PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

St. George Campus

DECEMBER 2007 EXAMINATIONS

CSC 207H1F
Duration — 3 hours

Examination Aids: Any handwritten or printed materials. No electronic aids.

Student Number: |_|_|_|_|_|_|_|_|_|_|

Last (Family) Name(s): _____

First (Given) Name(s): _____

---

*Do **not** turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above, and read the instructions below *carefully*.)

This final examination consists of 7 questions on 16 pages (including this one). *When you receive the signal to start, please make sure that your copy of the examination is complete.*
There are two mostly-blank pages at the end of the exam that you may use if you run out of space.

Good Luck!

MARKING GUIDE

# 1:	_____/ 8
# 2:	_____/10
# 3:	_____/18
# 4:	_____/ 6
# 5:	_____/10
# 6:	_____/10
# 7:	_____/ 8
TOTAL:	_____/70

Question 1. [8 MARKS]

Part (a) [2 MARKS]

Write a command (or series of commands) that could be executed on the command line on a typical Unix system that would have the effect of copying any and all java source-code files (with names ending in `.java`) in the current directory to the parent directory of the current directory. Answer using as few commands as possible, and keep each command as short as possible.

Part (b) [4 MARKS]

Below are pairs of Unix commands with corresponding descriptions. If the description does not match the command, provide a brief description which does match in the space provided. Do **not** change the command — just the description. (If the description does match, just write "OK".)

- `pmpt> java MyClass | output.txt`
  Send the output of MyClass to a file named output.txt

- `pmpt> cat < file.txt`
  Print the contents of `file.txt` to standard out.

- `pmpt> grep hello > file.txt < extract.txt`
  Extract lines containing "hello" from file.txt, and store them in file extract.txt.

- `pmpt> java MyClass < file.txt | java MyClass`
  Send `file.txt` to standard input of MyClass, and send any resulting output to another instance of MyClass.

Part (c) [2 MARKS]

Consider the following segment of a Unix session:

```
dhcp0-126:~ $ cat test.py
#!/usr/bin/python
for i in range(0, 3):
        print "Hello World!"

dhcp0-126:~ $ ls -l test.py
-rw-r--r--   1 robertsw  robertsw  63 Nov 12 10:34 test.py
dhcp0-126:~ $
```

Write the Unix command(s) necessary to invoke `test.py` without calling the Python interpreter directly. (In other words, do not use "`python test.py`".) Use as few commands as possible. Briefly explain your reasoning.

Commands:

Reasoning:

Question 2. [10 MARKS]

Consider a file that contains definitions of words in the following form: The columns, in order from left to right, contain information about the word being defined, its part of speech, the year of earliest known usage, and a short definition. The columns are separated by exactly one tab character. Any value in any column can be "`null`". Otherwise, the word is a series of letters and can include spaces, the part-of-speech is one of {noun, verb, adverb, adjective}, the year is an integer between 1500 and 2007 inclusive, and the definition is a series of letters, punctuation, and spaces. Punctuation can include any of `,:;-'.?()`

Here are two example lines from the file:

```
erstwhile     adjective  1569   in the past
big wheel     noun       1942   bigwig, big shot
```

Write a Python program that expects to receive on the standard input the text of a file similar to that described above. Use a regular expression to match lines from the file to ensure that they conform to the description. If any lines of the file do not match the description, print them to the standard error.

Additional space for question 2

Question 3. [18 MARKS]

Write a Python function `normalize` that has a single string parameter representing a Canadian postal code and returns that same postal code in a standard form.

Canadian postal codes have the format letter,digit,letter,digit,letter,digit. For example they may be entered as `m5c2L7`. They are normally expressed with upper-case letters and written with a space after the third character: `M5C 2L7`.

In the input to your function, the postal code may have some lower-case letters and the space is optional. However the function should "normalize" the postal code to use only uppercase letters and include the space.

If the input to `normalize` is not a string, the function should raise a `TypeError`. If the format of the input string is a string but is incorrect in any other way, `normalize` should raise a `ValueError`.

Part (a) [5 MARKS]

Write your `normalize` function here.

Part (b) [3 MARKS]

Below is the beginning of a unittest suite to test normalize. Add one more test which tests the behaviour of the function, when the input is a string that is not a valid postal code because it has too many spaces.

```
import unittest
from normalize import normalize

class normalizeTest(unittest.TestCase):

  def testExample(self):
      self.assertEqual(normalize('l5r3J4'),'L5R 3J4')




if __name__ == '__main__':
    unittest.main()
```

Part (c) [10 MARKS]

Write a Python program that read lines from an input file where the lines contain zero, one or more postal codes. The postal codes are not necessarily normalized. Use the function from part (a) to normalize the codes and then count them and print them out in order of decreasing frequency. So, for example, `m2n3b4` and `M2N 3B4` should both be counted as `M2N 3B4`.

Your program will obtain the input file name from the only command line argument. If the file given in the command line argument does not exist, print a message to standard error .

Question 4. [6 MARKS]

A co-worker who is collaborating with you on a project has set up a Subversion repository for the two of you and has already added some files. You are using a computer on which SVN has already been properly set up.

Part (a) [1 MARK]

Assuming that your co-worker set the repository up at `http://big.company.com/svn/trunk`, write the SVN command(s) necessary to make a local copy of repository files on your machine.

