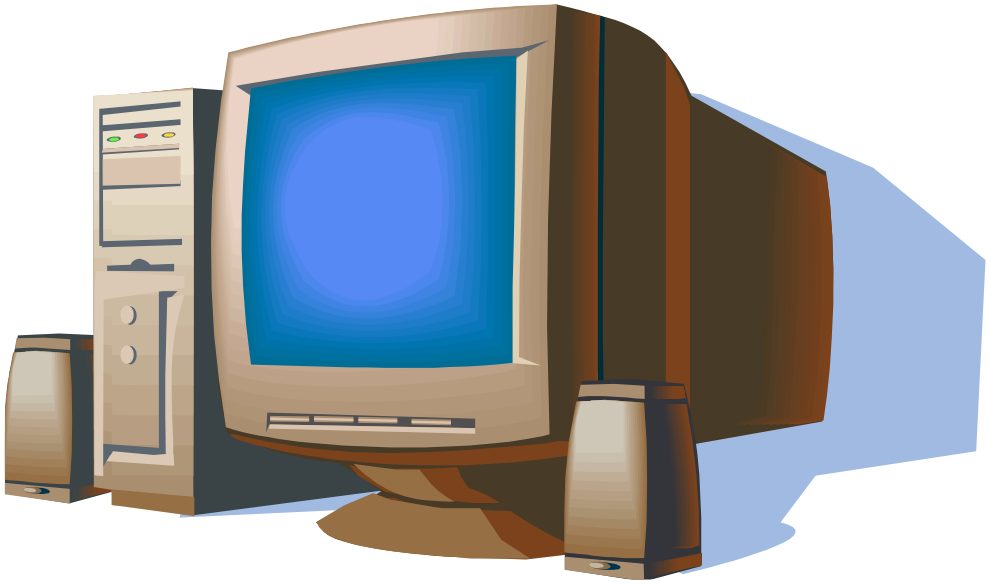
# COMPUTER SCIENCE (PYTHON)

# PROJECT FILE

# ON

# MANAGEMENT SYSTEM




**PROJECT PREPARED BY:**

**SASIN NISAR**

**XII**

**Session: 2018-2019**

**Vagad Pace Global School**

# TABLE OF CONTENTS

# Acknowledgement

The success and the final outcome of this project required a lot of guidance and assistance from Teacher and we are extremely fortunate to have got this all along the completion of our assignment work.

I respect and thank Computer Science teacher (Mr. Arun Sharma) for giving us an opportunity to do this project and providing us all the support and guidance which made us complete the project on time.

Most of all I thank our school management and principal (Mrs. Monika Kapoor), for providing us the facilities and opportunity to do this project.

Lastly, I would like to thank my school mates who have rendered and done this project along with me. Their support made this project fruitful.

CBSE Affiliation No. 1130091 ISO 9001: 2015 Certified

# VAGAD PACE GLOBAL SCHOOL

Residential cum Day School

# *CERTIFICATE*

This is to certify that

_____

Student of Class _____ Roll no_____

Vagad Pace Global School, has successfully completed project on given topic:

_____

_____

In the academic session 2018-2019 under the guidance of Computer Science teacher Mr. Arun Sharma.

Submission Date:

Signature of Subject In-charge: Signature of Principal

School Stamp:

Exam Date: Signature of External Examiner:

# HEADER FILES USED AND THEIR PURPOSE

import pickle: - The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. Pickling and unpickling is alternatively known as “serialization”, “marshalling,” or “flattening”, however, to avoid confusion, the terms used here are “pickling”and “unpickling”.

import os: - This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

Some other methods and functions like class, try and except, EOFerror, while, if, else, elif, def etc.

