**COMPUTER SCIENCE**

BINARY DATAFILE BASED PROJECT

**TOPIC- DEPARTMENTAL STORE**

**Name-** Aditya Mittal

**Class-** XII- A

**Group Members:**

Arth Kumar

Raj Mishra

**INDEX**

|  |  |  |
| --- | --- | --- |
| S. NO. | TOPIC | PAGE NO. |
| 1 | Certificate | 1 |
| 2 | Acknowledgement | 2 |
| 3 | Introduction to Python | 3 |
| 4 | User-Defined Functions | 5 |
| 5 | Data Files in Python | 8 |
| 7 | Hardware & Software Specifications | 9 |
| 8 | Binary File & Table | 10 |
| 9 | Python Coding | 11 |
| 10 | Input & Outputs | 23 |
| 11 | Bibliography | 30 |
|  |  |  |
|  |  |  |

**MAHARAJA AGARSAIN PUBLIC SCHOOL**

**ASHOK VIHAR, DELHI**

****

**CERTIFICATE**

**This is to certify that Binary Datafile Based**

**Computer Science project is developed by**

**Aditya Mittal of XII- A**

**under my supervision and guidance.**

**The work done by is original.**

**NITIN GUPTA**

**PGT, COMPUTER SCIENCE**

**DATE- 30/12/2020**

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my Computer Science teacher "Mr. Nitin Gupta" for their able guidance and support in completing my Project.

I would also like to extend my gratitude to the Principal Mam "Mrs. Punam Gupta" for providing me with all the facility that was required.

DATE: Aditya Mittal

30/12/2020 XII- A

Introduction to Python

Python is one of the high-level programming which works in the interpreted mode. It is a high level language with very programmer friendly interface.   
Python works in two different modes: interactive mode and script mode.

Interactive Mode: In interactive mode, the command will be executed on python prompt (>>>) and result will be displayed then and there only. The command will not be saved for future reference.  
Script Mode: In script mode, the python code is written in the editor and it is saved with .py extension and this code can be executed using the shortcut F5 and used for future reference.  
  
**DATA TYPES IN PYTHON**  
Data types define the types of values stored by the variable. The type of values can be with or without decimal, alphabets or special characters. Python support variety of data types. They are:

Numbers Sequence

Integer Float Complex String List Tuples Dictionary

**SELECTION STATEMENT**  
The program is executed depending upon the particular condition and only one part of the expression will be executed on the basis of a particular condition. We can say that, “if” with “else” provides an alternative execution, as there are two possibilities and the condition determines which one gets executed.

**CONDITION**

FALSE TRUE

**STOP**

**WORKING**

**WORKING**

**ITERATION**

We know that computers are often used to automate the repetitive tasks. One of the advantages of using computer to repeatedly perform an identical task is that it is done without making any mistake. Loops are used to repeatedly execute the same code in a program. Python provides two types of looping constructs:

* While Loop
* For Loop

Iteration works in 4 parts:-

1. Starting value
2. Testing Condition/ End Value
3. Working of Loop
4. Increment/ Document

**USER-DEFINED FUNCTIONS**

A function is a set of statements that take inputs, do some specific computation and produce output. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can call the function.

Functions that readily comes with Python are called built-in functions. Python provides built-in functions like print(), etc. but we can also create your own functions. These functions are known as user defines functions.

-> Below are the steps for writing user defined functions in Python:

1.In Python, def keyword is used to declare user defined functions.

2.An indented block of statements follows the function name and arguments which contains the body of the function:

# Python program to

# demonstrate functions

# Declaring a function

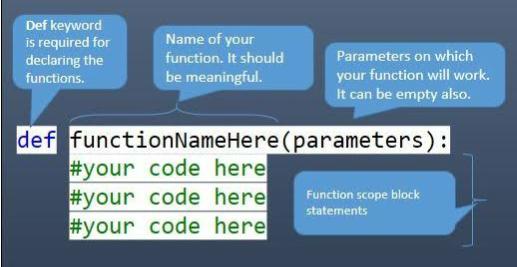
def fun():

print("Inside function")

# Driver's code

# Calling function

fun()



The function may take **arguments**(s) also called **parameters** as input within the opening and closing parentheses, just after the function name followed by a colon.

