery.py`":

```
#!/opt/local/bin/python   # Python 3 interpreter

import sys

def main():
    aaa = 0
    bbb = 0
    ccc = 0
    ddd = 0

    for eee in sys.stdin:
        ccc = ccc + 1
        ddd = ddd + 1
        aaa += len(eee)
        eee =  eee.rstrip()
        fff = eee.split()
        if (aaa % 5 == 0):
            print (ccc, bbb, end=" ")
            if fff:
                print (fff[-1])
        bbb += len(fff)

    print ("%04d %04d %04d" % (ccc, bbb, aaa))

if __name__ == "__main__":
    main()
```

and the following input, named "`in01.txt`":

```
There are the mud-flowers of dialect
And the immortelles of perfect pitch
And that moment when the bird sings very close
To the music of what happens.
-- Seamus Heaney (1938-2013)
```

Write on the lines below the output that results when the input is redirected into the Python program (i.e., "`cat in01.txt | ./mystery.py`"). Assume the file permissions on `mystery.py` are correct and the Python interpreter is located as shown in the bang path.

Question 22: (30 marks)

Write a Python function called `compute_hlav()` for analyzing a file containing stock-price or index-price data. The function will accept three parameters: the name of the file, a starting date in the form YYYYMMDD, and an ending date in the form YYYYMMDD. The function will return the highest price, lowest price, and average daily volume indicated by the data. Each line in the file is formatted as "`date open high low close volume adjusted_close`". For example, if the file "`GE.txt`" contains the following five lines:

```
20130312    8.73    9.78    8.43    9.57    355002000    9.44
20130313    9.9     9.98    9.3     9.62    216287500    9.49
20130311    9.23    9.36    8.31    8.49    251329800    8.37
20130316    9.97    10.36   9.6     9.66    226193300    9.52
20130310    8.01    8.99    7.95    8.87    368705700    8.75
```

and if the following line appears in a script using your function:

```
(high, low, average_volume) = compute_hlav("GE.txt", "20130311", "20130313")
```

then `high` equals 9.98, `low` equals 8.31, and `average_volume` equals 274206433. Assume all date values are legal.

Hints: Use "`int(somestring)`", "`float(somestring)`" to convert a string to an integer and float (respectively). Use "`str(someval)`" to covert a integer or floating point value to a string. Use "`input = open(filename, 'r')`" to open a file named "`filename`" and store the file pointer in "`input`". Make proper use of tuples and lists.

Your indentation must be clearly indicated. Some marks will be given for the quality of your answer.

(This next page is available, if needed, for your answer to Question 22.)

Question 23: (20 marks)

(a) What is the value of `list2` produced below given the following code? Show all work.

```
def functionA (m, n):
      return (m * 10) - n;

list1 = [(12, 10, 5), (5, 2, -22), (5, 5, 1)]

list2 = [functionA(r, s) for (r, t, s) in list1]
```

(b) Consider the following code for a "Window" class:

```
class Window:
     """A class representing a GUI window. """
     def __init__(self, n, w, h):
          self.window_title = n
          self.window_width = w
          self.window_height = h
     def get_dim(self):
          return (self.window_width, self.window_height)
```

Write the code for a subclass of "Window" called "Scrollable_Window" where there is are additional fields called "horiz_scroll" and "vert_scroll" meant to store boolean values (e.g., if "horiz_scroll" is true, that means there will be a horizontal scroll bar for the window, etc.). There will be additional methods named "is_horiz_scroll()" and "is_vert_scroll()" to return the new field values, plus an appropriate constructor for the subclass.

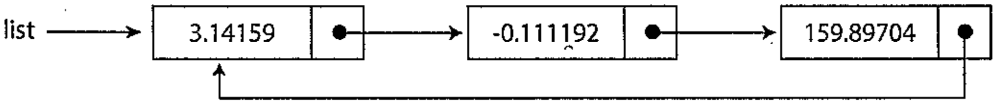
Section C: C programming (Total marks: 70)

Question 24: (30 marks)

Consider the following data structure:

```
typedef struct Node_t Node_t;
struct Node_t {
    float value;
    Node_t *next;
};
```

Also consider the following diagram showing one possible configuration of nodes using the data structure defined above:



Assuming the data-structure declaration in the box above is already given in of some C file, *write the remaining C code necessary in order to create dynamically* the structure shown in the diagram. Some marks will be given for the quality of your answer.

(The next page is available, if needed, for your answer to Question 24.)

(This page is available, if needed, for your answer to Question 24.)

Question 25: (40 marks)

Consider an array of random integers:

```
int[] arr = { 3, 14, 1, 59, 25, 53, 58, 9, 79, 323, 8,
     42, 2, 643, 38, 32, 7, 9, 50, 288, 0 };
```

The final "0" indicates that there are no more numbers to follow in the array. *This means that no value of 0 will ever be part of a calculation of the statistics itemized below.*

Write a C function `compute_stats` accepting such an array as a parameter and returning:

- the minimum value
- the maximum value
- the average value
- the number of elements in the array (excluding 0).

More precisely, these values must be returned in a dynamically-allocated structure and this structure is to be of your own design. You must provide all code for the function and for the structure declaration. Some marks will be given for the quality of your solution.

Section D: Regular Expressions (Total marks: 30)

Question 26: (30 marks)

Below is a table with one regular expression per row (with rows lettered):

A	`r'\b(\w\w?)\b'`
B	`r'^((\w\w?)\d*)+$'`
C	`r'[!$%#<>][a-z]\w*$'`
D	`r'[!$%#<>].?[a-z]\w*'`
E	`r'((\d{4})(\d{2})(\d{2}))|(\d{6})'`

Below is a table with a list of strings in each row written in Python:

`['x^2-1=0', 'x^2+5x+3=0', 'x^2-3x-24=4']`	
`['#include', '#define', '#ifdef', '#endif']`	
`['-40 F', '20C', '68.4F', '+10C', '1250 K']`	
`['H2O', 'H2SO4', 'AgBr', 'Na2S2O3', 'H8N2O3S2']`	
`['abc def 789', 'xyz yyzz rra', 314 1592 65v']`	
`['213-5222', '250-472-7000', '7172', '604-413-2198']`	
`['H', 'He', 'Li', 'Be', 'B', 'C', 'N', 'O', 'F', 'Ne']`	
`['<a>', '</a>', '<body>', '</body>', '<id div="menu">']`	
`['20131211T090000', '20131225T000000Z', '20131231', '129984Z']`	
`['pre-Christmas sales', 'after-dinner mints', 'hairy-legged droids']`	

For every row in the table above, ***write in the right-hand column the letter for each regular expression that matches all strings in the set.*** Some rows will have multiple regular expressions that match all strings in the set; some rows will have no matching regular expression, in which write "none". The letter "oh" is written as "O", while the number "zero" is written as "0".

END OF EXAM

This page is for the sole use of examination evaluators.

Section A	/40
Section B	/60
Section C	/70
Section D	/30
Total	**/200**