

Name: _____

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

1. This problem concerns the program **psax.py**, which monitors processes, in the **curses** chapter of our book.

- (a) (5) State the number of a line in which an instance variable is accessed. If there is no such line, write NONE.
- (b) (10) State the number of a line in which a class variable is accessed. If there is no such line, write NONE.
- (c) (10) State the number of a line in which a class method is accessed. If there is no such line, write NONE.
- (d) (10) Suppose the file **gy** consists of a single line with contents `'uuddkkr\n'`. State in a SINGLE, BRIEF line what will happen.

2. (30) Consider the binary tree example, Section 1.20. We will add a new method **max()** to the class **treenode**. Note since it is a method rather than a freestanding function, it will not conflict with the built-in Python function **max()**, which works as follows:

```
>>> max(12,5,13)
13
```

If **z** is an object of the class **treenode**, then

```
z.max()
```

will return the maximum value in the tree rooted at **z**.

Example:

```
>>> x = [12,5,13,10,8,6,28]
>>> tr = bintree.tree()
>>> for n in x: tr.insrt(n)
>>> tr.root.max()
28
```

Fill in the blanks:

```
def max(self):
    s = blank (a)
    if blank (b) : s = blank (c)
    return s
```

3. (25) Here we will deal with a class representing a vending machine. Each object of this class represents one machine, but all the machines carry the same items (though the current size of the stock of a given item may vary from machine to machine).

The **inventory** variable will be a dictionary with keys being item names and values being the current stocks of those items, e.g. 'Kit Kat':8 signifying that this machine currently holds a stock of 8 Kit Kat bars.

The method **newstock()** adds to the stocks of the given items; e.g. `m.newstock({'Kit Kat':3,'Sun Chips':2})` would

record that the stocks of Kit Kat bars and bags of Sun Chips at machine **m** have been replenished by 3 bars and 2 bags, respectively.

Fill in the blanks:

```
class machine:
    itemnames = []
    def __init__(self):
        self.inventory = blank (a)
        for nm in blank (b) :
            self.inventory[nm] = 0
    def newstock(self,newitems):
        for itm in blank (c) :
            blank (d) += blank (e)
```

4. (10) This is a continuation of Problem 3. The following test of the above code produces an error:

```
>>> m = machine()
>>> machine.itemnames = ['a','b']
>>> m.newstock({'b':3})
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 14, in newstock
KeyError: 'b'
```

State in a SINGLE, BRIEF line how to fix this test.

Solutions:

1. Note that **os** and **curses** are modules, not classes, as can be seen by the fact that they are **imported**.

1a. NONE

1b. any line containing "gb."

1c. NONE

1d. The **psax.py** program itself would be killed!

2a.

self.value

2b.

```
self.right != None
```

2c.

```
self.right.max()
```

3a.

```
{}
```

3b.

```
machine.itemnames
```

3c.

```
newitems.keys()
```

3d.

```
self.inventory[itm]
```

3e.

```
newitems[itm]
```

4. Swap the first two lines.