**Default arguments**

A default argument is a parameter that assumes a default value if a value is not provided in the function call for that argument.The following example illustrates Default arguments.

**Keyword arguments**

The idea is to allow caller to specify argument name with values so that caller does not need to remember order of parameters.

**Function with return value**

A function should also returns a value when it finishes it’s execution. This can be achieved by return statement.

A return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. **The statements after the return statements are not executed**. If the return statement is without any expression, then the special value None is returned.

def fun():

statements

.

.

return [expression]

example...

def add(a, b):

# returning sum of a and b

return a + b

DATA FILES IN PYTHON

A file is a sequence of characters / data which occupies named place on the disk where a sequence of related data is stored.

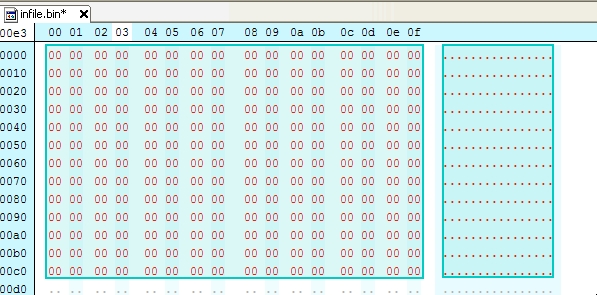
**Data file operation**

* **Opening a file.**
* **Performing operations(read, write)**
* **Closing the file**

A **binary file** is just a file contain information in the same format in which the information is held in the memory.. In binary file there is no delimiter for line.

Python **pickle module** is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

**Advantages of using Pickle Module:**

1. **Recursive objects (objects containing references to themselves):** Pickle keeps track of the objects it has already serialized, so later references to the same object won’t be serialized again. (The marshal module breaks for this.)
2. **Object sharing (references to the same object in different places):** This is similar to self- referencing objects; pickle stores the object once, and ensures that all other references point to the master copy. Shared objects remain shared, which can be very important for mutable objects.
3. **User-defined classes and their instances:** Marshal does not support these at all, but pickle can save and restore class instances transparently. The class definition must be importable and live in the same module as when the object was stored.

HARDWARE AND SOFTWARE SPECIFICATIONS

HARDWARE SPECIFICATIONS:

* CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU. Cores: Single (Dual/Quad Core is recommended)
* RAM: 4 GB (6 GB recommended)
* Graphic Accelerators: nVidia or ATI with support of OpenGL 1.5 or higher.
* Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

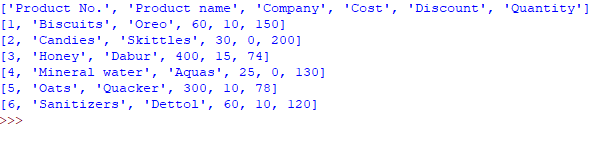
SOFTWARE REQUIREMENTS:

* Windows 7 or higher
* Python 3.9

**BINARY FILE AND TABLE**

**BINARY FILE-** store

**TABLE:**

****

**CODE:**

import pickle

data=[["Product No.","Product name","Company","Cost","Discount","Quantity"]]

ch="y"

while ch=="y":

pno=int(input("Enter product number\_"))

pname=input("Enter product name\_")

comp=input("Enter product Company\_")

cost=int(input("Enter product cost\_"))

disc=int(input("Enter discount percentage\_"))

qty=int(input("Enter product quantity\_"))

list=[pno,pname,comp,cost,disc,qty]

data.append(list)

ch=input("enter 'y' to continue\_")

f=open("store.dat","wb")

pickle.dump(data, f)

f.close()