# BANK MANAGEMENT

```
"""*****************************************************************************
                    MODULES USED IN PROJECT
*****************************************************************************"""
import pickle
import os

"""*****************************************************************************
                    CLASS USED IN PROJECT
*****************************************************************************"""
class account(object):
   def __init__(s):
      s.acno=0
      s.name=""
      s.deposit=0
      s.type=""
   def create_account(s):  #function to get data from user
      name=raw_input("\n\nEnter the name of the account holder: ")
      s.name=name.capitalize()
      type=raw_input("\nEnter type of the account (C/S): ")
      s.type=type.upper()
      s.deposit=input("\nEnter initial amount\n(>=500 for Saving and >=1000
for Current): ")
   def show_account(s):    #function to show data on screen
      print "\nAccount No. :", s.acno
      print "\nAccount holder name: ", s.name
      print "\nType of account", s.type
      print "\nBalance amount: ", s.deposit
   def modify(s):         #function to get new data from user
      print "\nAccount No. : ", s.acno
      s.name=raw_input("\n\nEnter the name of account holder: ")
      type=raw_input("\n\nEnter type of account (C/S): ")
      s.type=type.upper()
      s.deposit=input("\nEnter the amount: ")
   def dep(s,x):          #function to accept amount and add to balance
      s.deposit+=x
   def draw(s,x):         #function to accept amount and subtract from balance
amount
      s.deposit-=x
   def report(s):         #function to show data in tabular format
      print "%-10s"%s.acno,"%-20s"%s.name,"%-10s"%s.type,"%-6s"%s.deposit
   def retacno(s):        #function to return account number
      return s.acno
   def retdeposit(s):     #function to return balance amount
      return s.deposit
```

```
    def rettype(s):       #function to return type of account
        return s.type
"""*****************************************************************************
                FUNCTION TO GENERATE ACCOUNT NUMBER
*****************************************************************************"""
def gen_acno():
    try:
        inFile=open("account2.dat","rb")
        outFile=open("text2.dat","wb")
        n=inFile.read()
        n=int(n)
        while True:
            n+=1
            outFile.write(str(n))
            inFile.close()
            outFile.close()
            os.remove("account2.dat")
            os.rename("text2.dat","account2.dat")
            yield n
    except IOError:
        print "I/O error occured"
"""*****************************************************************************
                FUNCTION TO WRITE RECORD IN BINARY FILE
*****************************************************************************"""
def write_account():
    try:
        ac=account()
        outFile=open("account.dat","ab")
        ch=gen_acno()
        ac.acno=ch.next()
        ac.create_account()
        pickle.dump(ac,outFile)
        outFile.close()
        print "\n\n Account Created Successfully"
        print "\n\n YOUR ACCOUNT NUMBER IS: ",ac.retacno()
    except IOError:
        print "I/O error occured"
"""*****************************************************************************
            FUNCTION TO DISPLAY ACCOUNT DETAILS GIVEN BY USER
*****************************************************************************"""
def display_sp(n):
    flag=0
    try:
        inFile=open("account.dat","rb")
        print "\nBALANCE DETAILS\n"
        while True:
            ac=pickle.load(inFile)
```

```
            if ac.retacno()==n:
                ac.show_account()
                flag=1
        except EOFError:
            inFile.close
            if flag==0:
                print "\n\nAccount number not exist"
        except IOError:
            print "File could not be open !! Press any Key..."
"""*************************************************************************
                    FUNCTION TO MODIFY RECORD OF FILE
*************************************************************************"""
def modify_account(n):
    found=0
    try:
        inFile=open("account.dat","rb")
        outFile=open("temp.dat","wb")
        while True:
            ac=pickle.load(inFile)
            if ac.retacno()==n:
                print 30*"-"
                ac.show_account()
                print 30*"-"
                print "\n\nEnter The New Details of Account"
                ac.modify()
                pickle.dump(ac,outFile)
                print "\n\n\tRecord Updated"
                found=1
            else:
                pickle.dump(ac,outFile)
