University of Toronto

Duration: 25 minutes
Examiner: Kuei (Jack) Sun

Please fill your student number, last and first name below and then read the instructions carefully.

Student Number: ____ ____ ____ ____ ____ ____ ____ ____ ____ ____

Last Name: _____

First Name: _____

Instructions

Examination Aids: Ruler and examiner approved aid sheet are allowed.

Do not turn this page until you have received the signal to start.

You may remove the aid sheet from the back of this test book. Do not remove any other sheets from this test book. Answer all questions in the space provided. No additional sheets are permitted. Use the blank space in last page as scratch space. Its content will not be marked.

This exam consists of 5 questions on 7 pages (including this page). The value of each part of each question is indicated. The total value of all questions is 25 marks.

For the written answers, explain your reasoning clearly. Be as brief and specific as possible. Clear, concise answers will be given higher marks than vague, wordy answers. Marks will be deducted for incorrect statements in an answer. Please write legibly!

Work independently.

MARKING GUIDE

Q1:	_____	(2)
Q2:	_____	(2)
Q3:	_____	(5)
Q4:	_____	(6)
Q5:	_____	(10)
Total:	_____	(25)

Question 1. What's the Output? [2 marks]

```
a = b = [1, 2]
a.append(3)
print(b)
```

a) What is the output of the code shown above? [1 mark]

`[1, 2, 3]`

b) Describe the relationship between the variable *a* and *b*. [1 mark]

b is an alias of *a*, and vice versa.

Question 2. Dictionary Key [2 marks]

Given the variable *x*, defined here:

```
x = "hello", 5, (3.14, 1.41)
```

a) Can *x* be used as a dictionary key? [1 mark]

Yes.

b) Why or why not? [1 mark]

x is hashable because it is a tuple (immutable) that contains no mutable objects.

Question 3. **Scope** [5 marks]

For the following code written in C:

```
int x = 0;

void f() {
    printf("%d\n", x);
}

void h() {
    x += 2;
    f();
}

void g() {
    int x = 5;
    h();
}
```

```
int main() {
    g();
    if (x > 1) g();
    else {
        int x = 1;
        h();
    }
    h();
    return 0;
}
```

Assume block scope is used for the two questions below.

a) What is the output if the language uses lexical scope? [2 marks]

2
4
6

b) What is the output if the language uses dynamic scope? [3 marks]

7
3
2

Question 4. **Import Produce** [6 marks]

Given the following Python script:

```
NAME = 0
PRICE = 1

from produce import *

def print_menu(items):
    for item in items:
        print("%s: %.2f"%(item[NAME], item[PRICE]))

print_menu([("Pizza", 7.99), ("Burrito", 2.49), APPLE, BANANA])
```

a) What is the content of produce.py that will allow running the above script to generate the following output? [2 marks]

```
Pizza: 7.99
Burrito: 2.49
Apple: 1.29
Banana: 0.99
```

produce.py

```
APPLE = ("Apple", 1.29)
BANANA = ("Banana", 0.99)
```

Suppose someone added one line to produce.py, and you now get the following error instead:

```
Traceback (most recent call last):
  File "menu.py", line 10, in <module>
    print_menu([("Pizza", 7.99), ("Burrito", 2.49), APPLE, BANANA])
  File "menu.py", line 8, in print_menu
    print("%s: %.2f"%(item[NAME], item[PRICE]))
IndexError: tuple index out of range
```

b) Give one possibility of the line that was added to produce.py. [2 marks]

```
NAME = 3
```

or

```
PRICE = 3
```

Any integer value outside of the range [-2, 1] is accepted.

-1 mark if NAME or PRICE is set to non-integer values (because they would not cause IndexError)

c) Without removing the line that was added in part b., add another line to the top of produce.py so that the problem would go away and that output in part a. is shown again. [2 marks]

```
__all__ = [ 'APPLE', 'BANANA' ]
```

-1 mark if quotation is missing around APPLE and/or BANANA (they must be strings).

Question 5. **Gradebook** [10 marks]

Given a file named grades.csv, that is formatted as a comma separated file, like this:

```
1002375892, A+
1008192357, B
1015785975, B-
0995278912, C
1001367140, A
```

Write a Python function, `process_grades()`, that will process the file to return a dictionary that maps student numbers to their grades, e.g.:

```
{ 1002375892 : "A+", 1008192357 : "B", 1015785975 : "B-",
   995278912 : "C",  1001367140 : "A" }
```

Note that you should convert the student numbers to integers. If there is an error opening the file, print the error and return an empty dictionary. You may assume there are no duplicate entries in this file.

```
def process_grades():
    try:
        f = open('grades.csv', 'rt')
    except OSError as err:
        print(err)
        return {}

    out = dict()
    for line in f:
        token = line.split(",")
        # assume file is well-formatted
        out[int(token[0])] = token[1].strip()

    f.close()
    return out
```

[*Use the space below for rough work*]

END OF EXAMINATION