**PYTHON CODING**

import pickle

import sys

from tabulate import tabulate

print("Datafile created and modified by :")

print(" Arth,Raj,Aditya- 12A")

print('THIS IS A DBMS FOR DEPARTMENTAL STORE')

print('Please Enter the database to be accessed')

column\_headers=['Product no','Product name','Company','Cost','Discount','Quantity']

name=input('Enter File Name-')

fname=name+'.dat'

def addrecord():

ch="y"

print(column\_headers)

while ch=="y":

f=open(fname,"rb+")

stu=pickle.load(f)

pno=int(input("Enter product Number-"))

pname=input("Enter product Name-")

comp=input("Enter product Company-")

cost=int(input("Enter product Cost-"))

disc=int(input("Enter discount Percentage%-"))

qty=int(input("Enter product Quantity-"))

ls=[pno,pname,comp,cost,disc,qty]

stu.append(ls)

f.seek(0)

pickle.dump(stu,f)

f.close()

print("Record Added!")

ch=input("Enter y to add record-")

def show\_file():

f=open(fname,'rb')

g=pickle.load(f)

s1=[]

print('Table in file:')

for i in g:

s1.append(i)

print(tabulate(s1,headers="firstrow",tablefmt="pretty"))

print("Datafile created and modified by :Arth,Raj,Aditya- 12A")

f.close()

def searchbynumber():

xi=int(input("Enter Product Number-"))

f=open(fname,"rb+")

stu=pickle.load(f)

for i in stu:

if i[0]==xi:

print("Record Found!")

print(i)

break

else:

print("No records found for this product number!")

f.close()

def searchbyname():

xi=input("Enter Product Name-")

f=open(fname,"rb+")

stu=pickle.load(f)

for i in stu:

if i[1]==xi:

print("Record Found!")

print(i)

break

else:

print("No records found for this product name!")

f.close()

def searchbycompany():

xi=input("Enter Product Company-")

f=open(fname,"rb+")

stu=pickle.load(f)

for i in stu:

if i[2]==xi:

print("Record Found!")

print(i)

break

else:

print("No records found for this company!")

f.close()

def searchrecord():

ch="y"

while ch=="y":

print("1- To search by Product No.")

print("2- To search by Product Name")

print("3- To search by Company.")

c=input("Enter choice 1/2/3-")

if c=="1":

searchbynumber()

elif c=="2":

searchbyname()

elif c=="3":

searchbycompany()

else:

print("Wrong Input.")

ch=input("Enter y to search again-")

def billing():

ch="y"

sum1=0

sumcp=0

bill=[["Product: ","Quantity:","Price","Discount","Net Amount","Final Cost"]]

f=open(fname,'rb+')

stu=pickle.load(f)

mn=[]

print("Available Today:")

for i in stu:

s=[i[1],"Rs."+str(i[3]),str(i[4])+"%"]

mn.append(s)

print(tabulate(mn,headers="firstrow",tablefmt="pretty"))

print()

sm=0

ds=0

while ch=="y":

buyname=input("Enter Product Name -")

buyquantity=int(input("Enter Quantity You Need-"))

sm=sm+buyquantity

for i in stu:

if i[1]==buyname:

cost=i[3]

b=i[5]

discount=i[4]

ds=ds+discount

bq= b-buyquantity

while bq<0:

print("Max quantity available-",i[5])

buyquantity=int(input("Enter quantity again-"))

bq= b-buyquantity

i[5]=bq

costprice=cost\*buyquantity

finalcost=(cost-(cost)\*(discount/100))\*buyquantity

plist=[buyname,str(buyquantity)+"pcs","Rs."+str(cost),str(discount)+"%","Rs."+str(costprice),"Rs."+str(round(finalcost))]

bill.append(plist)

sum1=sum1+finalcost

sumcp=sumcp+costprice

ch=input ("Enter y to continue shopping-")

break

else:

print("------Item not available-----")

sm=sm-buyquantity

ch=input ("Enter y to continue shopping-")

if sm==0:

sys.exit()

totalsum=round(sum1)

b2=["-------","--------","-------","-------","--------","---------"]

bl=["TOTAL",str(sm)+"pcs","","","Rs."+str(sumcp),"Rs."+str(totalsum)]

bill.append(b2)

bill.append(bl)

print()

print("DBMS DEPARTMENTAL STORE")

print("For enquiry Contact:")

print("9811XXXXXX")

print("dbmsshopping@xyzmail.com")

print()

print(" -----YOUR SHOPPING BILL----- ")

print(tabulate(bill,headers="firstrow",tablefmt="pretty"))

saving=((sumcp-totalsum)/sumcp)\*100

print()

print("Net Total- Rs.",round(sumcp))

print()

print("Total Bill- Rs.",totalsum)

print()

print("Congratulations, you saved- "+str(round(saving))+"%")

print()

print("--------------------------------------------------------")

print("Thanks for shopping with us!")