    except EOFError:
        inFile.close()
        outFile.close()
        if found==0:
            print "\n\nRecord Not Found "
    except IOError:
        print "File could not be open !! Press any Key..."
    os.remove("account.dat")
    os.rename("temp.dat","account.dat")
"""*************************************************************************
                FUNCTION TO DELETE RECORD OF FILE
*************************************************************************"""
def delete_account(n):
    found=0
    try:
        inFile=open("account.dat","rb")
        outFile=open("temp.dat","wb")
```

```
      while True:
         ac=pickle.load(inFile)
         if ac.retacno()==n:
            found=1
            print "\n\n\tRecord Deleted .."
         else:
            pickle.dump(ac,outFile)
   except EOFError:
      inFile.close()
      outFile.close()
      if found==0:
         print "\n\nRecord Not Found"
   except IOError:
      print "File could not be open !! Press any Key..."
   os.remove("account.dat")
   os.rename("temp.dat","account.dat")
"""*************************************************************************
              FUNCTION TO DISPLAY ALL ACCOUNT DETAILS
*************************************************************************"""
def display_all():
   print "\n\n\tACCOUNT HOLDER LIST\n\n"
   print 60*"="
   print "%-10s"%"A/C No.","%-20s"%"Name","%-10s"%"Type","%-
6s"%"Balance"
   print 60*"=","\n"
   try:
      inFile=open("account.dat","rb")
      while True:
         ac=pickle.load(inFile)
         ac.report()
   except EOFError:
      inFile.close()
   except IOError:
      print "File could not be open !! Press any Key..."
"""*************************************************************************
        FUNCTION TO DEPOSIT/WITHDRAW AMOUNT FOR GIVEN ACCOUNT
*************************************************************************"""
def deposit_withdraw(n,option):
   found=0
   try:
      inFile=open("account.dat","rb")
      outFile=open("temp.dat","wb")
      while True:
         ac=pickle.load(inFile)
         if ac.retacno()==n:
            ac.show_account()
            if option==1:
```

```
            print "\n\n\tTO DEPOSIT AMOUNT"
            amt=input("Enter the amount to be deposited: ")
            ac.dep(amt)
         elif option==2:
            print "\n\n\tTO WITHDRAW AMOUNT"
            amt=input("Enter amount to be withdraw: ")
            bal=ac.retdeposit()-amt
            if((bal<500 and ac.rettype()=="S")or(bal<1000 and
ac.rettype()=="C")):
               print "Insufficient balance"
            else:
               ac.draw(amt)
         pickle.dump(ac,outFile)
         found=1
         print "\n\n\tRecord Updated"
      else:
         pickle.dump(ac,outFile)
   except EOFError:
      inFile.close()
      outFile.close()
      if found==0:
         print "\n\nRecord Not Found"
   except IOError:
      print "File could not be open !! Press any Key..."
   os.remove("account.dat")
   os.rename("temp.dat","account.dat")
"""*****************************************************************************
                  INTRODUCTORY FUNCTION
*****************************************************************************"""
def intro():
   print "\n\n\tBANK"
   print "\n\tMANAGEMENT"
   print "\n\n\nMADE BY : Student Name"
   print "\nSCHOOL : Vagad Pace Global School"
"""*****************************************************************************
                  THE MAIN FUNCTION OF PROGRAM
*****************************************************************************"""
intro()

while True:
   print 3*"\n",60*"="
   print """MAIN MENU

   1. New Account
   2. Deposit Amount
   3. Withdraw Amount
   4. Balance Enquiry
```

```
    5. All Account Holder List
    6. Close An Account
    7. Modify An Account
    8. Exit
    """
    try:
       ch=input("Enter Your Choice(1~8): ")
       if ch==1:
          write_account()
       elif ch==2:
          num=input("\n\nEnter Account Number: ")
          deposit_withdraw(num,1)
       elif ch==3:
          num=input("\n\nEnter Account Number: ")
          deposit_withdraw(num,2)
       elif ch==4:
          num=input("\n\nEnter Account Number: ")
          display_sp(num)
       elif ch==5:
          display_all()
       elif ch==6:
          num=input("\n\nEnter Account Number: ")
          delete_account(num)
       elif ch==7:
          num=input("\n\nEnter Account Number: ")
          modify_account(num)
       elif ch==8:
          break
       else:
          print "Input correcr choice...(1-8)"
    except NameError:
       print "Input correct choice...(1-8)"
raw_input("\n\n\n\n\nTHANK YOU\n\nPress any key to exit...")
```

