University of Washington
CSE 140 Introduction to Data Programming
Winter 2013

Midterm exam
February 6, 2013

Name: ________________________________

UW Net ID (username): ________________________________

This exam is closed book, closed notes. You have 50 minutes to complete it. It contains 18 questions and 14 pages (including this one), totaling 90 points. Before you start, please check your copy to make sure it is complete. Turn in all pages, together, when you are finished. Write your initials on the top of ALL pages (in case a page gets separated during test-taking or grading).

Please write neatly: we cannot give credit for what we cannot read.

Good luck!

<table>
<thead>
<tr>
<th>Page</th>
<th>Max</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>90</td>
<td></td>
</tr>
</tbody>
</table>
1 Short answer

1. An immutable data structure is one that cannot change after being created. Give three reasons to use immutable data. Give reasons that are as different from one another as possible. Use no more than one sentence each.

(a) ____________________________________________________

(b) ____________________________________________________

(c) ____________________________________________________
2 Execute Python expressions

Execute each of the following expressions.

- If it executes without an error, then:
  
  **value** state the value that it evaluates to

- If it suffers an error during evaluation:

  **error** describe the error (in one phrase — a brief explanation in your own words)
  
  **frame** state the name of the current environment frame: “global” or a function name
  
  **operator** state the operator or operation that caused the error, such as + or function invocation
  
  **arguments** state the values to which the operator was being applied

Your answer will contain either part “value”, or parts “error”, “frame”, “operator”, and “arguments”.

  Hint: none of these is a syntax error.

2. `len(set(1, 2, 3))`

   **value** ______________

   **error** ______________

   **frame** ______________

   **operator** ______________

   **arguments** ______________

3. `len(set([1, 2, 3]))`

   **value** ______________

   **error** ______________

   **frame** ______________

   **operator** ______________

   **arguments** ______________
4. `len(set([(1, 2), (3, 4)]))`

    value ______________________
    error ______________________
    frame _______________________  
    operator _____________________
    arguments ____________________

5. `len(set([[1, 2], [3, 4]]))`

    value ______________________
    error ______________________
    frame _______________________  
    operator _____________________
    arguments ____________________

6. `len(set(((1, 2), (3, 4))))`

    value ______________________
    error ______________________
    frame _______________________  
    operator _____________________
    arguments ____________________
The following questions assume these definitions:

```python
def double(x):
    return x + x

def at42(f):    # The name "at42" is short for "apply_to_42"
    return f(42)
```

7. `at42(double)`

| value | _____________ |
| error | _____________ |
| frame | _____________ |
| operator | _____________ |
| arguments | _____________ |

8. `at42(double(42))`

| value | _____________ |
| error | _____________ |
| frame | _____________ |
| operator | _____________ |
| arguments | _____________ |
9. `double(at42(42))`

value ___________________
error ___________________
frame ___________________
operator ___________________
arguments ___________________

10. `double(double)(42)`

value ___________________
error ___________________
frame ___________________
operator ___________________
arguments ___________________
3 \textbf{Draw the environment}

11. Draw the entire environment, including all active environment frames and all user-defined variables, at the moment that the division operation is performed.

\begin{verbatim}
def sum(input):
    result = 0
    for i in input:
        result = result + i
    return result

def length(input):
    return len(input)

def mean(input):
    return float(sum(input)) / length(input)

input = [5, 2, 2, 3]
print sum(input), length(input), mean(input)
\end{verbatim}
4 Debugging

You have written the following program (in file `printdicts.py`), and you have successfully loaded it into the Python interpreter (say, by pressing F5 within IDLE while viewing `printdicts.py`, or alternately by typing `from printdicts import *`).

```python
print_many_dicts(dicts):
    """Print each dictionary in the input, which is a list of dictionaries."""
    for dict in dicts:
        print_dict(dict)

def print_dict(dict):
    """Print the given dictionary."""
    print '{
    for k in dict:
        print ' ' + str(k) + ': ' + str(dict[k]) + ',
    print '}'
```

