CSC 108H1 F 2010 Test 2
Duration — 45 minutes
Aids allowed: none

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

Last Name: _____ **First Name:** _____

Lecture Sections: L0101 and L0102 **Instructors: Horton and Engels**

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm consists of 4 questions on 8 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments and docstrings are not required except where indicated, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code. No error checking is required: assume all user input and all argument values are valid.
If you use any space for rough work, indicate clearly what you want marked.

1: _____/ 4
2: _____/ 6
3: _____/ 6
4: _____/ 8

TOTAL: _____/24

Question 1. [4 MARKS]

Beside each code fragment below, show the output that it would create. If it would generate an error say so, and give the reason why.

Part (a) [1 MARK]

```
L = ["this", "is", "fun"]
for x in L:
    x = x + "!"
print L
```

Part (b) [1 MARK]

```
s = "hellllo"
d = {}
for i in range(len(s)):
    d[s[i]] = i
print d
```

Part (c) [1 MARK]

```
L = [[10, 12, 14], [1, 2, 3, 4, 5], ["a", "b", "c"]]
print L[1][3]
```

Part (d) [1 MARK]

```
s = "what!sup?"
k = s.index("!")
print s[1:k-1] + s[k+1:]
```

Question 2. [6 MARKS]

Write the function below, according to its docstring. You must not use a for-loop in this question or your solution will earn zero.

```
def first_neg(L):
    '''L is a list of ints.  Return the index of the first element of L that is
    negative.  If none are negative, return -1.'''
```

Question 3. [6 MARKS]

Suppose we have two dictionaries whose values are `int`s. Define the **dictionary maximum** of the two dictionaries to be a new dictionary containing every key that is in both of the dictionaries. The value associated with a key is the maximum of the values for that key from d1 and d2. For example, if we have these two dictionaries:

```
d1 = {"a": 5, "d": 11, "c": -2, "j": 99}
d2 = {"d": 4, "j": 101, "z": 8}
```

their dictionary maximum is `{'d': 11, 'j': 101}`.

Write the function below, according to its docstring.

```
def dict_max(d1, d2):
    '''d1 and d2 are dicts whose values are ints.  Return a new dict that
    is the dictionary maximum of d1 and d2.'''
```

Question 4. [8 MARKS]

Write the function below, according to its docstring.

```
def big_deposits(filename):
    '''str filename is the name of a file that stores deposits into a bank
    account. Each deposit is stored in a single line as an amount
    preceded by a dollar sign (for example: $1254.95). Return the number of
    deposits that exceed $1000.'''
```

[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.]

Short Python function/method descriptions:

```
__builtins__:
  len(x) -> integer
    Return the length of the list or string x.
  max(L) -> value
    Return the largest value in L.
  open(name[, mode]) -> file object
    Open a file.
  range([start], stop, [step]) -> list of integers
    Return a list containing the integers starting with stop and ending witt  stop - 1 with step
    specifying the amount to increment (or decrement).  If start is not specified, the list starts
    at 0.  If step is not specified, the values are incremented by 1.
dict:
  D[k] --> value
    Return the value associated with the key k in D.
  k in d --> boolean
    Return True if k is a key in D and False otherwise.
  D.keys() --> list of keys
    Return the keys of D.
  D.values() --> list of values
    Return the values associated with the keys of D.
  D.items() -> list of 2-tuples.
    Return a list of D's (key, value) pairs.
file (also called a "reader"):
  F.close(): Close the file.
  F.read([size]) -> read at most size bytes, returned as a string.
    If the size argument is negative or omitted, read until EOF is reached.
  F.readline([size]) -> next line from the file, as a string. Retain newline.
    A non-negative size argument limits the maximum number of bytes to return (an incomplete
    line may then be returned).  Return an empty string at EOF.
float:
  float(x) -> float
    Convert a string or number to a float, if possible.
list:
  x in L --> boolean
    Return True if x is in L and False otherwise.
  L.append(x): Append x to the end of the list L.
  L.index(value) -> integer
    Return the lowest index of value in L.
  L.insert(index, x): Insert x at position index.
  L.sort(): Sorts the list in ascending order.
int:
  int(x) -> integer
    Convert a string or number to an integer, if possible.  A floating point argument
    will be truncated towards zero.
```

Continued on reverse

Last Name: _____ **First Name:** _____

```
str:
  S.find(sub[,i]) -> integer
    Return the lowest index in S (starting at S[i], if i is given) where the
    string sub is found or -1 if sub does not occur in S.
  S.index(sub [,start [,end]]) -> int
    Like S.find() but raise ValueError when the substring is not found.
  S.lower() -> string
    Return a copy of the string S converted to lowercase.
  S.lstrip([chars]) -> string
    Return a copy of the string S with leading whitespace removed.
    If chars is given and not None, remove characters in chars instead.
  S.replace(old, new) --> string
    Return a copy of string S with all occurrences of the string old replaced with the string new.
  S.rstrip([chars]) -> string
    Return a copy of the string S with trailing whitespace removed.
    If chars is given and not None, remove characters in chars instead.
  S.split([sep]) --> list of strings
    Return a list of the words in S, using string sep as the separator and
    any whitespace string if sep is not specified.
  S.startswith(prefix) -> bool
    Return True if S starts with the specified prefix and False otherwise.
  S.strip() --> string
    Return a copy of S with leading and trailing whitespace removed.
  S.upper() -> string
    Return a copy of the string S converted to uppercase.
```