AlgoHack #8

PROGRAMING OOP, TIME & FILES

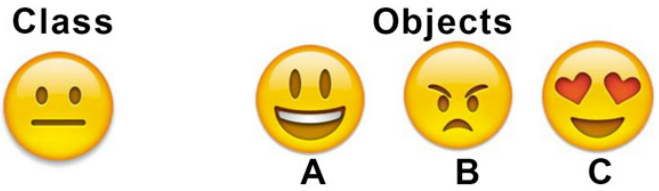
Authors
Niranjan Meegammana
N P Vishwa Kumara

Reviewers
Ravindu Ramesh Perera, Devanjith De Silva,
Prabhashana Hasthidhara, Yamuna Ratnayake.

AlgoHack aims to teach Computer Science and Programing to young people, initiated by Shilpa Sayura Foundation, supported by GOOGLE RISE and Computer Society of Sri Lanka..

Classes
A class is a group of data and functions.
A Class used to create run time objects.
Data in class are called **“attributes”.**
A function in a class is called a **"method"**

If human is a class.
We can create many **objects** from **human class.**



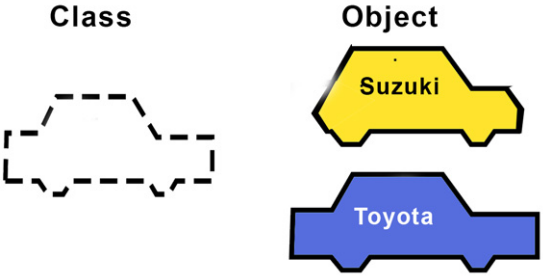
Human class has **common attributes:** 2 eyes, 1 mouth.
Human class has **common methods** : eat, sleep, run.
Each object has a **object-name** : A, B and C..
Each object has **own attributes:** funny, naughty or loving

Name 3 **types of classes** of things you find everyday.
Name **common attributes** of the class.
Name **common methods** for the class.
Create 3 **objects** from each.
What are the **attributes** of each object.

So far, we learned structured programming in python. **Object Oriented Programing** help make powerful programs. For example instead of passing many variables as parameters to functions, we can pass an object with attributes, as well as return objects.

Make your own car

The Car class has attributes and methods common to all objects created from the Car class. It is like a car mold. We can create many objects from a single class.




```
class Car(object):
    wheels = 4  # common data

    def __init__(self, make, model):
        self.make = make  # own attribute
        self.model = model

    def start(self):
        print("..brroom", self.make, self.model, "started")

# end of class definition
# main program
myCar = Car("Suzuki", "Swift")
yourCar = Car("Toyota", "Corolla")
print (myCar.make, "Wheels", myCar.wheels)
myCar.start()
print (yourCar.make, "Wheels", myCar.wheels)
yourCar.start()
```

def __init__() is special function called **constructor,** called whenever you create an object. The object will have passed and common attributes and class methods.

Study above Car program and explain what happens in each statement.

Modify the program to
1. **include** 2 more common attributes for the class.
2. **include** car registration number on creation.
3. **Write methods** for breaking, turning, speeding, reversing, sound and stop.
4. **Do you think** flowcharts adequate for OOP design?

OOP uses several method for **program design**

Use Cases	Create diagrams to show functionality of the system and user interactions.
Activity Diagrams	Show the control flow for each activity and their process.
Statecharts Diagrams	Show system responses to external events.
Sequence Diagram	Describes interactions between different classes with message/data exchange over time.
Class Diagram	Show classes in the system and their relationships.
Object Diagram	Snap shots of entities in the system at a given time.

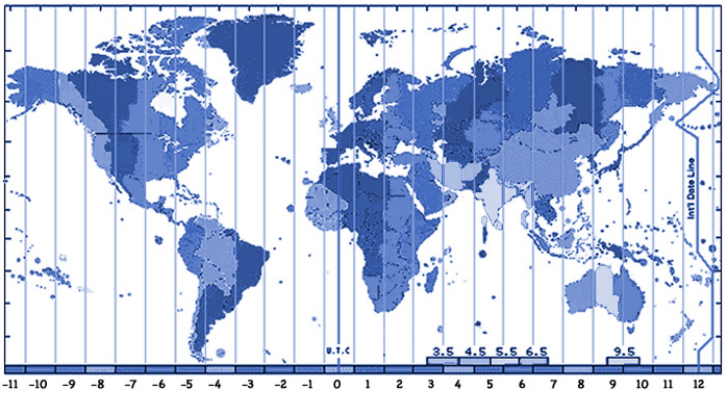
Time in Computers

A system clock inside your computer keeps count of the

number of ticks from a starting date, called the epoch. in Unix Time is the number of seconds passed since January 1970 00:00:00 UT. UT mean Universal time.
Unix time 1000000000 is 9 September 2001 01:46:40 UT.

