***STUDENT INTERNAL MARK CALCULATION***

**CLASS:CSE A S1**

**Group No:9**

**Amala Mary Jomy(16)**

**Eappen Mathew(44)**

**Jini Thomas(56)**

**Guided by,**

**Ms. Sumy Joseph**

**ABSTRACT**

This is a simple project made using python. This project enables the teachers to calculate the internal marks of each student uniquely. The input are the marks of first & second series test and marks secured by the student for his extra activities like assignment and online aptitudes for each subject. The overall result would be the total internal marks secured by a student in each of his semester for each subject. This project makes the work of the teachers simple and can easily get the results.

**1. INTRODUCTION**

**1.1 PROBLEM DEFINITION**

The biggest challenge of a school or college office authority is to manage each and every students mark. In the existing scenario each student’s marks are separately entered by their teachers first in a sheet of paper and then later an office staff copies the mark into a register.  It is extremely tedious to search data from this registers and usually mark registers are kept separate from student registers, moreover there are every chance of entering wrong marks of the student. To overcome all the cons and disadvantages of the existing system, the proposed program is developed to make the entry and the retrieval of student data much easier. This is to calculate the total internal mark of each student uniquely. The teacher will enter the marks secured by each student. Then this entered data will store in a file that can be further used

*.*

**2. SYSTEM STUDY**

**2.1 EXISTING SYSTEM**

In Existing System, there are chances for data loss. All these processes consume more time and require more manual efforts. With the emergence of computers, generations have moved from pencils to pixels, gradually process of calculation are done in computer using excel. Here marks of students are fed into the computer and the calculations are done by computer itself. But the conversions of marks are done by the faculty itself which is a difficult and time consuming task. There is no centralized storage and thereby no facility for quick storage and retrieval of data. There is a chance of ambiguous data collection which will result in wastage of resources. In order to solve the above problem assessment emulator is being proposed.

**2.2 PROPOSED SYSTEM**

This system is done by making use of python. The admin can enter the details of a student such as name , id and marks can be added in the student entry form. It enables the paperless administrative implementation. The wholesome computerization of the data will ensure that the data loss is kept to a minimum and the details can be searched pretty easily.

**3.3FLOWCHART**

**3. SYSTEM DESIGN**

Y

N

G

Read the choice

Start

Read the user name and password

Is the user id and password correct?

Display “invalid user id or password”

Is choice Input

Y

Enter the details of the student

Enter the student id:

N

Enter the student id:

Is choice Display Highest Mark

Is choice Search

H

Is choice Update

Is choice Delete

Is choice Display

Y

Y

Y

Y

Y

Enter the student id:

Display the internal mark of all students

Enter the student id:

N

N

N

N

N

N

N

Y

Y

Enter the mark of the student

Read the Subject

Read the type

Is id: valid?

Enter the student id:

Is chose single editing?

Is chose full editing?

Read the choice

Y

N

Enter the details of the student

Delete the details of the student

Display the internal mark of student

Display the internal mark of student

Do you want to continue?

Stop

G

Is id: valid?

Is id: valid?

Is id: valid?

Is id: valid?

A

A

Display details of the student deleted

Display “choice not found!”

Y

Y

Y

Y

N

N

N

N

**4. CONCLUSION AND FUTURE SCOPE**

**4.1 CONCLUSION**

Program to calculate internal mark of the student was successfully executed.

**4.2LIMITATIONS**

Program can’t run in a system where python is not installed.

**4.3FUTURE SCOPE**

Graphical interface is should be made for make it user friendly. Export this program into mobile application.

**REFERENCES**

1. Python.org
2. Python.docs
3. Stackoverflow.com

**SAMPLE CODE**

import pickle

import os

all\_id=[]

def clear():

os.system('cls')

class student():

sub=['MA','ICS','BEE','EM','PH','SE']

s=['Series 1','Series 2','Assignment','Aptitude']

k=0

def \_\_init\_\_(self):

self.s1marks=[]

self.s2marks=[]

self.asmarks=[]

self.apmarks=[]

self.total=[]

self.total1=0

self.eid=0

self.name='null'

def getdata(self,check=0):

self.eid=input("Enter Id: ")

while self.eid in all\_id and check==0:

self.eid=input("Already Exist, Enter Id: ")

self.name=raw\_input("Enter Name: ")

print "-----Subjects-----"

print "MA: Calcalus"

print "ICS: Introduction to Computer Science"

print "BEE: Basic Electrical Engg."

print "EM: Engg. mechanics"

print "PH: Engg. physics"

print "SE: Introduction to Sustainable engg."

print "------------------"

self.getmark()

self.calculate()

def getdata1(s):

print "-------Type-------"

print "1: Series 1"

print "2: Series 2"

print "3: Assignment"

print "4: Aptitude"

print "------------------"

a=9

while a not in [0,1,2,3] or a<0:

a=input("Enter type:")-1

print "-----------------Subjects-----------------"

print "1: MA: Calcalus"

print "2: ICS: Introduction to Computer Science"

print "3: BEE: Basic Electrical Engg."

print "4: EM: Engg. mechanics"

print "5: PH: Engg. physics"

print "6: SE: Introduction to Sustainable engg."

