**Exercise 3** – **Tapio Koskinen 1800509**

1. Explain following terms:
2. Abstraction

Abstraction in object oriented programming is showing essential information or attributes and hiding the non relevant. For example an object called user might have a lot of attributes, so with abstraction the relevant can be filtered.

1. Accessor and mutator methods

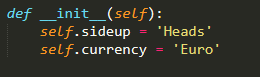
Internal data is often kept private, so the way a program can access private data is to create a class method called accessor, usually named with ‘get’ something. Mutator is a method that modifies data, like the value of a variable and is usually named ‘set’ something.

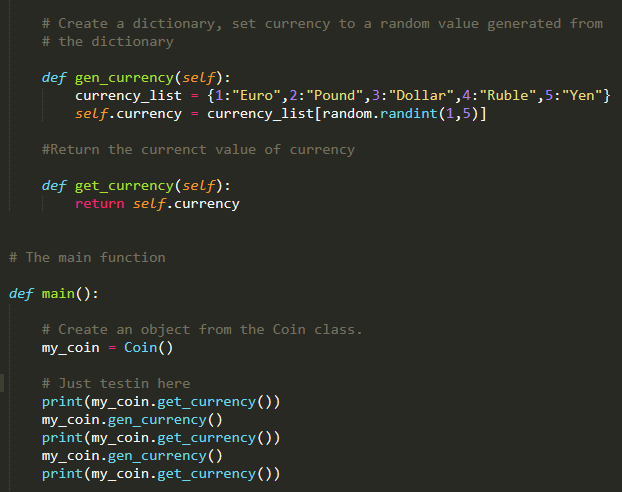
1. Public and private methods

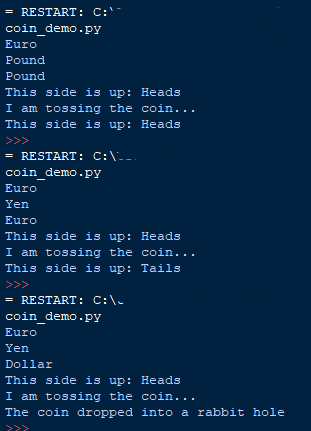
In OOP Public method can be used both inside **and** outside of the class. Everything that is **not** marked as private within a class can be called and used. A class that is marked private cannot be called outside of the class. To pass private methods or attributes, they must be handled inside the function.

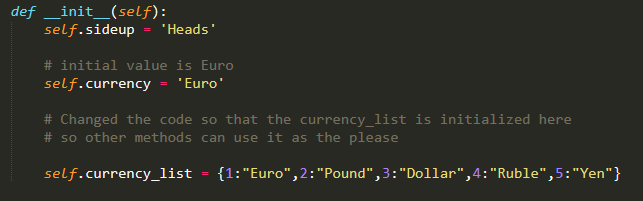
1. \_\_str\_\_ method (in python)

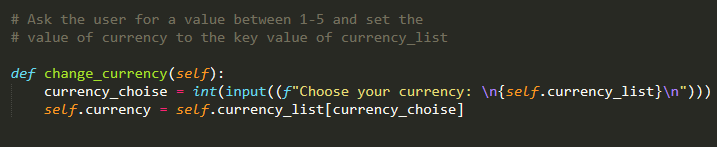
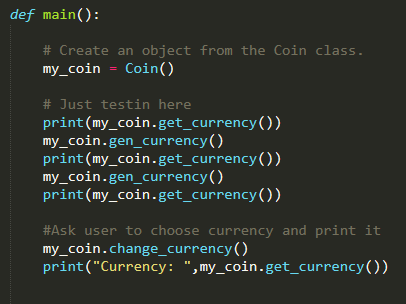
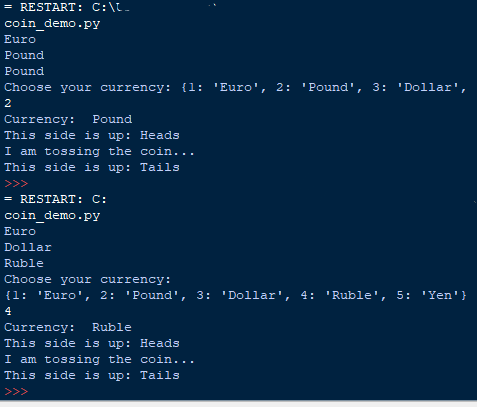
\_\_str\_\_ method, which is defined at the end of a class, returns a string when called. The interesting part is that it can take object values within a string and print the value for example “a {self.\_\_color} {self.\_\_animal}”, where color would be blue and animal bird, would print out “a blue bird” instead of the brackets etc.