Time Zones

We know when it's morning in Sri Lanka, America has night. This is because countries in the world are in different time zones. UT 0 is at Greenwich, London.



We need to add 5 hours and 30 minutes to UT to get Sri Lanka time. **Why ?**
If the time in Sri Lanka is 7AM in morning find **local time** in India, Singapore, China, Japan, Germany, and Portugal.
Name three countries has **multiple time zones.**
Why is it important us use **UT and Local Time** in programs running on websites?

Python date and time

The datetime class provides information on hour, minute, second, microsecond and timezone.

The datetime.time () function takes some parameters and output a **data object**. A data object is a variable in memory. It can have multiple values called **attributes.**

Run following code

```
import datetime
t = datetime.time(1,2,3)
print ('hour  :', t.hour)
print ('minute:', t.minute)
print ('second:', t.second)
print ('microsecond:', t.microsecond)
```

In above program identify object and attributes.

Python also has time class

```
import time
ut=time.time()
dt=time.ctime()
print (ut,dt)
```

Explain output. What are ut and dt variables?

Local Time

```
import time
lto=time.localtime()
print (lto)
```

Can you find Year, Month, Date, Hour and Minutes from the output lto?

The localtime() function in time class returns object lto. lto keeps local time attributes in an array.

What does lto[0] return ?.
What does lto.tm_year return?
How do you obtain hour, weekday and year date?
Why do we have attribute and array based methods?.

Challenge

Write a program to display time in H:M:S format.
Tip : convert numbers to string with str(n) function.
str(4) + "X" + str(5) returns string 4X5.

What is the local time of 172677777 Unix Time?
localtime = time.localtime(172677777) gives the answer
.
So we know **time.time() gives unix time** in python.
We can use it to build a clock.

```
import time
def dispclock():
    ut=time.time()
    localtime = time.localtime(ut)
    tm=str(lt.tm_hour) + ":" + str(lt.tm_min) + ":" +
str(lt.tm_sec)
    print (tm)

dispclock()
```

Explain above code.
If there is a bug, correct it.

What next ?
We have dispclock() funtion to display the time, what we need is to call this function in one second intervals.
This means we need a loop. This simple loop will do it.
Use ctrl +c to interrupt the program.

```
while (True) :
  dispclock()
```

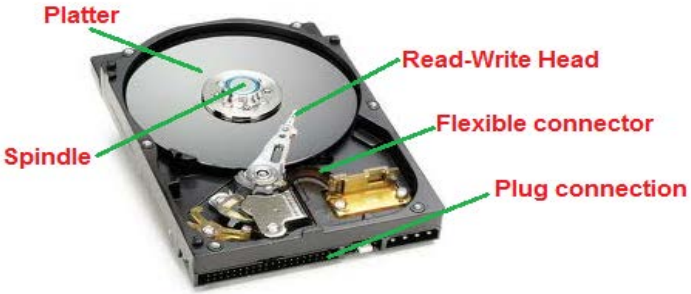
What are the problems you see in this program?
What should we do to improve it?

Making Python Sleep
The function **time.sleep(n),** stops running of program code for n seconds. time.sleep(5) will stop for 5 seconds.

Modify the program to display time in every second.

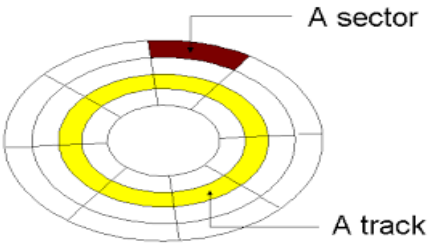
Computer Files

The data we store in memory is lost when program exists or computer is shut down. We use files to store **data on the disk** for later use. We use a **bit pattern** for every character called a **byte**. A bit has a positive or negative magnetic attribute represented by 1 and 0. Files are stored on disk tracks and sectors using **magnetism**.



When program asks for a file, the processor request the file from the disk. The read-write head moves along rotating disk, reads the file and return content in file. In the disk platter data is stored in sectors.

A file can occupy multiple **sectors** in **tracks**. Once we open a file, we can read, write or append data to a file.



Why we keep data in files in a computer ?
Describe 5 type of files in your computer.
Listen to disk rotating sound in your computer.
Create a cardboard platter with 5 tracks and 8 sectors per track. **Write** names of your friends in sectors.
Can you **suggest a mechanism** to read their names?

Does **memory sticks have rotating disks?**
Are they **different** from disk. **Why?**

Run this code

```
name=input(“Enter Your Name :”)
fileobject = open("name.txt", "w")
fileobject.write(name);
fileobject.close()
```

You just created name.txt file in disk, wrote data and finally closed it before exiting the program..

Modify this program to write your name and address.

Open() File

The **open function** takes two arguments, the **filename**

and the **mode**. The mode tells python, how the file is going to be opened. If you don't give mode, the file is opened for reading only. It is the default for open().