print "------------------------------------------"

b=9

while b not in [0,1,2,3,4,5] or a<0:

b=input("Enter Sub:")-1

if a==0:

s.s1marks[b]=check1(input("Enter Mark"))

if a==1:

s.s2marks[b]=check1(input("Enter Mark"))

if a==2:

s.asmarks[b]=check2(input("Enter Mark"))

if a==3:

s.apmarks[b]=check2(input("Enter Mark"))

s.calculate()

def calculate(self):

print "lol"

self.total=[]

for i in range(0,len(self.sub)):

a=self.s1marks[i]+self.s2marks[i]+self.asmarks[i]+self.apmarks[i]

self.total.append(a)

self.total1=0

for i in self.total:

self.total1+=i

def getmark(self):

self.s1marks=[]

self.s2marks=[]

self.asmarks=[]

self.apmarks=[]

self.total=[]

for j in self.s:

for i in self.sub:

print '%30s'%("Enter "+i+" mark for "+j+" : "),

if j=="Series 1" or j=="Series 2":

a=check1(input())

else:

a=check2(input())

if j=="Series 1":

self.s1marks.append(a)

elif j=="Series 2":

self.s2marks.append(a)

elif j=="Assignment":

self.asmarks.append(a)

else:

self.apmarks.append(a)

def outdata1(self):

student.k+=1

print student.k,"ID:",self.eid, "|","Name:",self.name

print"---------------------------------------------------------------------------------------------"

def outdata(self):

print"---------------------------------------------------------------------------------------------"

print "ID:",self.eid

print "Name:",self.name

self.outmarks()

def outmarks(self):

#print self.sub,self.s1marks,self.s2marks,self.total

print"---------------------------------------------------------------------------------------------"

print"| Subject | %s | %s | %s | %s | Total | %s "%(self.s[0],self.s[1],self.s[2],self.s[3],"Status")

print"---------------------------------------------------------------------------------------------"

for i in range(0,len(self.sub)):

if self.total[i]>=22.5:

p="PASS"

else:

p="FAIL\*"

#print self.total1

#print "%3s | %4.2f | %3.2f | %3.2f | %3.2f | %3.2f " % (self.sub[i],self.s1marks[i],self.s2marks[i],self.asmarks[i],

# self.apmarks[i],self.total[i])

print "%3s | %4s | %4s | %4s | %4s | %4s | %4s " % (self.sub[i],str(float(self.s1marks[i])),str(float(self.s2marks[i])),str(float(self.asmarks[i])),

str(float(self.apmarks[i])),str(float(self.total[i])),p)

print"--------------------------------------------------------------------------------------------------"

def check1(a):

while a>20 or a<0:

a=input("Enter a mark between 0 and 20 :")

return a

def check2(a):

while a>5 or a<0:

a=input("Enter a mark between 0 and 5 :")

return a

def einput():

get\_all\_id()

f=open('student.dat','ab')

e=student()

e.getdata()

pickle.dump(e,f)

f.close()

def edisplay():

f=open('student.dat','rb')

try:

while True:

e=pickle.load(f)

e.outdata()

except EOFError:

pass

f.close()

def edisplay1():

print"------------------------------------------------------------------------------------------------"

f=open('student.dat','rb')

try:

while True:

e=pickle.load(f)

e.outdata1()

except EOFError:

pass

student.k=0

f.close()

def esearch(r=0):

if r==0:

edisplay1()

f=open('student.dat','rb')

if r==0:

r=input('Enter student ID: ')

try:

while True:

e=pickle.load(f)

if e.eid==r:

print "student Found!"

e.outdata()

break

except EOFError:

print "student not found!!"

f.close()

def high\_mark():

h=0;hid=0

f=open('student.dat','rb')

try:

while True:

e=pickle.load(f)

if e.total1>h:

h=e.total1

hid=e.eid

#print h,hid

except EOFError:

pass

f.close()

esearch(hid)

def edelete():

edisplay1()

f=open('student.dat','rb')

f1=open('temp.dat','wb')

flag=0

id=input('Enter student ID: ')

try:

while True:

e=pickle.load(f)

if e.eid!=id:

pickle.dump(e,f1)

else:

flag=1

except EOFError:

pass

f.close();f1.close()

os.remove('student.dat');os.rename('temp.dat','student.dat')

if flag==1:

print "Details Deleted!"

else:

print "Id not found!"

def eupdate():

get\_all\_id()

flag=0

edisplay1()

f=open('student.dat','rb')

f1=open('temp.dat','wb')

id=input('Enter student ID: ')

clear()

try:

while True:

e=pickle.load(f)

if e.eid==id:

e.getdata(1)

pickle.dump(e,f1)

print "Updated!"

flag=1

else:

pickle.dump(e,f1)

except EOFError:

pass

if flag!=1:

print "Student not found!"

f.close();f1.close()

os.remove('student.dat');os.rename('temp.dat','student.dat')

def eupdate1():

get\_all\_id()

flag=0

f=open('student.dat','rb')

f1=open('temp.dat','wb')

edisplay1()

id=input('Enter student ID: ')

clear()

try:

while True:

e=pickle.load(f)

if e.eid==id:

e.outdata()

e.getdata1()

pickle.dump(e,f1)

print "Updated!"

e.outdata()

flag=1

else:

pickle.dump(e,f1)

except EOFError:

pass

if flag!=1:

print "Student not found!"

