Fundamentals of Python programing:

chapter 9, section 10 (Exercises):

1. What is the difference between a class and an object?

=>class is a blueprint for objects and objects are instances of a class.

2. What are some other names for the term instance variable?

=>attributes and fields.

3. What is another name for the term method?

=>oparations.

4. What symbol associates an object with a method invocation?

=>to call a method in a object we use "dots", so class_name.method(parameter list)

5. How does a method differ from a function?

=>a method is ultimately a function, but it is defined and exists inside an object.

6. What method from the string class returns a new string with no leading or trailing whitespace?

=>the strip method, so: string_obj.strip()

7. What function returns the length of its string argument?

=>The len function, so len(string_obj)

8. What type of object does the open function return?

=>well, it returns a TextIOWrapper object which is in the io module, but we normally call it a file object.

9. What does the second parameter of the open function represent?

=>it is the mode we want to open the file with.
there are 3 modes:

"r" for read (just to read the file)

"w" for write (which creates a new file/ or deletes existing files data and starts fresh)

"a" for append (to append data to the file)

10. Write a program that stores the first 100 integers to a text file named numbers.txt. Each number should appear on a line all by itself.

=>

```
f = open('numbers.txt', 'w');

for i in range(100):

        f.write(f"{i}\n");


f.close();
```

11. Complete the following function that reads a collection of integers from a text file named numbers.txt. Each number in the file appears on a line all by itself. The function accepts asingle parameter, a string text file name. The function returns the sum of the integers in the file.

```
def sumfile(filename):
    # Add your code here . . .
```

=>

```
def sumfile(filename):

        f = open(filename, 'r');

        sum = 0;

        for line in f:
```

sum += int(line);

return sum;

f.close();

12. Provide the syntactic sugar for each of the following methods of the Fraction class:

(a) __sub__ *=>a.__sub___(b) is a - b*

(b) __eq__ *=>a.__eq__(b) is a == b*

(c) __neg__ *=>a.__neg() is -a*

(d) __gt__ *=>a.__gt__(b) is a > b*

13. How is using a Turtle object from Python's Turtle graphics module different from using the free functions; for example, t.penup() versus penup()?

=>the functionality of the methods remains the same,

by creating a Turtle object ourselves, we can manage multiple turtles and say pen colors in the same time.

the module turtle normaly creates the turtle object itself too, but it is a single hidden global object so since its global by calling the function/method we just do the same thing as t.penup(), and in this way we dont have multiple turtles to work with.

14. For each of the drawings below write a program that draws the shape using a Turtle object from Python's Turtle graphics module.

=>

```
from turtle import *
t = Turtle();
t.pensize(5)

for i in range(3):
        t.forward(200);
        t.left(120)

t.hideturtle();
exitonclick();
```

=>

```
from turtle import *
t = Turtle();
t.pensize(5)

t.left(36)
t.forward(300);

for i in range(4):
        t.left(144)
        t.forward(300);

t.hideturtle();
exitonclick();
```

=>

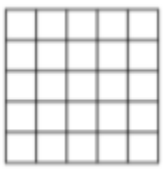
```
from turtle import *
t = Turtle();
```

```
t.pensize(5)

t.left(75);
t.forward(150);

right_bool = True;
for i in range(9):
        if right_bool:
                t.right(150);
        else:
                t.left(150);
        t.forward(150);
        right_bool =    not right_bool

t.hideturtle();
exitonclick();
```



=>

```
from turtle import *
```

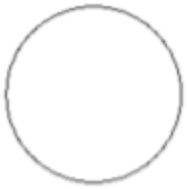
```
def create_square(amount):
    for i in range(amount):
        for i in range(4):
            t.forward(20);
            t.right(90);
        t.penup();
        t.forward(20);
        t.pendown();

t = Turtle();
t.pensize(5)

t.left(90);
x = 0;
for i in range(5):
    t.penup();
    t.setposition(x, 0);
    t.pendown();
    create_square(5)
    x += 20;
```

t.hideturtle();

exitonclick();



=>

*from turtle import **

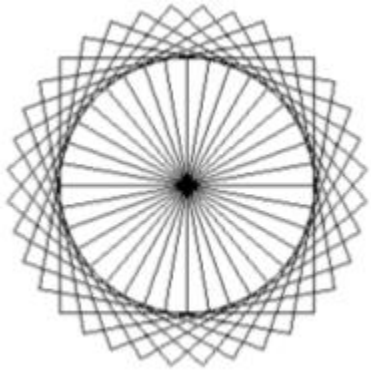
t = Turtle();

t.pensize(5)

t.circle(100);

t.hideturtle();

exitonclick();



=>

```
from turtle import *

def square():
        for i in range(4):
                t.forward(150);
                t.right(90);

t = Turtle();
t.pensize(3)
t.left(90);

for i in range(36):
        square();
        t.right(10);

t.hideturtle();
```

exitonclick();

15. Does Python permit a programmer to change one symbol in a string object? If so, how?

=>

we can not change the string object itself since its immutable,

but we can create a copy of it in which we can have the symbol changed.

one way could be:

```
s = 'te$st'

symbol = s.find('$');

new_symbol = '*'

new = s[:symbol] + new_symbol + s[symbol+1:];

print(new)
```

a better way would be using the method replace which return a copy of old string where all the symbols are replaced so:

```
s = 'te$st'

new = s.replace('$',"*")

print(new)
```

16. What would be the consequences if a turtle.Turtle object were immutable?

=> we could not change its attributes, like pencolor or pensize.

17. In the context of programming, what is garbage?

=> a piece of data that is no longer referred to by the program, and its value is lost (since nothing is referring to it) can not be used anymore, so it is considered garbage.

18. What is garbage collection, and how does it work in Python?

=>

python uses a reference counting garbage collection system.

which mean whenever a value is not referenced anymore (considered garbage) its space gets freed.

19. Consider the following code:

a = "ABC"

b = a

c = b

a = "XYZ"

(a) At the end of this code's execution what is the reference count for the string object "ABC"?

=>2, b and c.

(b) At the end of this code's execution is b an alias of a?

=>no since a is changed in the last line.

(c) At the end of this code's execution is b an alias of c?

=>yes, both refer to "ABC"