Part (b) [1 MARK]

You have two files called `todo.txt` and `notes.txt` that need to be added to the version control system. Write the SVN command(s) necessary to get both of those files into the repository that was set up by your co-worker.

Part (c) [4 MARKS]

Assume that the repository now contains the following files: `file1.py`, `file2.java`, `todo.txt`, and `notes.txt`. Assume also that you (User A) and your co-worker (User B) have up-to-date repositories. Assuming that the commands in the table are executed in order, fill in the table with YES or NO as appropriate, depending on whether the file `notes.txt` exists on the specified user's computer. Assume that no other commands are being executed except those indicated.

If you make any assumptions, please state them below the table.

User	Command	Local copy exists on A's computer after command is executed?	Local copy exists on B's computer after command is executed?	Copy exists in repository after command is executed?
A	svn remove notes.txt			
B	svn update			
A	svn commit			
B	svn update			

Question 5. [10 MARKS]

Here is part of an XML document representing the inventory for a certain retailer.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="item_list.xsl" alternate=no ?>
<doc>
    <item name="fish sticks">
        <price>7.99</price>
        <units>kg</units>
        <stock>75</stock>
    </item>
    <item name="light sweet crude oil">
        <price>94.62</price>
        <units>barrel</price>
        <stock>793237</stock>
    </item>
    .
    .
    .
</doc>
```

Here is an XML file representing an order from a customer for the purchase of items from the retailer.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="item_list.xsl" alternate=no ?>
<doc>
    <item name="fish sticks">
        <quantity>10</quantity>
        <units>kg</units>
    </item>
    <item name="mouse pad">
        <quantity>13</quantity>
        <units>box</units>
    </item>
</doc>
```

Assume that the samples above demonstrate all of the tags and attribute values contained within the XML files.

Write a function using Python's DOM (Document Object Model) facility that takes two parameters as input. The first parameter is the name of an XML file representing items for sale. The second parameter is the name of an XML file representing a customer's order. Your function should return the total value of the customer's order, subject to the following rules: you must verify that units specified for an item in the purchase order match the units specified in the stock information. Second, you must verify that the retailer has enough stock to cover the quantity of each item ordered. For each item, if the units match and the stock is large enough to cover the quantity ordered, add this item's total price to the price of the order. Otherwise, leave this item out of the final calculation. In other words, you will compute the price for as much of the order as the retailer can fill, even if that is less than the full order.

You may assume that there are no errors in either file, and that all numerical values are integers.

Additional space for question 5

Question 6. [10 MARKS]

Here is a Python module in a file called "mod.py":

```
val = 3.4
erie = 'boo'

def f(x, y):
    "f(x, y) prints a greeting and then x and y."
    print 'Hi,', x, y

def g(y):
    '''g(y) returns the square of y.'''
    return y*y*y


class C(object):
    '''A helpful class.
    '''

    def meth(self, p):
        '''Compare this C with p, assumed to be another instance of C.'''
        if self.f < p.f:
            return -1
        elif self.f > p.f:
            return +1
        else:
            return 0

    def __init__(self, val):
        self.f = val
```

Assume that you are writing a program in a Python source file that has already imported `mod`. For each of the parts below, supply the required code.

Part (a) [2 MARKS]

Supply a function 'q' that does exactly what `mod.f` does, and give an example call of `q`.

Part (b) [2 MARKS]

Create an instance of `mod.C`; the instance should be called `ci`. Cause `mod.C` to have a method `mod.C.r` that uses the code of `mod.f`.

Part (c) [2 MARKS]

Assuming that your code from part (b) works, give an example call of `r` as an instance method of `ci`. What is the output from this call?

Part (d) [2 MARKS]

Again assuming you have an instance of `C` called `ci`, show how to add a new instance variable `vbl` to `ci`. Does every object of the class `mod.C` now have an instance variable `vbl`?

Part (e) [2 MARKS]

For an object `obj` belonging to any class `Cl`, how can you find out what instance variables `obj` has?

Question 7. [8 MARKS]

```
FBC = foobar -9

all: a1 a2

a1: a1.foo

a2: a2.foo

a3: a3.bar
    ${FBC} a3.bar > a3

%.foo: %.bar
   cat $< > $@

clean:
   rm *.bar *~
```

The text above is the contents of a file called `project.mk`. Refer to it when answering the following questions. If you make any assumptions, please state them.

Part (a) [1 MARK]

How many rules does the file contain?

Part (b) [1 MARK]

How many prerequisites are explicitly mentioned?

Part (c) [2 MARKS]

List the targets that will probably never be considered up-to-date. Briefly explain your reasoning and in particular explain why there is a small probability that they might be up-to-date.

Part (d) [3 MARKS]

Make operates by automatically executing commands that could, if necessary, be executed by manually typing them on the command line. In a directory that contains only `a1.bar`, `a2.bar` and `project.mk`, what command line instructions could be executed that would be identical to those executed by running `make -f project.mk`?

Part (e) [1 MARK]

In a directory that contains only `a1.bar`, `a2.bar` and `project.mk`, what command line instructions could be executed that would be identical to those executed by running `make -f project.mk a3`?

[This is a "blank" page. Use the space below for rough work. This page will **not** *be marked, unless you clearly indicate the part of your work that you want us to mark.]*

[This is a "blank" page. Use the space below for rough work. This page will **not** *be marked, unless you clearly indicate the part of your work that you want us to mark.]*

Total Marks = 72