f.close();f1.close()

os.remove('student.dat');os.rename('temp.dat','student.dat')

def get\_all\_id():

global all\_id

f=open('student.dat','rb')

try:

while True:

e=pickle.load(f)

all\_id.append(e.eid)

except EOFError:

pass

f.close()

get\_all\_id()

uid=""

pas=""

loop='y'

while loop=='y':

if (uid=="" and pas==""):

clear()

print "-----Login-----"

uid=raw\_input("Username :")

pas=raw\_input("Password :")

print "--------------"

if (uid=="admin" and pas=="admin")or(uid=="allen" and pas=="allen")or(uid=="amit" and pas=="amit"):

clear()

print "-----Menu-----"

print "1: Input"

print "2: Update"

print "3: Delete"

print "4: Display"

print "5: Search"

print "6: Highest Mark"

print "--------------"

n=input("Enter Choice: ")

if n==1:

einput()

elif n==2:

print "-----Update-----"

print "1: Full Editing"

print "2: Single Editing"

print "--------------"

w=input("Enter Choice :")

while (w>2 or w<=0):

w=input("Invalid, Enter again :")

if w==1:

eupdate()

elif w==2:

eupdate1()

elif n==3:

edelete()

elif n==4:

edisplay()

elif n==5:

esearch()

elif n==6:

high\_mark()

else:

print "Choice not found!"

else:

print "Username or Password is incorrect!"

uid=""

pas=""

loop=raw\_input("Do you want to continue? y/n: ")

while True:

if loop=="y" or loop=="n":

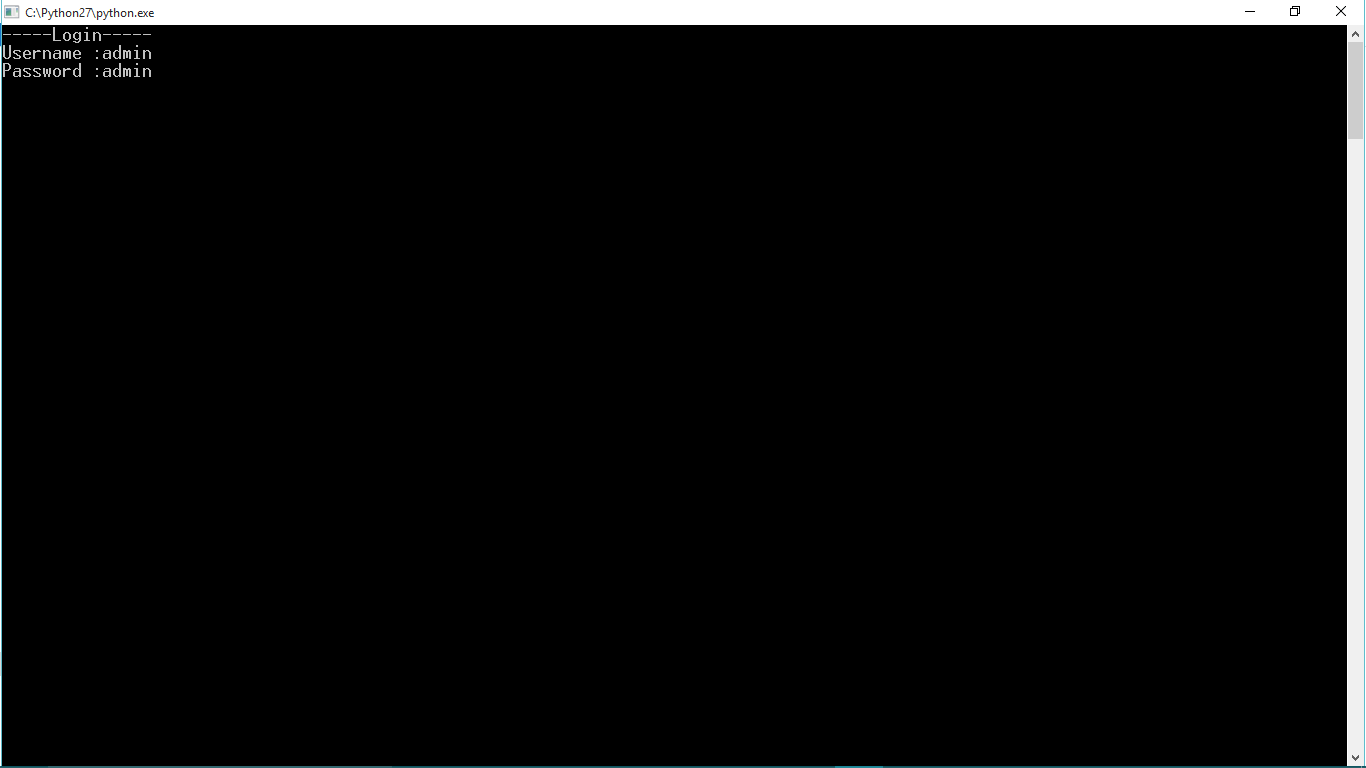
break

else:

