CptS355 - Programming Language Design
Spring 2018

Sakire Arslan Ay, PhD

School of Electrical Engineering and Computer Science
Washington State University

Midterm 2 - Sample Exam

- Print your name and WSU ID below.

Name: ________________________________ WSU ID:___________________

Directions: Answer all of the questions. You may have **one 8 1/2 X 11 sheet of notes** during the exam. You may not use a computer, electronic portable device, calculator or phone during the test. Remove all caps and visors.

There are N major problems on M pages totaling X points. **Make sure that you have a complete exam before beginning.**

Write your name on your exam **now**.

Q1 (12pts)	**Q2 (15pts)**	**Q3 (12pts)**	**Q4 (8pts)**	**Q5 (20pts)**	**Q6 (12pts)**	**Q7 (15pts)**	**Q8 (6pts)**	**Total**

1) [12 pts] Directions: Consider the following Python code.

```
x = 0
def add(a):
   global x
   x = a+x
   return x

def sumAll (L):
   if not L:
      return 0
   else:
      return L[0] + sumAll(L[1:])

def grow (c):
   return sumAll (range (c, c +2))

def outer (f):
   def inner (g, y):
      return f(y)+g(y)
   return inner
```

For each of the following Python expressions, write the value to which it evaluates, assuming that the expressions are evaluated in a new session (i.e., answers **should not** depend on the previous lines)

a)
```
L=[1,2,3,4,5]
list(map(grow,L))
```

b)
```
L=[1,2,3,4,5]
list(map(add,L))
```

c)
```
x=1
outer(add)(grow,5)
```

d)
```
x=2
outer(grow)(add,x)
```

2) [15 pts] Consider the following Python function `stream_mystery`. It takes a function, an integer, and an infinite `Stream` as input and returns another `Stream` as output. (The definition of the `Stream` class we defined in class is available on the last page)

```
def stream_mystery(fn, n, s1):
   def compute_rest():
      return stream_mystery(fn, n, s1)
   t = n
   temp = 0
   while (t > 0):
      temp = fn(temp,s1.first)
      s1 = s1.rest
      t -= 1
   return Stream(temp, compute_rest)
```

(a) [5 pts] Explain what **`stream_mystery`** function does in a sentence or two.

(b) [10 pts] Evaluate the following expressions and write down the output of the print statements.

(1)

```
#generates a Stream of integers starting at 'first'.
def make_integer_stream(first=1):
   def compute_rest():
      return make_integer_stream(first+1)
   return Stream(first, compute_rest)

S2 = stream_mystery(lambda x,y: x+y, 2, make_integer_stream(1))

print(S2.first,S2.rest.first,S2.rest.rest.first,
S2.rest.rest.rest.first)
```

(2)

```
#generates a Stream of 1's.
def make_one_stream():
   def compute_rest():
      return make_one_stream()
   return Stream(1, compute_rest)

S3 = stream_mystery(lambda x,y: x+y, 3, make_one_stream())
print(S3.first,S3.rest.first,S3.rest.rest.first,
S3.rest.rest.rest.first)
```

3) [12 pts]

(a) [6 pts] Define a Python function, `aroundL`, that takes a list of integers as input and returns a list of pairs containing one more and one less than each number in the list. For example, `aroundL ([1,2,3])` should return `[(0,2),(1,3),(2,4)]` as its answer. Your function may involve a loop or can be recursive.

```
def aroundL (L):
```

(b) [6 pts] Re-write the `aroundL` function from problem 4(a) using a list comprehension.

4) [8 pts] Define a Python function wordCount that is given a list of words (strings) as an argument and returns a dictionary mapping each word to the number of times it appears in the list. Remember to use the dictionary get operation when appropriate to eliminate an if-else.

```
def wordCount(L):
```

5) **(Postscript) [10 points]** Describe what occurs when each `def`, `dict`, `begin`, `end`, and `mul` operation is executed in the following PostScript program. You must list the operations **in the order they are executed.** For each one say which line number it occurs on and what operands it uses. The first one is done for you as an example. (Of course, the line numbers are not part of the program. There are more blank lines than required answers.)

```
Line
 0       /y 3 def
 1       /f { /z y def  1 dict
 2            begin
 3           /z 4 def
 4            /y 5 def
 5            z y mul
 6           end
 7         } def
 8      f y mul
```

Line	Operation	Operands
0	def	/y 3
__	___	_______
__	___	_______
__	___	_______
__	___	_______
__	___	_______
__	___	_______
__	___	_______
__	___	_______
__	___	_______
__	___	_______

5b) **[5 pts]** What values are on the operand stack when the program finishes execution?

5c) **[5 pts]** What is on the dictionary stack when the program finishes execution?

6) (Scoping) Consider the ML code:

```
val x = 6
fun addx y = y + x (* HERE *)
fun map f [] = [] (* this is just the standard map: no tricks *)
  | map f (x::rest) = (f x) :: (map f rest)
val z = (map addx [1])
```

(a) (6pts) Knowing that ML uses static scoping, what value will be bound to **z** at the end of this code? Explain your answer.

(b) (6 pts) Draw the stack of activation records (ARs) as it exists at the point labeled **(*HERE*)** in the code above. The intention is that you draw a stack of ARs that shows the definitions of `x`, `addx`, `map` and `z` as well as the call from the last line to map and the call from map to addx.)

In your drawing clearly identify in each AR:

- the dynamic link
- the static link
- the names of any variables held in that AR, and
- the values of those variables.

For ARs that correspond to function calls, label the AR with the function name and arguments. For ARs that contain function definitions, show the code body of the function as "…".

7) **[15 pts]** Below is a sequence of four Python assignments. Following those assignments are a number of questions. Answer each question assuming that the 4 assignments have been performed immediately prior to executing the code in that question. The answer to some of the questions may be “an error occurs” in which case describe why it is an error.

```
L = [1, 2, 3]
S = (1, 2, 3)
C = ”1 2 3”
D = {3:4, 2:5, 1:6}
```

a) What is L[-1] ?

b) What is C[1] (make sure you give an answer of the right type!)?

c) What is S[1:-1] ?

d) What is the value of C after executing C[1] = ”3” ?

e) What is the value of D[3]?

8) **[6 pts]** After the following Python code:

```
L = [1, 2, 3, 4]
M = L + [5, 6]
L[2:3] = [5, 6]
L[2] = [7,8]
```

a) What is the value of M?

b) What is the value of L?

```
class Stream(object):
   def __init__(self, first, compute_rest, empty= False):
      self.first = first
      self._compute_rest = compute_rest
      self.empty = empty
      self._rest = None
      self._computed = False

   @property
   def rest(self):
      assert not self.empty, 'Empty streams have no rest.'
      if not self._computed:
         self._rest = self._compute_rest()
         self._computed = True
      return self._rest
empty_stream = Stream(None, None, True)
```