# STUDENT MANAGEMENT

```
"""*****************************************************************************
                    MODULES USED IN PROJECT
*****************************************************************************"""
import pickle
import os
"""*****************************************************************************
                    CLASS USED IN PROJECT
*****************************************************************************"""
class student(object):
   def __init__(self):
      self.grno=0
      self.name=""
      self.f_name=''
      self.age=0
      self.Class=0
      self.section=''
      self.mobile_number=''
   def create_student(self):  #function to get data from user
      name=raw_input("\n\nEnter the name of the Student: ")
      self.name=name.capitalize()
      f_name=raw_input("Enter the father name of the Student: ")
      self.f_name=f_name.capitalize()
      self.age=input("Enter age of Student: ")
      self.Class=input("Enter Class of Student: ")
      section=raw_input("Enter Section of the Student: ")
      self.section=section.upper()
      if len(self.section)>1:
         while len(self.section)>1:
            section=raw_input("Wrong Section, Section must one charecter!!(i.e 'A') Re-
Enter Section again: ")
            self.section=section.upper()
      self.mobile_number=raw_input("Enter the Mobile Number of Student: ")
      if len(self.mobile_number)!=10  or  not self.mobile_number.isdigit() :
         while len(self.mobile_number)!=10  or  not self.mobile_number.isdigit():
            self.mobile_number=raw_input("Wrong mobile number, mobile number must
10 digit and digits!!(i.e '8411944029') Re-Enter Mobile Number Again : ")
   def show_student(self):    #function to show data on screen
      print "\nGR Number of Student. :", self.grno
      print "Name of Student: ", self.name
      print "Father Name of Student: ", self.f_name
      print "Class of Student: ", self.Class
      print "Section of Student: ", self.section
```

```
        print "Age of Student: ", self.age
        print "Mobile Number of Student: ", self.mobile_number
    def modify(self):          #function to get new data from user
        print '1. To Edit Name of Student: '
        print '2. To Edit Father Name of Student: '
        print '3. To Edit Age of Student: '
        print '4. To Edit Class of Student: '
        print '5. To Edit Section of Student: '
        print '6. To Edit Mobile Number of Student: '
        cho=input('Enter your choice: ')
        if cho==1:
            self.name=raw_input('Enter Name of Student: ')
            print 'Student Name updated successfully'
        elif cho==2:
            self.f_name=raw_input('Enter Father Name: ')
            print 'Student Father Name updated successfully'
        elif cho==3:
            self.age=input('Enter age : ')
            print 'Student Father Name updated successfully'
        elif cho==4:
            self.Class=input('Enter class: ')
            print 'Student Class updated successfully'
        elif cho==5:
            self.section=input('Enter Section: ')
            print 'Student section updated successfully'
        elif cho==6:
            self.mobile_number=input('Enter Mobile Number : ')
            print 'Student Mobile Number updated successfully'
        else:
            pass
    def report(self):          #function to show data in tabular format
        print "%-10s"%self.grno,"%-15s"%self.name,"%-15s"%self.f_name,"%-
10s"%self.age,"%-10s"%self.Class,\
              "%-10s"%self.section,"%-15s"%self.mobile_number

    def retgrno(self):         #function to return account number
        return self.grno

    def retmobile_number(self):      #function to return balance amount
        return self.mobile_number
"""*****************************************************************************
                FUNCTION TO GENERATE ACCOUNT NUMBER
*****************************************************************************"""
```

```
def gen_grno():
   try:
      inFile=open("student_gr.dat","rb")
      outFile=open("temp.dat","wb")
      n=inFile.read()
      n=int(n)
      while True:
         n+=1
         outFile.write(str(n))
         inFile.close()
         outFile.close()
         os.remove("student_gr.dat")
         os.rename("temp.dat","student_gr.dat")
         yield n

   except IOError:
      print "I/O error occured"
"""*****************************************************************************
              FUNCTION TO WRITE RECORD IN BINARY FILE
*****************************************************************************"""
def write_student():
   try:
      stu=student()
      outFile=open("student_data.dat","ab")
      ch=gen_grno()
      stu.grno=ch.next()
      stu.create_student()
      pickle.dump(stu,outFile)
      outFile.close()
      print "\n\n Student Record Created Successfully"
      print "\n\n YOUR GR NUMBER IS: ",stu.retgrno()
   except:
      pass