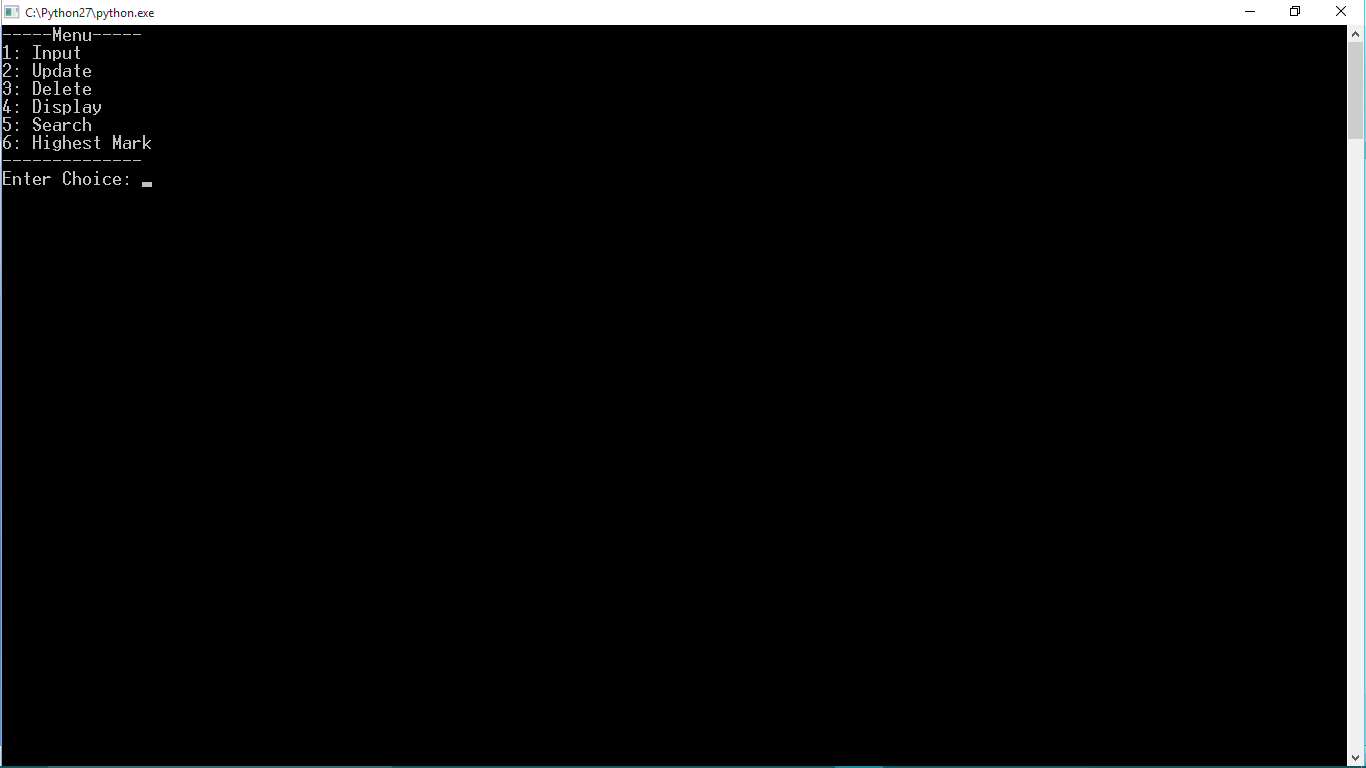
loop=raw\_input("Invaid Input, Do you Continue? y/n: ")

