Question #1

i)
```
def volume(x):
        if ( x > 0 and isinstance(x ,(int,float))):
            return x * x * x
        else: return 0
```

```
  def test_Volume(self):
        self.assertEqual(HomeWorkFour.volume(5), 125)
        self.assertEqual(HomeWorkFour.volume(-100), 0)
        self.assertEqual(HomeWorkFour.volume(4.5), 91.125)
```

ii) The initial test is checking that the volume function is performing the way it is supposed to be when used for the most basic inputs. Followed by a test that checks for negative values, which will only return a zero. The third test checks for float type values, to determine that the correct value is returned when calculated.

Question #2

i)
```
def average(list):
      if len(list) > 0:
       result = sum(list) / len(list)
       return result
      else: return 0
```

```
   def test_Average(self):
        list0 = [1,2,3,4,5,6,7,8,9]
        list1 = []
        list2 = [-2,-1,0,1,2]
        self.assertEqual(HomeWorkFour.average(list0), 5)
        self.assertEqual(HomeWorkFour.average(list1), 0)
        self.assertEqual(HomeWorkFour.average(list2), 0)
```

ii) The initial test checks for proper functionality with a basic list given, meaning only positive numbers. The next test is taking into consideration empty lists, which would lead to the problem of dividing by zero, with the way my code is written. If an empty list is given the return value should be zero. The third test checks the performance of the average function if positive and negative numbers are given.

Question #3

i)
```
def fullname(x,y):
      if(isinstance(x,str) and isinstance(y,str)):
            z = x + " " + y
            return z
      else: return 0
```

```
def test_Name(self):
      self.assertEqual(HomeWorkFour.fullname('Phillip', 'Renaud'), 'Phillip Renaud')
      self.assertEqual(HomeWorkFour.fullname('Santa', 4),0)
      self.assertEqual(HomeWorkFour.fullname(1, 2), 0)
```

ii) As with the previous questions, the initial test checks for the intended inputs given. I used my name because I would be able to determine if that was correct the best. The next test though, I check what would happen if one of the inputs was not a string, in that case a zero would be returned. The last test for the fullname function checks what would happen if the user just didn't understand what was going on and gave numbers instead of strings which would return a zero for completely failing.

Question #4

i) The component I would like to test from the system in my in-class activity 1 would be for the reactions on posts. For example, when a participant in the social media network sees a post they should be able to choose the reaction they want after hovering over the like button which will display a list of all the reactions available.

ii) The first example of unit test that I would write for this component would be checking if when the like button is hovered on do the reactions that are available get displayed in a side by side manner.
The second example of a unit test would be to check after the like button is hovered if I do not select a reaction the list of reactions go away so I can continue my scrolling spree. Finally, the third example of a unit test would be to check if I can leave a reaction to the post, which for this example we assume that all the reactions will work the same way but ideally each reaction should be tested. After the reaction is made the list should also disappear so the participant on the social media network can react to other posts as well.