

Name: _____

Directions: MAKE SURE TO COPY YOUR ANSWERS TO A SEPARATE SHEET FOR SENDING ME AN ELECTRONIC COPY LATER.

1. (10) Fill in the blank: The examples on p.30 show that in general Python is not a _____ language.

2. Consider the grading program example, pp.22-23.

(a) (10) Suppose there is an input line equal to line 13 in the source code, for a student John Paul Jones (without the #). Then in line 66, what value will `lw` have?

(b) (10) The instructor has the option of discarding some quiz grades. Give the line number on which this is done.

(c) (10) What is the purpose of the -1 in line 81?

3. (30) A common operation on file names is to extract a *suffix*, e.g. extracting '.c' from 'xyz.c'. The rest of the file name is typically called the *basename*. The code below will return a Python list consisting of the basename and the suffix. Fill in the blanks.

```
def fnamesplit(fname):
    fnamesp = blank (a)
    suff = blank (b)
    bname = blank (c)
    return [bname, suff]
```

Examples:

```
>>> fnamesplit('abcde.f')
['abcde', '.f']
>>> fnamesplit('abcde.f.1.23')
['abcde.f.1', '.23']
```

4. (30) Consider the vending machine value on p.28. Every time we add stock for new type of item, we will need to update our data. The code in the book needs to be modified for this, as shown below. Fill in the blanks:

```
def newstock(self, newitems):
    for itm in newitems.keys():
        blank (a):
            self.inventory[itm] += newitems[itm]
        blank (b):
            blank (c).append(itm)
        blank (d)
```

Example:

```
>>> m1 = machine()
>>> machine.itemnames
[]
>>> m1.newstock({'Kit Kat':5, 'Sun Chips':8})
>>> machine.itemnames
['Sun Chips', 'Kit Kat']
>>> m1.inventory['Sun Chips']
8
>>> m1.newstock({'Fritos':2, 'Sun Chips':8})
>>> m1.inventory
{'Sun Chips': 16, 'Fritos': 2, 'Kit Kat': 5}
>>> m2 = machine()
>>> m2.inventory
{'Sun Chips': 0, 'Fritos': 0, 'Kit Kat': 0}
```

Solutions:

1. functional

2.a 15

2.b 53

2.c This deletes the EOL character, which would be make an extra line print out.

3.

```
def fnamesplit(fname):
    fnamesp = fname.split('.')
    suff = '.' + fnamesp[-1]
    bname = fname[0: (len(fname) - len(suff)) ]
    return [bname, suff]
```

4.

```
class machine:
    itemnames = []
    def __init__(self):
        # in (itemname, stock) form
        self.inventory = {}
        for nm in machine.itemnames:
            self.inventory[nm] = 0
    # adds the new stock to inventory; items is in dictionary form,
    # (itemname, newstock form)
    def newstock(self, newitems):
        for itm in newitems.keys():
            # as alt approach, could do something like
            # if itm in self.inventory...
            try:
                self.inventory[itm] += newitems[itm]
            except:
                machine.itemnames.append(itm)
                self.inventory[itm] = newitems[itm]
```