**SCREEN SHOTS**

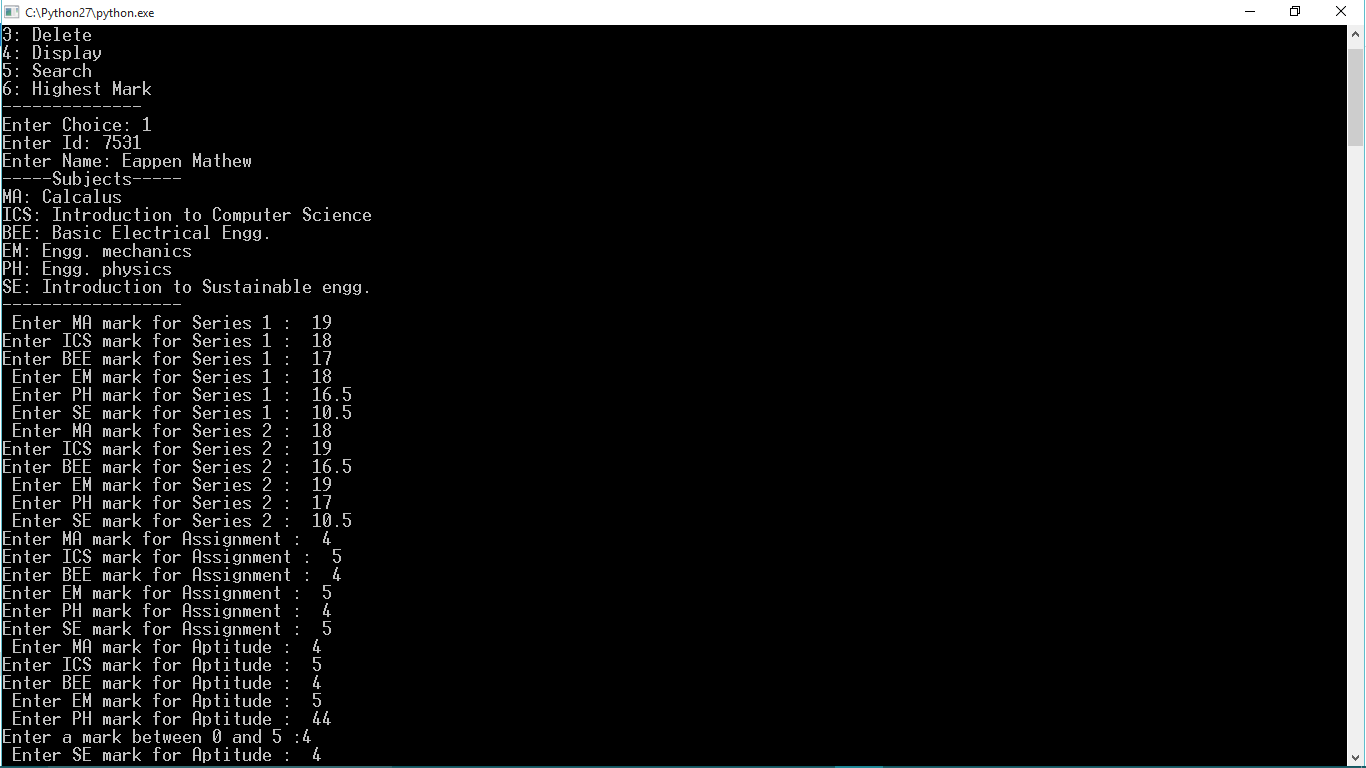
1. Login page



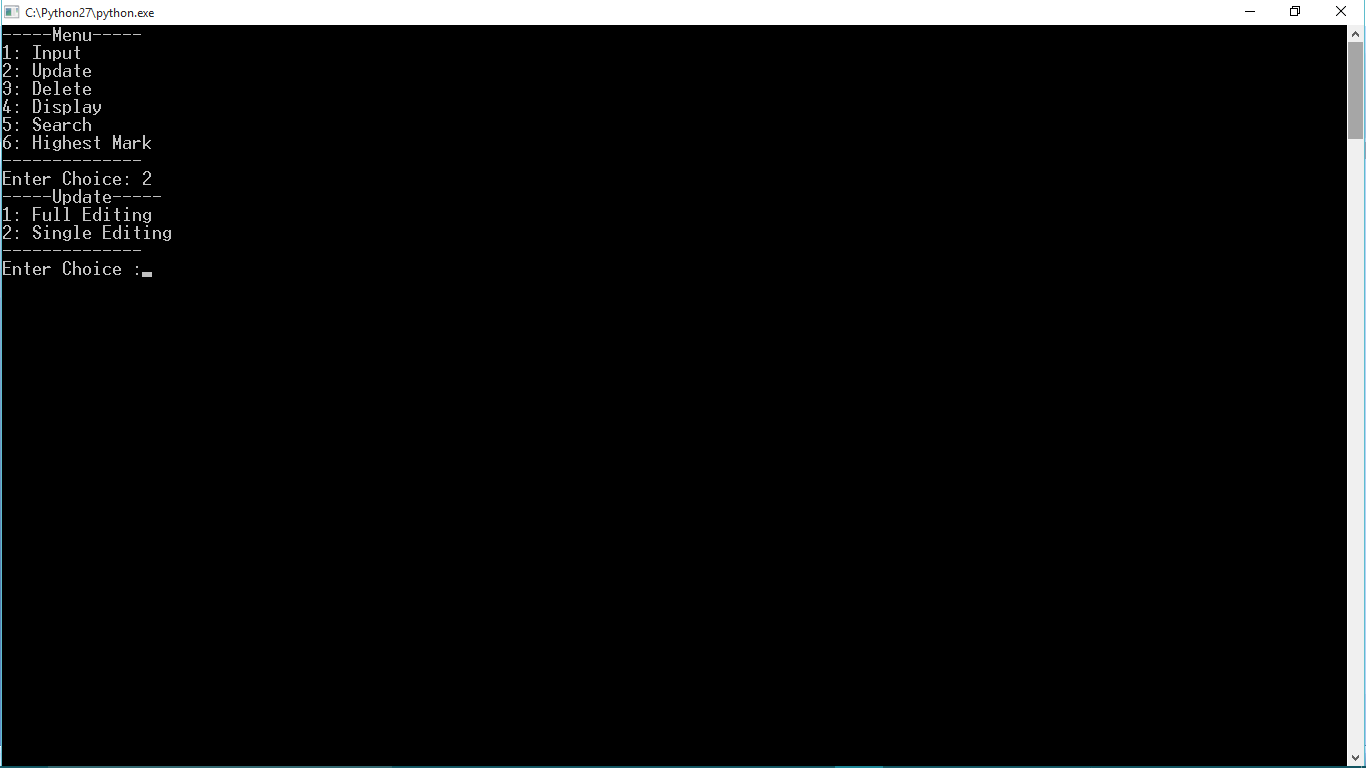
2. Menu



1. Adding a new student