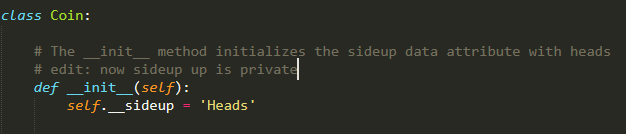
1. Modify the coin class to have currency attribute and currency generator  
   Fist create the currency  
   

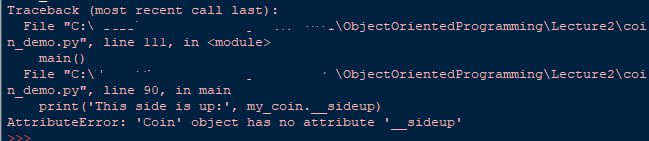
Next I created the methods to generate and get the current value  


On console(with the coin stuff)  


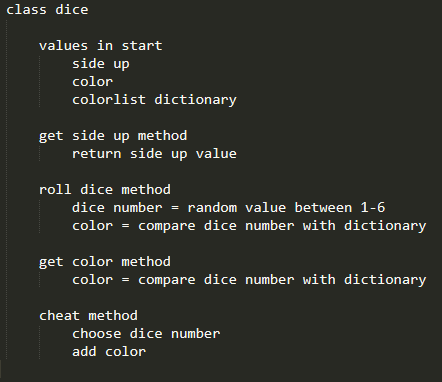
3) Choose currency method  


The method  
  
  
From the console  


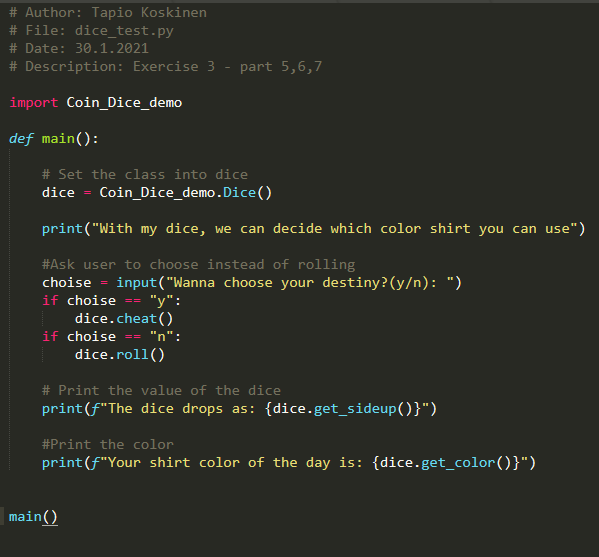
4) Change to private with the dunderscore(double underscore)  
  
Trying to get the attribute straight  

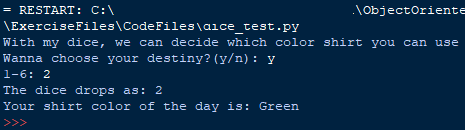
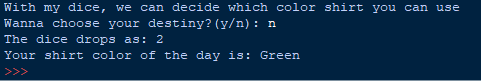

