

582332: Programming in Python, Fall 2009

Exercise sheet 3
21-25.9.2009

Note that the time and place of the exam has changed. The new time and place is on Tuesday 20th of October at 16-19 in rooms A111 and B123 of Exactum.

1. Create a program that can compute the areas of three shapes, triangles, boxes and circles, when their dimensions are given.

An endless loop should ask for which shape you want the area be calculated. If the user gives a string that is none of the given shapes, the message "unknown shape" should be printed.

Then it will ask for dimensions for that particular shape. When all the necessary dimensions are given, it prints the area, and starts the loop all over again.

The dimensions of the shapes must be positive. If this doesn't hold, use the statement `raise ValueError` to signal erroneous parameters.

You can use the call `sys.stdin.readline()` to read a line from a user, once you have imported the module `sys`. Note that this returns a string that ends with a newline character (`\n`).

2. Suppose we have two lists `L1` and `L2` that contain integers which are sorted in increasing order. Create a function `merge` that gets these lists as parameters and creates a new sorted list `L` that has all the elements of `L1` and `L2`. So, `len(L)` should equal to `len(L1)+len(L2)`. Do this using the fact that both lists are already sorted. You can't use the `sorted` function or the `sort` method.

Remember to add a docstring for your function. You should do this always when creating new public functions.

3. Create a function named `detect_ranges` that gets a list of integers as parameter. The function should then sort this list, and transform the list into a string where a shorthand is used for all the detected intervals. So `3,4,5,6` is replaced by the string `"3-6"`. Numbers that are not part of any interval result just single numbers. The resulting list will be a concatenation of these number and interval strings, separated by commas.

An example of how the function works:

```
>>> detect_ranges([2,5,4,8,12,6,7,10,13])
"2,4-8,10,12-13"
```

4. Use a list comprehension to create the list of pairs from last week's exercise 5.
5. Let `d` be a dictionary that has English words as keys and a list of Finnish words as values. So, the dictionary can be used to find out the Finnish equivalents of an English word in the following way:

```
>>> d["move"]
["liikuttaa"]
>>> d["hide"]
["piilottaa", "salata"]
```

Make a function that creates a Finnish to English dictionary based on a English to Finnish dictionary given as a parameter. It should work like this:

```
>>> d={"move":["liikuttaa"], "hide":["piilottaa", "salata"]}
>>> reverse_dictionary(d)
{'liikuttaa': ['move'], 'salata': ['hide'], 'piilottaa': ['hide']}
```

Be careful with synonyms.

6. Use the string-formatting expression to print the following:

- π in decimal form using precision 4
- same as above but use field width 2
- same as the first except use field width 10 where the number is positioned on the left
- the tuple (1,2,3)

You can find the constant `pi` in module `math`.

7. Read sections “Call by value”, “Call by reference” and “Call by sharing” from page <http://en.wikipedia.org/wiki/Pass-by-value>.

Then run the following program, and decide whether the function argument passing of Python is by value, by reference or by sharing.

```
def f(a):
    print "Executing f"
    a=[4,5,6]
    print "a=%s\n" % a

def g(a):
    print "Executing g"
    a.append(10)
    print "a=%s\n" % a

i=[1,2,3]
print "i=%s" % i
f(i)
print "i=%s" % i
g(i)
print "i=%s" % i
```

. Explain why.

8. Create a class called `Prepend`. We create an instance of the class by giving a string as a parameter to the initializer. The initializer stores the parameter in an instance attribute `start`. The class also has a method `write(s)` which prints the string `s` prepended with the `start` string.

An example of usage:

```
>>> p=Prepend("+++ ")
>>> p.write("Hello");
+++ Hello
```