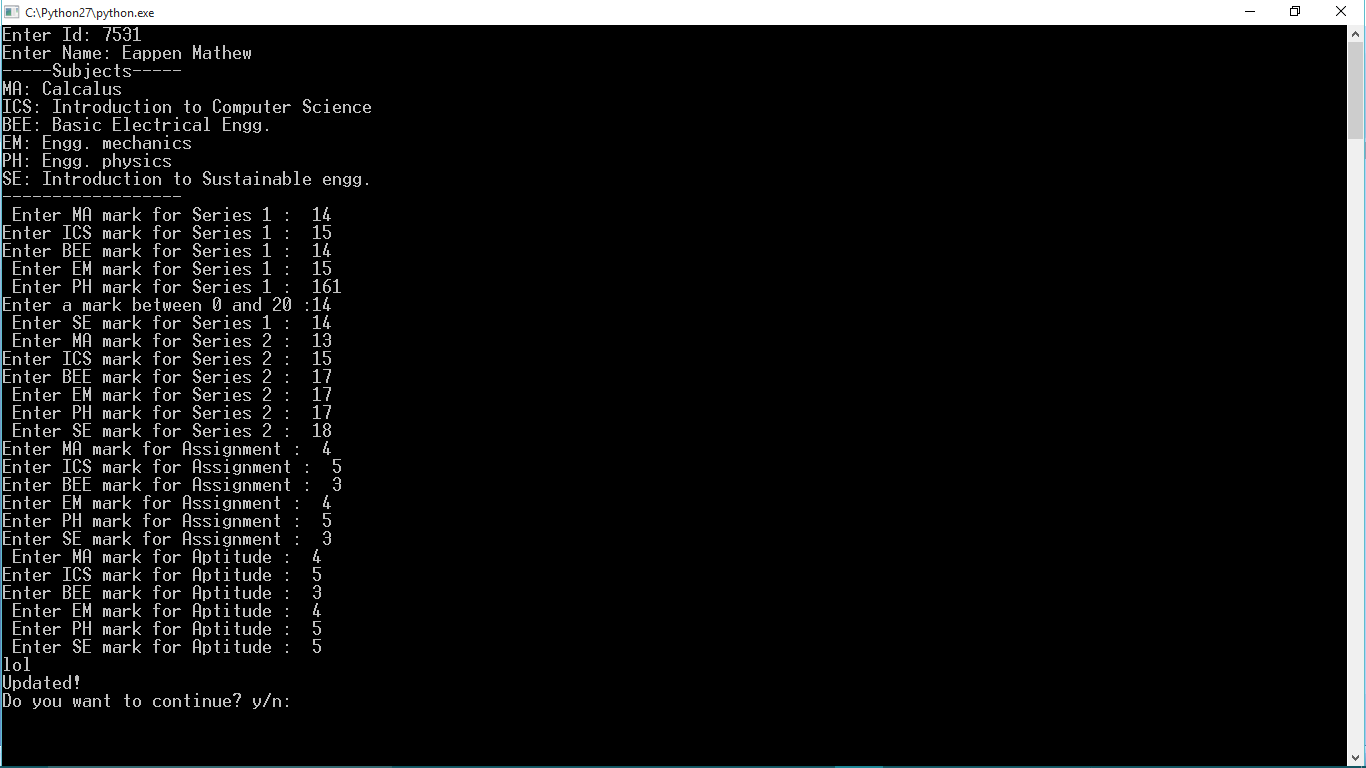


1. Update choice

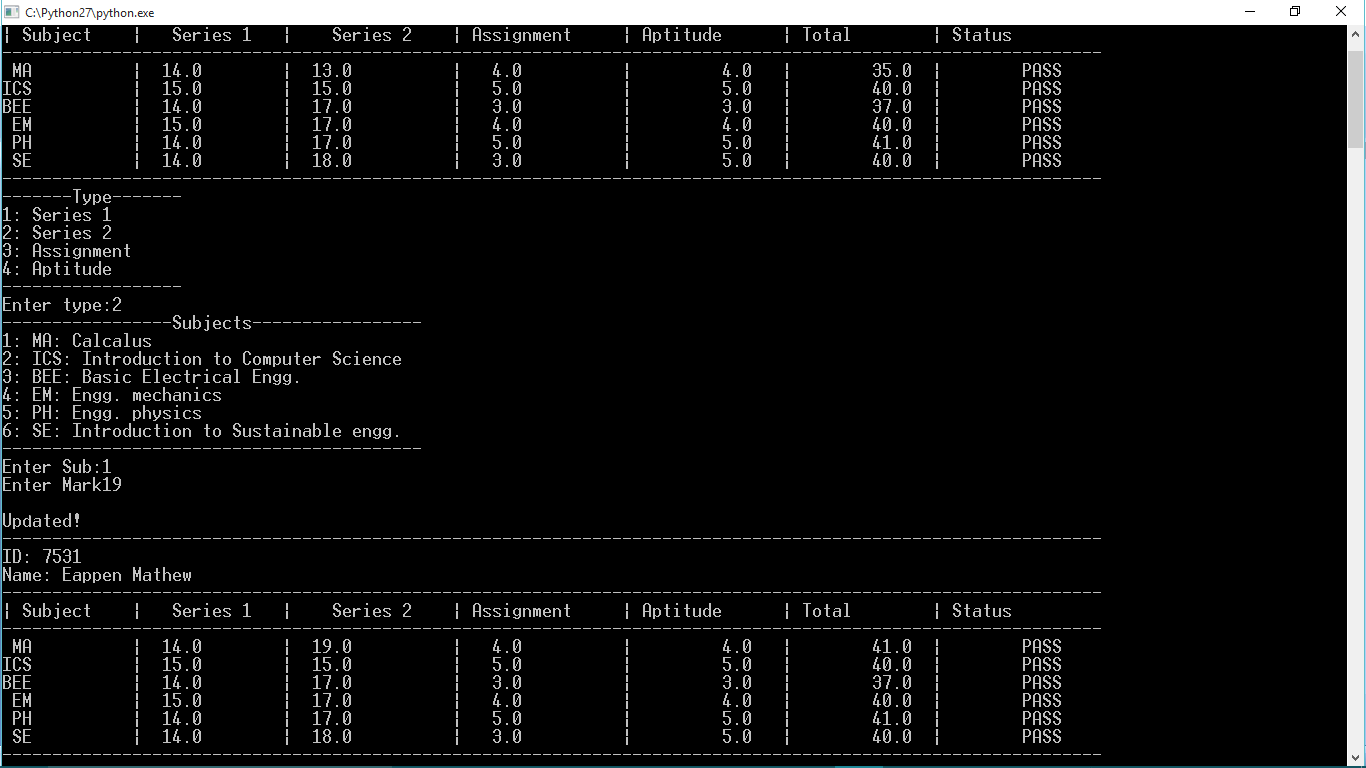


1. Full editing