Gives an error  


5) Class Dice

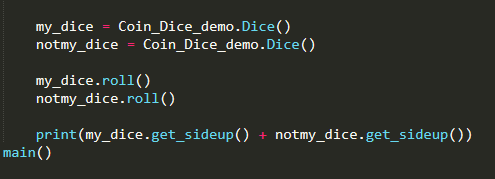
Pseudocode  


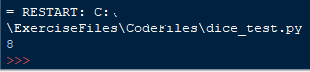
Class Dice, with the extra method cheat  


main  


Console  
  


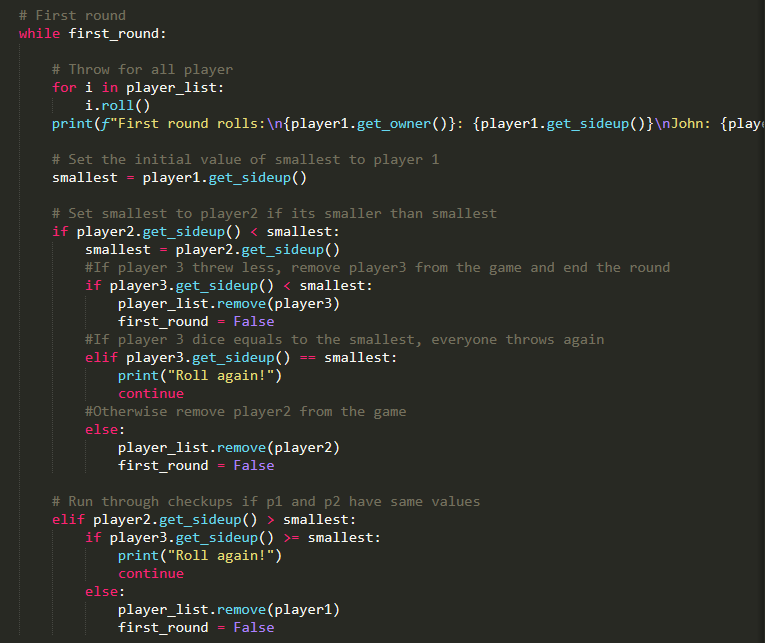
I just now realized the random value is the same as the userinput value, but it does generate random.

