# Python \_ Homework 3

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

A class is a blueprint for declaring and creating objects. An object is a class instance that allows programmers to use variables and methods from inside the class. Memory is not allocated to classes.

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

The term "instance variable" is another name for non-static field. The term "class variable" is another name for static field.

1. What is another name for the term method?

Some common synonyms of method are fashion, manner, mode, system, and way. But in Python, methods are functions of a class, they can be called functions.

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

By dat.

1. How does a method differ from a function?

A function is a set of instructions or procedures to perform a specific task, and a method is a set of instructions that are associated with an object.

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

The Trim method removes from the current string all leading and trailing white-space characters.

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

he strlen() function returns the length of a string.

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

The open() function in Python opens a file stored at a specific location in the system and returns the file object.

The literal name of Python’s file class is TextIOWrapper from the io module.

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

The second parameter of the open() function is the mode , a string with one character. That single character basically tells Python what you are planning to do with the file in your program.

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.

P11 in my github

file = open("numbers.txt", 'w')  
for num in range(1,101):  
 file.write(str(num) + '\n')

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 a single parameter, a string text file name. The function returns the sum of the integers in the file.

P12 in my github

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

def sumfile(filename):  
 sum = 0  
 for num in filename:  
 sum = sum + int(num)  
 return sum  
  
file = open("numbers.txt", 'r')  
print(sumfile(file))

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

(a) \_\_sub\_\_ - : f - g

(b) \_\_eq\_\_ == : f == g

(c) \_\_neg\_\_ Sign : -g or -23

(d) \_\_gt\_\_ f > g : f greater than g

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()?

If we import like this ; import turtle we need write t.penup() to use

Method penup for object t ,but if we import like this ; from turtle import \* ; \* means every thing importing and we can write penup() without dat , python understands by itself.

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

P13 : Triangle

import turtle  
from turtle import \*  
t = turtle.Turtle()  
  
t.fd(200)  
t.lt(120)  
t.fd(200)  
t.lt(120)  
t.fd(200)  
  
exitonclick()

P14 : Grid

import turtle  
from turtle import \*  
  
t = turtle.Turtle()  
t.shape("circle")  
t.hideturtle()  
  
def draw(t, size, num):  
 for y in range(num):  
 for x in range(num):  
 for \_ in range(4):  
 t.fd(size)  
 t.lt(90)  
 t.fd(size)  
  
 parity = y % 2 == 0  
 turn = t.lt if parity else t.rt  
  
 turn(90)  
 t.fd(size \* 2 \* parity)  
 turn(90)  
  
screen = Screen()  
  
yertle = Turtle(visible = False)  
yertle.speed('fastest')  
  
draw(yertle, 25, 5)  
  
screen.exitonclick()

other code for P14

import turtle  
  
t = turtle.Turtle()  
t.shape("turtle")  
t.color("black","olive")  
win = turtle.Screen()  
t.speed(10)  
  
for i in range(4):  
 t.fd(300)  
 t.lt(90)  
  
t.penup()  
t.goto(0,60)  
t.pendown()  
t.fd(300)  
  
t.penup()  
t.goto(0,120)  
t.pendown()  
t.fd(300)  
  
t.penup()  
t.goto(0,180)  
t.pendown()  
t.fd(300)  
  
t.penup()  
t.goto(0,240)  
t.pendown()  
t.fd(300)  
  
t.penup()  
t.goto(60,0)  
t.pendown()  
t.lt(90)  
t.fd(300)  
  
t.penup()  
t.goto(120,0)  
t.pendown()  
t.fd(300)  
  
t.penup()  
t.goto(180,0)  
t.pendown()  
t.fd(300)  
  
t.penup()  
t.goto(240,0)  
t.pendown()  
t.fd(300)  
  
win.exitonclick()

P15 : Star

import turtle  
from turtle import \*  
  
t = turtle.Turtle()  
  
for i in range(5):  
 t.forward(150)  
 t.rt(144)  
exitonclick()

P16 : Circle

import turtle  
  
t=turtle.Turtle()  
s = turtle.Screen()  
t.circle(60)  
  
s.exitonclick()

P17

import turtle  
from turtle import \*  
  
t = turtle.Turtle()  
t.shape("turtle")  
t.color("black","red")  
  
t.lt(70)  
t.fd(100)  
for i in range(3+1):  
 t.rt(140)  
 t.fd(100)  
 t.lt(140)  
 t.fd(100)  
  
t.rt(140)  
t.fd(100)  
  
exitonclick()

P18

from turtle import \*  
  
bgcolor("pink")  
shape("circle")  
color("black", "purple")  
  
for i in range(60):  
 for j in range(4):  
 fd(100)  
 rt(90)  
 rt(6)  
exitonclick()

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

A Python String object is immutable, so you can't change its value. Any method that manipulates a string value returns a new String object.

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

In this case, the pointer (turtle) will not move and will not take an angle, and the turtle will not be created and... .

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

The garbage collector collects explicit references to python object and delete object with no references. It is simple enough to be implemented efficiently but cannot detect cycle references.

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

The garbage collector is keeping track of all objects in memory. A new object starts its life in the first generation of the garbage collector. If Python executes a garbage collection process on a generation and an object survives, it moves up into a second, older generation.

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 , c ; The value of a at the end of this code has changed.

(b) At the end of this code’s execution is b an alias of a? No ; because the value of a was changed.

(c) At the end of this code’s execution is b an alias of c? Yes ; Because there is an assignment operator between b and c.