

Testing Lists, Tuples, NamedTuples

Introduction to Computer Science!



What is the output of this code

```
#doubling.py
def dbl( x ):
    return x*2
print(doubling.dbl(50))
```

- A. 100
- B. 50
- C. Error
- D. No output

Program Bugs



- Syntax or logic errors in programs that prevent correct behavior
- Expect bugs to show up (its normal!)
- Learn to find and squash them (debug)

Writing bug-free code via testing

```
#test dbl.py
import pytest
def dbl( x ):
    return 42

def test dbl_1():
    assert dbl(0)==0

def test dbl_2():
    assert dbl(2)==4

def test dbl_3():
    assert dbl("UCSB")=="UCSBUCSB"
```

Run these tests from the unix command line:

```
$python3 -m pytest test dbl.py
```

Demo

- In class we will code a few functions and test them using pytest
- Square a number
- Find the area of a circle with radius r

Floating point inaccuracies

```
import math  
  
print(math.sqrt(2))  
print(math.sqrt(2) * math.sqrt(2) == 2)
```

What is printed by the last line?

- A. True
- B. False
- C. Error

Storing sequences

- Strings – stores sequence of characters
- Lists – stores sequence of any type (including mixed types)
- Tuples – Similar to lists with the difference that they cannot be modified
- NamedTuples – Like tuples but more convenient way of indexing

Lists

- Lists - A list is a collection of multiple values (similar to how a str is a collection of characters).
- Note: In python, lists can be of heterogenous (different) types
- Lists can have duplicate values
- The elements of a list can be modified (lists are mutable)

Concep Test

```
fruits = ["apple", "banana", "orange"]  
fruits[1] = "pear"  
print(len(fruits))
```

What is the output of this code?

- A. 1
- B. 3
- C. 4
- D. None of the above

List methods

Type `dir(list)` to get all the methods:

```
[...'append', 'clear', 'copy', 'count',
'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']
```

```
>> help(list.sort)
```

Help on method_descriptor:

```
sort(...)
```

```
    L.sort(key=None, reverse=False) -> None
```

```
-- stable sort *IN PLACE*
```

Concep Test

```
fruits = ["apple", "banana", "orange"]  
fruits[-1] = 3  
print(fruits.count(3))
```

What is the output of this code?

- A. 1
- B. 3
- C. 4
- D. Error

Tuples

- Similar to lists: store a sequence of elements

lst = [10, 20] //ex of a list

tup = (10, 20) //ex of a tuple

- Elements are ordered and can be accessed using the appropriate index

tup[0]

tup[1]

- Different from lists in the following ways

- Can't change an element in the tuple

- Can't sort the elements in a tuple

Concep Test

```
fruits = ("apple", "banana", "orange")
fruits[-1] = 3
print(fruits.count(3))
```

What is the output of this code?

- A. 1
- B. 3
- C. 4
- D. Error

Named Tuples

- Used to package data with multiple attributes: e.g. representing a student in your program
- A student's attributes may be: name, perm number, major etc.
- Named tuples make it easier to access each attribute

Named Tuples

```
from collections import namedtuple
```

```
#Design your named tuple object
```

```
Student = namedtuple('Student', 'name perm  
major gpa')
```

```
# Create objects of type Student
```

```
s1 = Student("Jack", 123443, CS, 3.8)
```

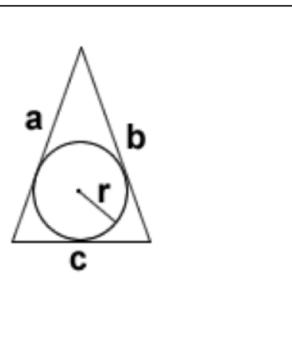
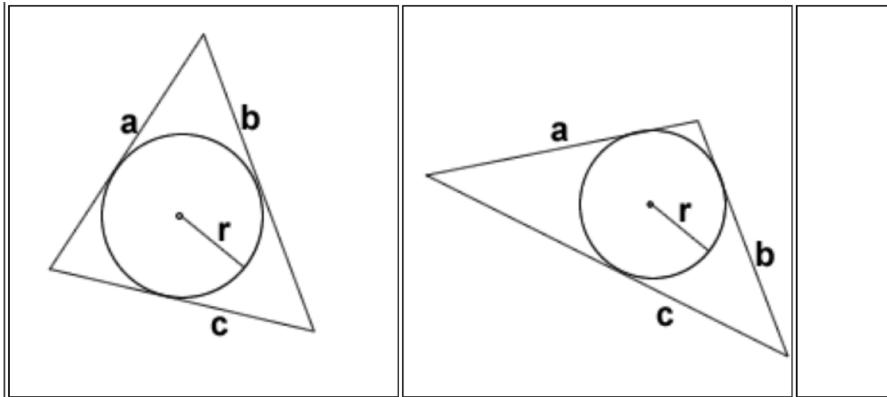
```
s2 = Student("Mary", 8932737, CE, 3.9)
```

```
# Access the elements of the objects
```

```
print(s1.name, s1.perm)
```

Put it to practice

- Inside *every* triangle (it doesn't have to be any particular kind of triangle), it is possible to *inscribe a circle* as shown in the three below.



radius of circle inscribed in triangle,
given sides of triangle, a,b,c

First, let:

$$s = \frac{a + b + c}{2}$$

Then:

$$r = \sqrt{\frac{(s-a)(s-b)(s-c)}{s}}$$

- Write a function to computer the radius of a circle inscribed in a triangle