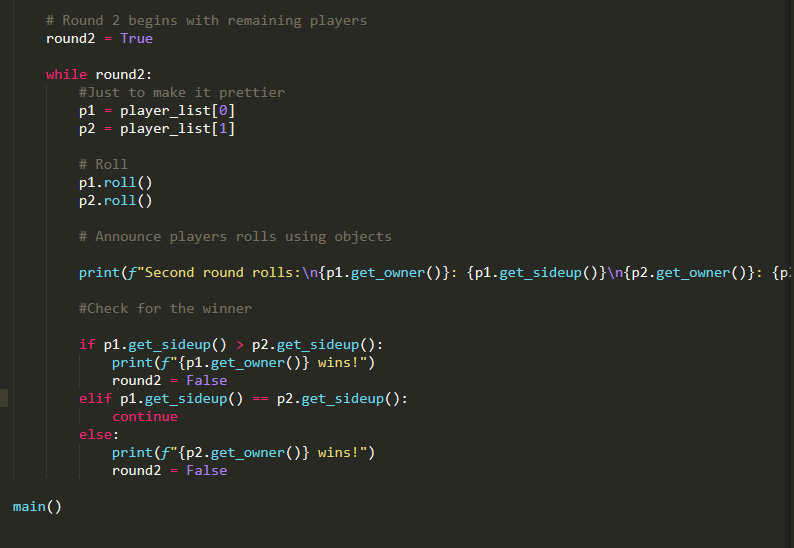
6) Two dice and sum  


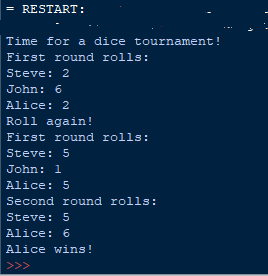
Console  


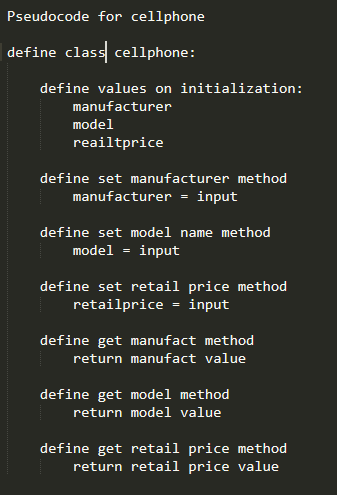
7) Dice game -- Pseudocode  

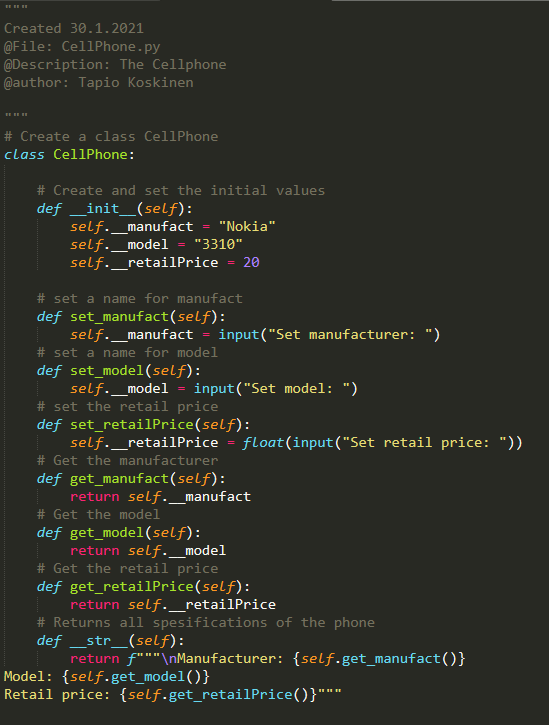

Initial values  



Console  


8) CellPhone – Pseudocode  


Class CellPhone  


Main  


Output:  


9)

a) Object: The my\_phone is an object

b) Encapsulation: The class Cellphone is an encapsulation

c) Data attributes: Manufacturer, model and retail price

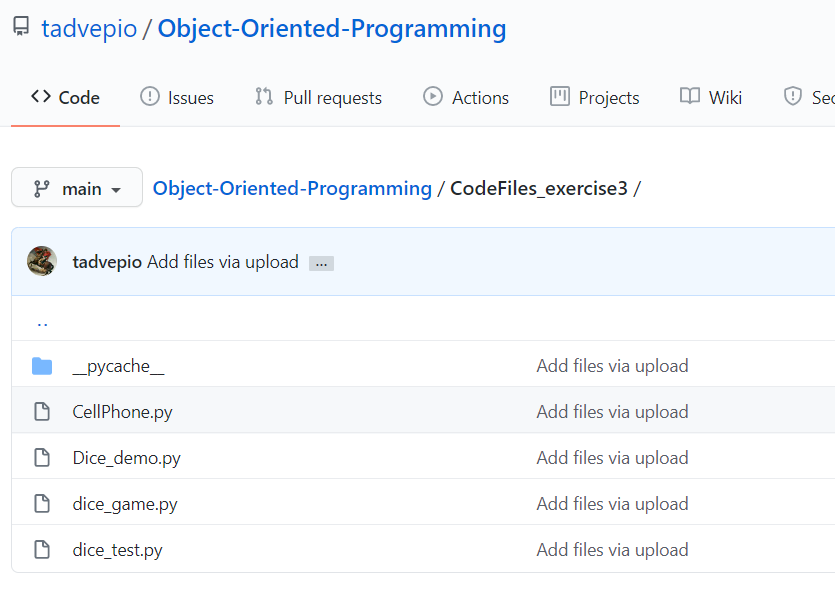
d) Hidded: The ones that are dundered(double underscore)

e) Public methods: All of the methods are public

f) Private methods: None are private, because none are specified private

g) Init method: \_\_Init\_\_ is in the beginning of the code, creating and assigning values to manufact, model, and retailprice.

Screen capture of the Git status:



Self-assessment:

I’m starting to understand the concept of object oriented programming better than in the beginning.

I did have trouble designing the dice game when juggling the values and comparisons and it could have been more elegant, but it works.

I feel more confident than before. This exercise took me the longest so far.