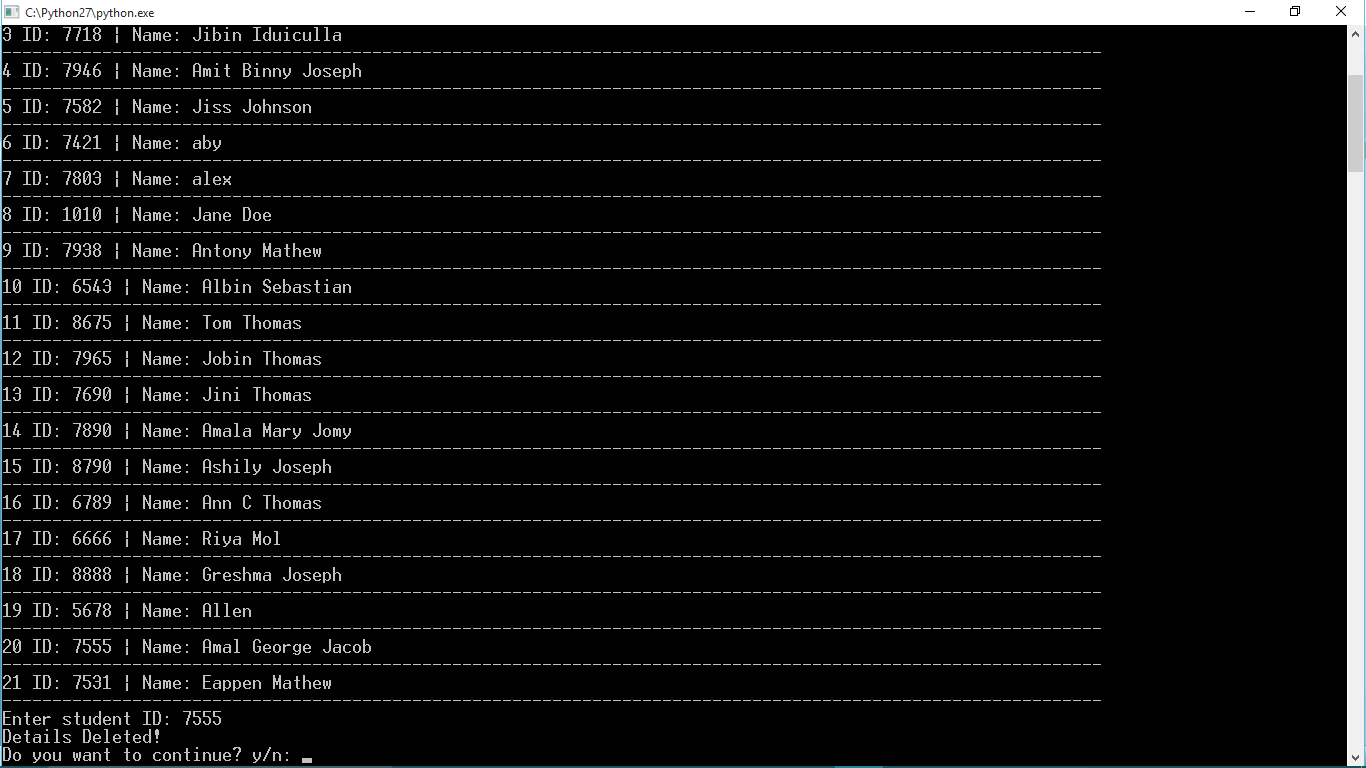




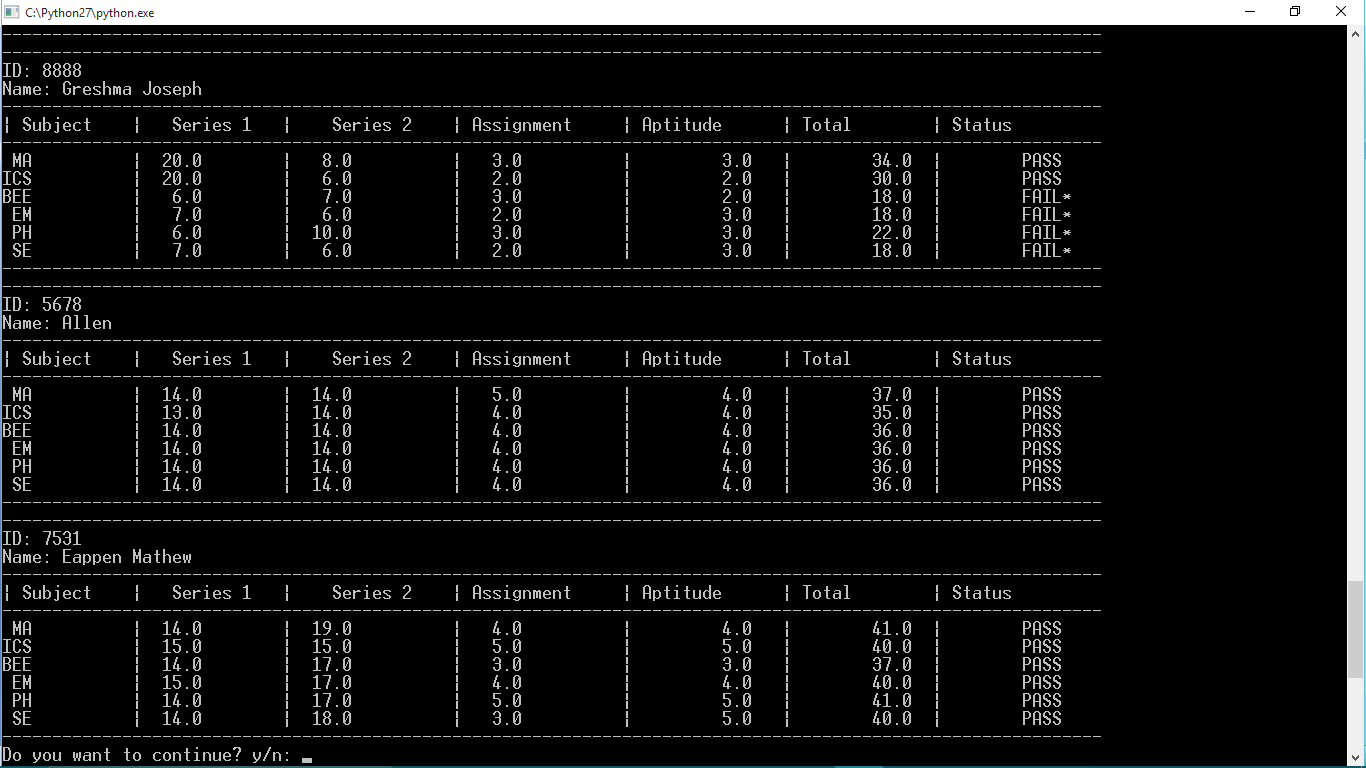
1. Single editing



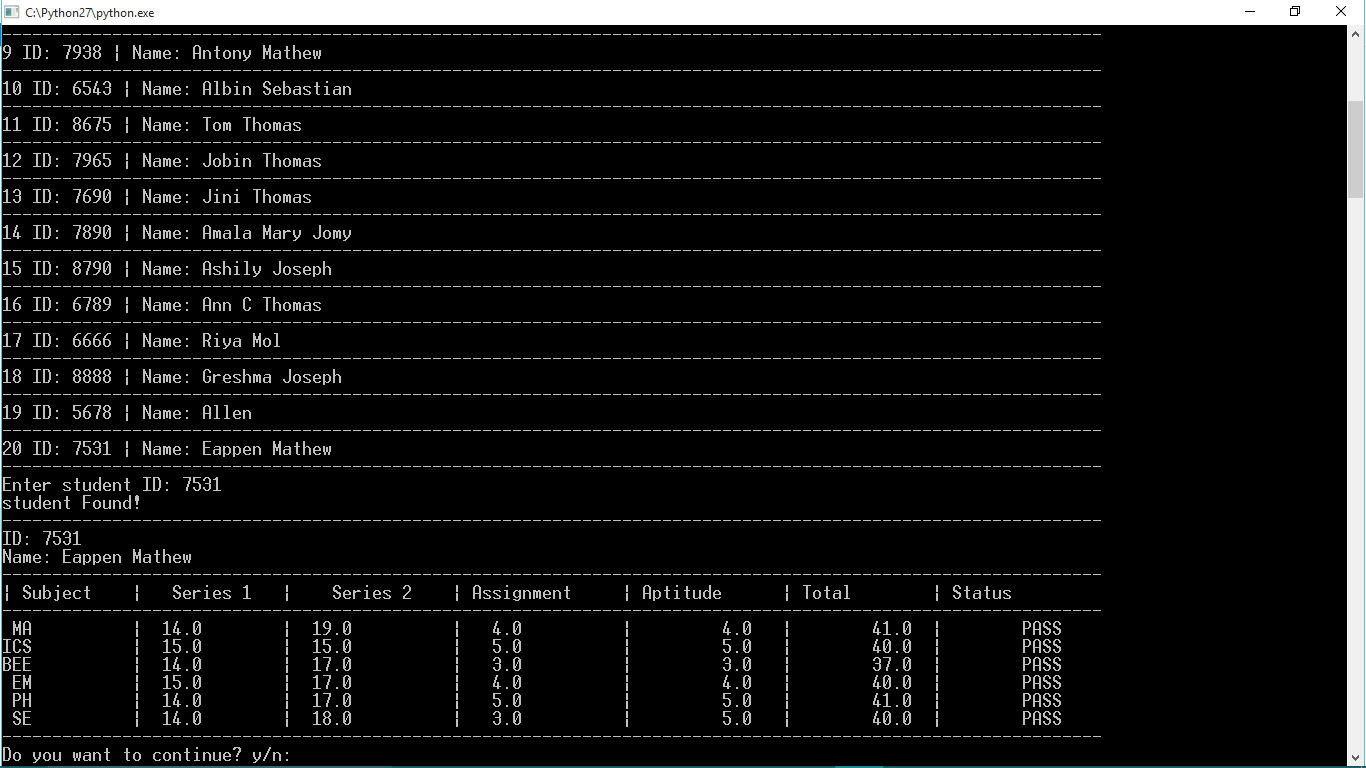
1. Deleting the details of the student



1. Display the details of all student



1. Search the details about student



1. Student who secured highest mark

