Forewarning: I *tend* to make my practice questions harder than the standard questions that you will see. The actual exam questions are generated solely by the Professor, any matches are pure coincidence.

1. Consider the following code segment:

```
li = [“I”, “can’t”, “fix”, “printers”]
start = 0, end = len(li)
for item in li:
     if len(item) % 2:
          start += 1
          end -= 1
     else:
          start -= 1
          end += 1

print(li[start:end])
```

Assume this code is loaded up and executed, what is the output?

a. [‘fix’]
b. [‘can’t’, ‘fix’]
c. []
d. Nothing is printed to the terminal
e. There is an error in the code

Explanation: We initialize the list with 4 elements as well as start to 0 and end to len(li) which in this case is 4. We then loop through each element in the list. If the item has an odd length we increase start by 1 and decrease end by 1 and do the reverse for even length words. Here is the values of start and end after each iteration of the loop:

start = 1, end = 3
start = 2, end = 2
start = 3, end = 1
start = 2, end = 2

We then print out the section of the list starting from index 2 going to but not including index 2 yielding the empty list

2. What is the value stored in x after the following code:

```
x = (2023 - 23) % 2 * 29 / 5 + 32
```

a. 95.0
b. 32
c. 32.0
d. 67
e. Cannot be determined

Explanation: This question tests your ability to apply the order of precedence (Operator Precedence in Python - Python Geeks) This plus the fact we are using floating point division '/' yields 32.0

3. This one is easy! What is the type of foo after running the following code:

```
foo = {}
```

a. List
b. DefaultDict
c. Set
d. Dictionary
e. We don't know until we populate it

Explanation: This is just lecture recall, you were taught that the default type when using the {} is a dictionary

4. What is the value of x after the following code executes:

```
def foo(a, b):
    while a <= b:
        a += 1 #a = a + 1
        b -= a
    return a*b

x = print(foo(5,7))
```

a. 35
b. 6
c. 42
d. “6”
e. There is an Error in the Code*

Explanation: This course assumes the print functions as it does in Python 2.7 meaning you cannot make assignments with functions that do not return anything, print is the prime example.

```
*File "main.py", line 9
 x = print(foo(5,7))
       ^
SyntaxError: invalid syntax
```

This is an error in Python 2.7 not in Python 3+