PIC 16A
Python with Applications

Midterm

Instructions:

- You have **50 minutes** to complete the exam.
- There are **6** problems (1a, 1b, 2a, 2b, 2c, 3, 4a, 4b, 5a, 5b, 6) worth a total of 44 points.
- You should aim to spend **less than 4 minutes per output question**.
- You may **not** use any books or notes.
- Write your solutions in the space **below** or **next to** the questions.
- If you need space for thinking, ask for **scratch paper**.
- If you write important work on the scratch paper, **indicate** that you have done so **next to the relevant question**, and make sure to **hand in** your scratch work
- Do not forget to write your name and UID in the space below.

Name: _____
Student ID number: _____

Question	Points	Score
1	5	
2	9	
3	8	
4	7	
5	9	
6	6	
Total:	44	

Problem 1. *5pts.*

In each of the following parts, write down the output from running the code.

Box your answer to separate it off from any scratch work.

(a)
```
L1 = [0]

L2 = L1
L2.append(1)

L3 = L2 + []
L3.append(2)

print(L1, L2, L3)
```

(b)
```
def f(L1):
    L2 = L1

    for i in range(len(L1)):
        if L1[i]  > i: L2[i] =  1
        if L1[i] == i: L2[i] =  0
        if L1[i]  < i: L2[i] = -1

    # Yes, this type of spacing is okay;
    # I've done it this way to make it clearer for you.

    return L2

L = [8, 8]
print(f(L))
```

Problem 2. *9pts.*

In each of the following parts,

- **either** say "this results in an error";
- **or** write down the output from running the code (error free).

(a)
```
def f():
    x = y
    x.append(0)
    return x

x = []
y = [0]
z = f()

print(x, y, z)
print(x == z, x is z, y == z, y is z)
```

(b)
```
def f():
    x.append(0)
    x = y
    return x

x = []
y = [0]
z = f()

print(x, y, z)
print(x == z, x is z, y == z, y is z)
```

(c)
```
def f():
    x.append(0)
    y = x
    return y

x = []
y = [0]
z = f()

print(x, y, z)
print(x == z, x is z, y == z, y is z)
```

Problem 3. *8pts.*

Write down the output from running the following code.

There should be 4 lines, a space, and then another 4 lines.

Don't accidentally miss a line.

Box your answer to separate it off from any scratch work.

```
ob = 0

def f(p = ob):
    print(p)
    p += 1

ob += 2
ob = 3

f()
f()
f(4)
f()

#--------
print('')
#--------

ob = [0]

def f(p = ob):
    print(p)
    p += [1]

ob += [2]
ob = [3]

f()
f()
f([4])
f()
```

Problem 4. *7pts.*

(a) Define a `lambda` function `f` such that for positive `ints` `n`, `f(n)` is the list consisting of square numbers up to and including `n` squared: `[0, 1, 4, 9, 16, ..., n**2]`.

```
f =
```

(b) Consider the following code.

```
L = []
L1 = [L]
L2 = [L]

print(L1 is L2, L1[0] is L2[0])
```

Its output is `False True`.

Explain this output by **using a picture**. If you draw a picture like the ones I draw in class, then you will barely need any words because I'll know that you understand.

Bear in mind that I have not explained this example in class (or if I did, I did so fleetingly). Some deductions have to be made based upon the output I have told you.

Problem 5. *9pts.*

In each of the following parts, write down the output from running the code.

Box your answer to separate it off from any scratch work.

(a)
```
class A:
    L = []

    def __init__(self, i):
        self.L.append(i)
        self.L = [0]

    def f(j, k = 1):
        return k

a = A(2)
print(A.L, a.f(3))
```

(b)
```
class B:
    def __init__(self):
        self.i = 0
        self.__j = 1
        self.f()

    def f(self):
        print(self.__j)

class C(B):
    def __init__(self):
        self.i = 2
        self.__j = 3
        super().__init__()

    def f(self):
        print(self.i)

c = C()
print(c.i, c._C__j)
```

Problem 6. *6pts.*

Write down the output from running the following code.

If output stops due to an unhandled error:

- write the output up to that moment, and then write "Unhandled Error".

```
try:
    try:
        print('0')
        raise ZeroDivisionError
        print('1')

    except:
        print('2')
        raise
        print('3')

    print('4')

except ZeroDivisionError:
    try:
        print('5')
        raise ZeroDivisionError
        print('6')

    except:
        print('7')

    print('8')
    raise NameError

except NameError:
    print('9')

print('10')
```