

Duration: 60 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.

MARKING GUIDE

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

Q1: _____ (3)

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.

Q2: _____ (5)

Q3: _____ (3)

This exam consists of 6 questions and 1 bonus question on 9 pages (including this page). The value of each part of each question is indicated. The total value of all questions is 40 marks.

Q4: _____ (6)

Q5: _____ (8)

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!

Q6: _____ (14)

Bonus: _____ (1)

Work independently.

Total: _____ (40)

Question 1. Mystery String [3 marks]

Given a mystery string of length 11, we created 3 different slices and counted the number of the letter *u* in the mystery string, as shown here:

```
>> mystery[1::3]
'ruum'
>> mystery[-1::-2]
'mlcuec'
>> mystery[3:6]
'pus'
>> mystery.count('u')
3
```

What is the mystery string?

```
'crepusculum'
1 mark for placing 'ruum' at the correct locations
1 mark for placing 'mlcuec' in the correct locations
3 marks for correct word
```

Question 2. Binary Format String [5 marks]

What is the format string, `fmt`, that will generate the following output?

```
>> result = unpack(fmt, b'\x00\x00\x00\x01john\x00\x00\x00\x00' \
.. b'\x00\x00\x00\x02\x00\x00\x00\xff')
>> print(result)
(1, b'john', 2, 255)
```

```
'!i4sqi' or '>i4sqi' or creative use of padding bytes.
E.g. '>xxxb4sqi'
```

1 mark per symbol

Question 3. Dynamic Dispatch [3 marks]

If we assume that Python uses dynamic dispatch to implement runtime polymorphism, what would be the output of this program? Justify your answer for full marks.

```
class A:
    x = 2
    def magic(self):
        return self.x * self.y

class B(A):
    y = 9
    def __init__(self):
        self.x = 7

class C(B):
    def __init__(self, y):
        self.y = y

t = C(3)

def add(self):
    return self.x + self.y
B.magic = add
A.x = 5

print(t.magic())
```

15 [1 mark]

If Python uses dynamic dispatch, then once the instance *t* is constructed, it cannot change behaviour. Therefore, adding `B.magic` should not change which method `t.magic` refers to, i.e., `t.magic` should invoke `A.magic`. However, updating `A.x` to 5 is still allowed (it does not change behaviour, only state). In this case, `self.y` is 3 (instance variable) and `self.x` refers to `A.x` and was updated to 5, therefore the output is 15. [2 marks]

Question 4. Operator Overloading [6 marks]

Given the follow expected output:

```
>> p1 = Point(2,3)
>> p2 = Point(3,4)
>> print(p1 + p2)
(5, 7)
```

Complete the class definition for the Point class below by adding more method(s), such that the above output will be printed.

```
class Point:
    def __init__(self, x=0, y=0):
        self.x, self.y = x, y

    # complete me here

    # 3 marks
    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)

    # 3 marks
    # any other string formatting solution is accepted
    def __str__(self):
        return "({}, {})".format(self.x, self.y)
```

Question 5. First Class Citizen [8 marks]

- a) Apply the three criteria of first class citizenship to *classes* in C++. Show at least one example per criterion, and state whether each criterion is satisfied or not. [6 marks]

1. Can be used. [1 mark]

Example regarding the use of class is accepted for 1 mark. E.g.

- Can be assigned to a variable? No.

```
typename t = Foo; // not allowed
```

- Can be passed to or return from a function? No.

```
int foo(typename cls); // not allowed
```

2. Can be constructed. [1 mark]

Two answers are accepted for 1 mark.

- Yes, you can create classes in local scope (i.e., inside a function)
- No, you cannot create new classes at runtime.

3. Have a type. [1 mark]

No, class itself does not have a type. [1 mark]

- b) Based on the analysis above, are classes in C++ first class citizens? Why or why not? [2 marks]

No. [1 mark]

Class in C++ does not pass all three criteria to be a first class citizen. [1 mark]

Question 6. Managed Attributes [14 marks]

- a) Complete the Bound descriptor class such that the value set to the descriptor must be within the range $[min, max)$. Raise `ValueError` if the assigned value is out of bound. Note that you do not need to implement the `__delete__` method. Hint: Look at the next question first, and see how it is used. Pay close attention to why an underscore is added to the name parameter in the `__init__` method of the Bound class. [7 marks]

```
class Bound:
    def __init__(self, name, min, max):
        self.name = "_" + name
        self.min = min
        self.max = max

    # complete me here

    # 3 marks
    def __get__(self, inst, owner):
        return getattr(inst, self.name)

    # 4 marks
    def __set__(self, inst, value):
        if value < self.min or value >= self.max:
            raise ValueError
        setattr(inst, self.name, value)
```

b) Complete the `set_time` and `get_time` methods of `Time` class shown below. [7 marks]

```
class Time:
    def __init__(self, value="00:00:00"):
        self.time = value

    # complete me here

    # 2 marks
    def get_time(self):
        return "%02d:%02d:%02d"%(self.hour, self.minute,
                                   self.second)

    # 5 marks
    def set_time(self, value):
        token = value.split(":")
        if len(token) != 3:
            raise ValueError
        self.hour = int(token[0])
        self.minute = int(token[1])
        self.second = int(token[2])

hour = Bound('hour', 0, 24)
minute = Bound('minute', 0, 60)
second = Bound('second', 0, 60)
time = property(get_time, set_time)
```

[IMPORTANT! MORE INSTRUCTION ON THE NEXT PAGE]

[Continued from previous page]

Time class keeps track of the time in 24-hour format. Your solution must break down the time string into three attributes, *hour*, *minute*, and *second*, like this:

```
>> t = Time("07:02:30")
>> t.hour, t.minute, t.second
(7, 2, 30)
>> t.time
'07:02:30'
>> t.time = "hello world"
ValueError
```

The time string returned from `get_time` must be in the format `hh:mm:ss`.

Your solution for `set_time` must raise `ValueError` upon any one of the two conditions:

1. If the format of the time is not in `hh:mm:ss` format (`5:5:10` is OK)
2. If any one of hour, minute, or second is out of range. (e.g. `25:03:01` is out of range)

Hint: Recall that `int()` will raise a `ValueError` if a string cannot be converted to integer (e.g. `int("hi")` will raise a `ValueError`)

Question 7. Bonus Mark [1 mark]

Would you like to see any changes to the way the ECE326 course is currently delivered? Let us know if you have any suggestions regarding the labs, the lectures, tutorials, etc. Any answer gets you a point, so don't leave a blank.

[Use the space below for rough work]

END OF EXAMINATION