

University of Washington
CSE 140 Data Programming
Winter 2013

Practice midterm exam

February 3, 2013

Name: Solutions _____

CSE Net ID (username): _____

UW Net ID (username): _____

This exam is closed book, closed notes. You have **50 minutes** to complete it. It contains 14 questions and 9 pages (including this one), totaling 100 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!

1 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 that caused the error
 - 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”.

1. `(4.0 + 5) * 6`

value *54.0*

error

frame

operator

arguments

2. `len(str(5 * 2)) + "10"`

value

error *cannot add an integer and a string*

frame *global*

operator *+*

arguments *2, "10"*

Execute Python statements

Execute each of the following code snippets, just as if they were written in a program. (Each one is a sequence of statements.) Your answer to each question contains up to 5 parts.

output write any output that it prints (before any error). This part might be blank. If it does not print any output.

If there is an error during execution:

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 that caused the error

arguments state the values to which the operator was being applied

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

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

output *[1, 2, 1, 2]*

error

frame

operator

arguments

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

output *[[1, 2, 3], [1, 2, 3]]*

error

frame

operator

arguments

```
5. lst = [1, 2, 3]
   myvar = lst[0]
   lst[0] = 18
   print lst
   print myvar
```

output *[18, 2, 3]*
1

error

frame

operator

arguments

```
6. myvar = 18
   lst = [myvar, 2, 3]
   myvar = 22
   print lst
   print myvar
```

output *[18, 2, 3]*
22

error

frame

operator

arguments

```
7. plane = ("Passengers", "Luggage")
   plane[1] = "Snakes"
   print plane
```

output

error *cannot change a tuple*

frame *global*

operator *tuple element assignment*

arguments *("Passengers", "Luggage") , "Snakes"*

```
8. numb3rs = ([1, 2, 3], [4, 5, 6])
   numb3rs[0][2] = 0
   print numb3rs
```

output *([1, 2, 0], [4, 5, 6])*

error

frame

operator

arguments

Note that an immutable object can reference mutable objects. The immutable object's elements are always the same Python object, but aren't always the same abstract value.

```
9. for x in [1, 2]:
    for y in [3, x]:
        print x, y
```

output 1 3

1 1

2 3

2 2

error

frame

operator

arguments

```
10. bar = 1
def foo():
    bar = 2
    foo = 3
    return bar
foo()
print foo()
print bar
```

output 2

1

error

frame

operator

arguments

11. total = 0

```
def sum1(n):  
    total = 0  
    for i in range(n):  
        total = total + i  
    return total
```

```
def sum2(n):  
    total = 0  
    for i in range(n):  
        total = total + i  
    print total
```

```
print sum1(5)  
print total  
print sum2(5)  
print total
```

output 10

0

10

None

0

error

frame

operator

arguments


```
12. def f1(n):
    print "A"
    return 2

print "B"

def f2():
    return 1
    print "C"

print f1(f2())
```

output B

A
2

error

frame

operator

arguments

```
13. x = 22
def reset_x():
    x = 0
reset_x()
print x
```

output 22

error

frame

operator

arguments

```
14. a = [42, 39, 123]
b = sorted(a, reverse=True)
a[0] = b
b[0] = 7
b.sort()
print a
```

output *[[7, 39, 42], 39, 123]*

error

frame

operator

arguments