"""*****************************************************************************
           FUNCTION TO DISPLAY ACCOUNT DETAILS GIVEN BY USER
*****************************************************************************"""
def display_student(n):
   flag=0
   try:
      inFile=open("student_data.dat","rb")
      while True:
         st=pickle.load(inFile)
         if st.retgrno()==n:
```

```
            st.show_student()
            flag=1

    except EOFError:
        inFile.close
        if flag==0:
            print "\n\nGR number of Student not exist"

    except IOError:
        print "File could not be open !! Press any Key..."
"""*******************************************************************************
                    FUNCTION TO MODIFY RECORD OF FILE
*******************************************************************************"""
def modify_student(n):
    found=0
    try:
        inFile=open("student_data.dat","rb")
        outFile=open("temp.dat","wb")
        while True:
            st=pickle.load(inFile)
            if st.retgrno()==n:
                print 30*"-"
                st.show_student()
                print 30*"-"
                print "\n\nEnter The New Details of Account"
                st.modify()
                pickle.dump(st,outFile)
                print "\n\n\tRecord Updated"
                found=1
            else:
                pickle.dump(st,outFile)
    except EOFError:
        inFile.close()
        outFile.close()
        if found==0:
            print "\n\nGR number is not valid!!!Record Not Found "
    except IOError:
        print "File could not be open !! Press any Key..."
    os.remove("student_data.dat")
    os.rename("temp.dat", "student_data.dat")
"""*******************************************************************************
                FUNCTION TO DELETE RECORD OF FILE
*******************************************************************************"""
def delete_student(n):
```

```
    found=0
    try:
        inFile=open("student_data.dat","rb")
        outFile=open("temp.dat","wb")
        while True:
            st=pickle.load(inFile)
            if st.retgrno()==n:
                found=1
                print "\n\n\tRecord Deleted .."
            else:
                pickle.dump(st,outFile)
    except EOFError:
        inFile.close()
        outFile.close()
        if found==0:
            print "\n\nRecord Not Found"
    except IOError:
        print "File could not be open !! Press any Key..."
    os.remove("student_data.dat")
    os.rename("temp.dat","student_data.dat")
"""*****************************************************************************
                FUNCTION TO DISPLAY ALL STUDENT DETAILS
*****************************************************************************"""
def display_all_students():
    print "\n\n\tAll STUDENTS LIST\n\n"
    print 90*"="
    print "%-10s"%"GR No.","%-15s"%"Name","%-15s"%"Father Name","%-
10s"%"Age","%-10s"%"Class", \
          "%-10s"%"Section","%-15s"%"Mobile Number"
    print 90*"=","\n"
    try:
        inFile=open("student_data.dat","rb")
        while True:
            st=pickle.load(inFile)
            st.report()
    except EOFError:
        inFile.close()
    except IOError:
        print "File could not be open !! Press any Key..."
"""*****************************************************************************
                    INTRODUCTORY FUNCTION
*****************************************************************************"""
def intro():
    print "\n\n\tSTUDENT"
```

```
    print "\n\tMANAGEMENT"
    print "\n\n\nMADE BY : Student Name"
    print "\nSCHOOL : Vagad Pace Global School"
"""*****************************************************************************
                    THE MAIN FUNCTION OF PROGRAM
*****************************************************************************"""
intro()
while True:
    print 3*"\n",90*"="
    print """MAIN MENU
    1. Create New Student
    2. Display Student
    3. Modify Student
    4. Delete Student
    5. Display All Student
    6. Exit
    """
    try:
        ch=input("Enter Your Choice(1~6): ")
        if ch==1:
            write_student()
        elif ch==2:
            gr_no=input('Enter GR Number of Student: ')
            display_student(gr_no)
        elif ch==3:
            gr_no=input('Enter GR Number of Student: ')
            modify_student(gr_no)
        elif ch==4:
            gr_no=input('Enter GR Number of Student: ')
            delete_student(gr_no)
        elif ch==5:
            display_all_students()
        elif ch==6:
            break
        else:
            print "Input correct choice...(1-8)"
    except NameError:
        print "Input correct choice...(1-8)"
raw_input("\n\n\n\n\nTHANK YOU\n\nPress any key to exit...")
"""*****************************************************************************
                END OF PROJECT
*****************************************************************************"""
```