Here is an example of using it in the interpreter:

```python
>>> print_many_dicts( [ {1:2, 3:4}, {"abra":"cadabra", "hello":"world"} ] )
{
    1: 2,
    3: 4,
}
{
    hello: world,
    abra: cadabra,
}
```

12. You type an expression to the interpreter, and you get the following output:

```python
>>> ... you typed your expression here ...
{
 Traceback (most recent call last):
   File "<stdin>", line 1, in <module>
   File "printdicts.py", line 4, in print_many_dicts
     print_dict(dict)
   File "printdicts.py", line 10, in print_dict
     print ' ' + str(k) + ': ' + str(dict[k]) + ','
TypeError: list indices must be integers, not str
```

Write the smallest expression that you can that produces this exact output.
13. After typing a different expression to the interpreter, you get the following output. The only difference from the previous problem is the last line of output.

```python
>>> ... you typed your expression here ...
{
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "printdicts.py", line 4, in print_many_dicts
    print_dict(dict)
  File "printdicts.py", line 10, in print_dict
    print ' ' + str(k) + ': ' + str(dict[k]) + ','
IndexError: list index out of range
```

Write the smallest expression that you can that produces this exact output.
14. In no more than one sentence, explain the following Python error and how it could arise:

   TypeError: unhashable type: 'list'

   This error occurs when trying to use a list as a dictionary key, as lists are mutable and cannot be used as dictionary keys.
5 Write Python code

15. Write the body of the max_even function.

```python
def max_even(lst):
    """Return the largest number in the input that is an even number.
    The input must be a list of integers.
    If there is no even number in the input, return None.
    Examples:
    max_even([3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 9]) => 6
    max_even([3, -1001]) => None
    """
```

16. Write the body of a function that prints each word in a given file, on its own line. For example, if you were to execute `print_words("gettysburg.txt")` where file `gettysburg.txt` is as follows:

```
Four score and seven years
ago, our
```

then the output would be

```
Four score and seven years ago, our
```

We have given you the first line of the solution.

```python
def print_words(filename):
    """Print each word of the given file, each on its own line."""
    file = open(filename, "r")
```
In the next two problems, suppose that you are given the following data structure.

```python
home_runs = [
    ("Babe Ruth", 47, 1926),
    ("David Ortiz", 54, 2006),
    ("Albert Pujols", 47, 2009),
    ("Alex Rodriguez", 54, 2007),
    ("Babe Ruth", 49, 1930),
    ("Alex Rodriguez", 47, 2003),
    ("Babe Ruth", 54, 1920),
    ("Sammy Sosa", 49, 2002)
]
```

17. Write code that will print this list, sorted alphabetically by first name and (when the name is the same) by date. In particular, your code should print this data structure (though the formatting, such as line breaks, might be different):

```python
[('Albert Pujols', 47, 2009),
 ('Alex Rodriguez', 47, 2003),
 ('Alex Rodriguez', 54, 2007),
 ('Babe Ruth', 54, 1920),
 ('Babe Ruth', 47, 1926),
 ('Babe Ruth', 49, 1930),
 ('David Ortiz', 54, 2006),
 ('Sammy Sosa', 49, 2002)]
```

Don’t define a function — just write statements that achieve the goal.
Don’t do anything special for formatting — just use Python’s `print` statement, applying it to a list of tuples.
18. Write code that will print this list, sorted in decreasing order by number of home runs and (when the number of home runs is the same) by increasing date. In particular, your code should print this data structure (though the formatting, such as line breaks, might be different):

```python
[('Babe Ruth', 54, 1920),
 ('David Ortiz', 54, 2006),
 ('Alex Rodriguez', 54, 2007),
 ('Babe Ruth', 49, 1930),
 ('Sammy Sosa', 49, 2002),
 ('Babe Ruth', 47, 1926),
 ('Alex Rodriguez', 47, 2003),
 ('Albert Pujols', 47, 2009)]
```

Don't define a function — just write statements that achieve the goal. Don't do anything special for formatting — just use Python's `print` statement, applying it to a list of tuples.