

Name: \_\_\_\_\_

**Directions: Work only on this sheet (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing. In order to get full credit, SHOW YOUR WORK.**

1. (15) Which of the following would be legal/illegal if inserted into the program **tfe.py** in Sec. 5.1 of our Python tutorial PLN? Write “legal” or “illegal.” Each is independent of the others, i.e. you are not being asked whether it would be legal to insert more than one of them together.

- (a) : `b.idnum = 1` after line 23
- (b) : `w = a.name + b.name` after line 23
- (c) : `b = 12` after line 27 (not indented)

2. (10) Suppose an exception occurs which our program does not handle using **try...except**. The name of the default function which is called at that time is \_\_\_\_\_.

3. (10) Write a single line of code which produces **cmdlnargs**, a list consisting of all the command-line arguments excluding the name of the program being run. (Note: Code like `if x == y: z = 8` counts as two lines, as if the second part were on a new line.)

4. (10) In our Python network programming PLN, we didn't call an analog of the function **gethostbyname()** seen in our network tutorial PLN, because that function's actions are included in the Python function \_\_\_\_\_. Similarly, the information returned by **getpeername()** in the network tutorial is included in what is returned by the Python function \_\_\_\_\_.

5. (15) Fill in the blanks in the following pair of functions which find all files of a given name (regardless of whether they be ordinary files, directories or links) in a given directory tree. Each file is reported in terms of the path relative to **dtroot**. Example: The tree starts at `/z/a`, which contains an ordinary file **b** and a directory **c**; the latter contains a directory **b**; the program is run from within **c**, with `..` as **dtroot** and `b` as **targetfile**. Then the output will be

```
../b
../c/b
```

Here is the code:

```
import os, sys

def checkthisdir(targetfile,dr,flst):
    if _____:
        print os.path.join(_____)

def findfile(dtroot,targetfile):
    os.path.walk(_____)
```

6. (20) Fill in the blanks in the following lines of code, which returns a list of all words in a text file. It is presumed that the file has already been opened by the calling program, with **fi** being assigned the result of **open()**, but no other operations have been performed on the file.

```
def getwords(fi):
    wrds = map(_____)
    return reduce(_____)
```

For example, if called on the file

```
a b cd
xyz

uu vv
```

the value returned should be `['a', 'b', 'cd', 'xyz', 'uu', 'vv']`.

7. (20) Suppose you are debugging a multi-module program by using **pdbw.py**. You wish to know how many breakpoints you've set in each module. You will be able to do so using a module **bpl.py**. For example, in debugging our example of **tf.py** and **tfest.py** in Sec. 9.1.1 of our Python tutorial PLN, we might have:

```
(Pdb) import bpl
(Pdb) bpl.bpc(debugger)
/a/b/tf.py 2
/a/b/tftest.py 1
```

Fill in the blanks in **bpl.py**:

```
def bpc(dbg):
    for f in _____:
        print f, _____
```

**Solutions:**

1.a Legal. Classes are implemented as dictionaries, and thus can be added to at any time.

1.b Legal. This is just string concatenation.

1.c Legal. The variable **b** is just a reference, i.e. pointer. You can use it to point to whatever type you wish.

2. `sys.__excepthook__()`; half credit for `sys.excepthook()`

3.

```
cmdlnargs = sys.argv[1:]
```

4. `socket.connect()`; `socket.accept()`

5.

```
import os, sys

def checkthisdir(targetfile,dr,flst):
    if targetfile in flst:
        print os.path.join(dr,targetfile)

def findfile(dtroot,targetfile):
    os.path.walk(dtroot,checkthisdir,targetfile)
```

6.

```
def getwords(for instance):  
    wrds = map(lambda u:u.split(),fi)  
    return reduce(lambda x,y: x+y,wrds)
```

7.

```
def bpc(dbg):  
    for f in dbg.breaks.keys():  
        print f, len(dbg.breaks[f])
```