Mode	File operation
r	Opens a file for reading only.
w	Opens a file for writing only.
a	Opens a file for appending..
r+	Opens a file for both reading and writing.
w+	Opens a file for both writing and reading
a+	Opens a file for both appending and reading

Read Files

```
file = open("name.txt", "r")
for line in file:
   print (line),
```

The **read()** has three methods.

read()	returns whole file as one big string
readline	return one line at a time
readlines	returns all lines as a list object

The **write ()** method writes strings to a file.

write ()	Writes a string to a file
writelines()	Writes a list of strings to a file
append ()	append string or list of strings to a file

Close()
When you're done with a file, use close() to close it.
It will free up system resources used to open the file.
Imagine what happens if you keep a sugar bottle opened.

Describe differences in following programs

```
fileobject = open("name.txt", "r")
line=fileobject.readline()
print(line)
```

```
fileobject = open("file1.txt", "w")
line1 = "1st line"
line2 ="2nd line"
fileobject.write(line1)
fileobject.write("\n")
fileobject.write(line2)
fileobject.close()
```

```
fileobject = open("file1.txt.", "r")
lines= fileobject.readlines()
```

```
print(lines)
```

```
fileobject = open("file2.txt", "w")
lines = ["1st line", "2nd line", "3rd line"]
fileobject.writelines(lines)
fileobject.close()
```

Open file.txt and file2.txt and see.
Are they different? Why?

Append adds a string at the end of file (EOF).

```
fileobject = open("file2.txt", "a")
write("I was added to the end")
fileobject.close
```

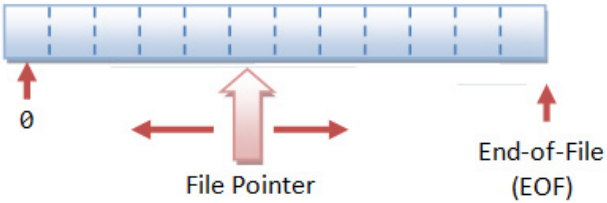
Miss Crow is late for School. Her teacher asks her to type “I will not be late for school” 100 times.

What she would do as a programmer to do it fast.
Write a program to write an input 100 times to a file to help all students who are late for school.
May be you can become very famous.

Binary Files
A file can be text or binary. A text file is a sequence of lines. A line is a sequence of characters. The line is terminated by a EOL character. A binary file is any file, a **stream of bytes**.

```
f = open("file1.txt", "rb") # open in binary mode
data = f.read(1)            # read one byte
while (len(data) > 0):
    print(ord(data))        # print ascii value
    data = f.read(1)        # read next byte
f.close()
```

Computers use ASCII values to represent 128 possible characters 1..9, A..Z, a..z etc. ASCII value 1 is 49, A is 65. An ASCII value is represented by 7 bits with 0 and 1.
ord() function returns the ASCII value of character.
chr() function returns the character from ASCII value.



We read chunks of data in a loop. After every read the pointer moves to next byte. At the end of stream we exit the loop.

What will happen if you open an image file from above program using image.jpg as file you open ?
f = open("image.jpg", "rb")

Python **os** module provides many file methods. Study **rename(), remove(), mkdir() , chdir()**. **Press F1 for help.**

File Positions

The **tell()** method tells you the current position within the file. The **seek(offset[, from])** method changes the current position in file. When you open a file, the pointer is at the beginning of

the file. As we read data, it moves towards the end of file. We can move the pointer position back and forward with code.

Create a file.txt with your name,
study and run following program.
Explain what happens.

```
fo = open("file.txt", "r+")
str = fo.read(5); # read 5 characters
print ("String is : ", str)
position = fo.tell(); #check current position
print ("Current position : ", position)
# Reposition pointer at the beginning again
position = fo.seek(0, 0)
position = fo.tell();
print ("Current position : ", position)
str = fo.read(4)  # read 4 characters
print ("String : ", str)
str = fo.read(2)  # read 2 characters
position = fo.tell();
print ("Current position : ", position)
position = fo.seek(0, 1) # move to position 1
str = fo.read(4)
print ("String : ", str)
position = fo.tell();
print ("Current position : ", position)
fo.close()
```

The Alien File
An alien has send file to professor Crow. It has no extension hence professor can't open it with any program. The alien has hidden file extension in the 10th,11th and 12th position of the file. Professor crow will give anyone a raspberry pi, who can write a program to solve this problem.

Take this challenge to write the code.

Write a program to output number of occurrences of an input character in a file. The file can be upto 50MB.
Write a program to **break a file** a large into 5 files.
Write a program to output number of occurrences of a string in a file. **Be very critical in thinking.**

AlgoHack aims to teach computer science for young people,

AlgoHack #8

PROGRAMING OOP, TIME & FILES