f.seek(0)

pickle.dump(stu,f)

f.close()

def removerecord():

ch="y"

f=open(fname,"rb+")

stu=pickle.load(f)

while ch=="y":

x=int(input("Enter product no. to be Deleted-"))

for i in stu:

if i[0]==x:

print(i)

stu.remove(i)

print("Record Deleted!")

break

else:

print("Record not found.")

ch=input("Enter y to delete another record-")

f.seek(0)

pickle.dump(stu,f)

f.close()

def modifyrecord():

ch="y"

f=open(fname,"rb+")

stu=pickle.load(f)

while ch=="y":

x2=int(input("Enter product no. to Update-"))

for i in stu:

h=i[0]

if h==x2:

print("Record Found!")

i[1]=input("Enter product Name-")

i[2]=input("Enter Company-")

i[3]=int(input("Enter Cost-"))

i[4]=int(input("Enter Discount-"))

i[5]=int(input("Enter Product Quantity-"))

print("Record Updated!")

break

else:

print("Record not found!")

ch=input("Enter y to update another record-")

f.seek(0)

pickle.dump(stu,f)

f.close()

choice="y"

while choice=="y":

print(" -------MENU-------- ")

print()

print("(1)To add record ")

print("(2)To show table in file")

print("(3)To delete record")

print("(4)To update field")

print("(5)To search record")

print("(6)Shopping")

print()

print("(Entering anything else will exit menu)")

print('-------------------------------------------------------')

choice=int(input('Enter Choice-'))

if choice==1:

addrecord()

elif choice==2:

show\_file()

elif choice==3:

removerecord()

elif choice==4:

modifyrecord()

elif choice==5:

searchrecord()

elif choice==6:

billing()

else :

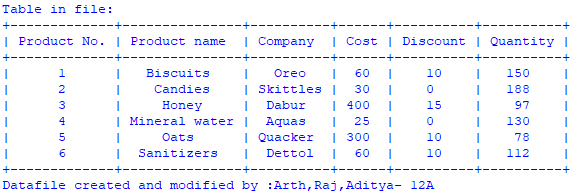
print('WRONG INPUT!')

choice="y"

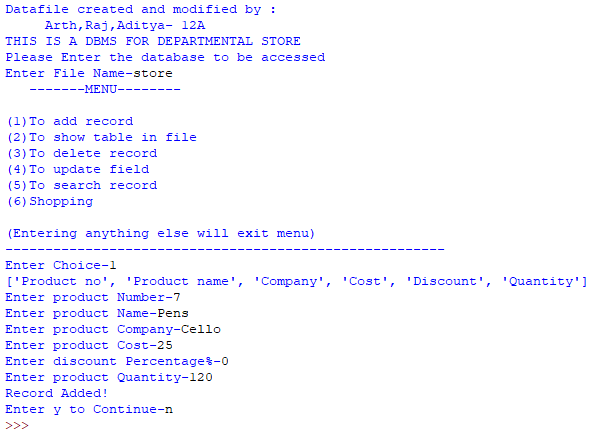
INPUTS AND OUTPUTS

**1.** Adding Records

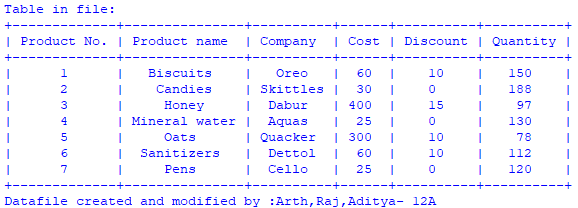
Before Execution Of Program:



Input:

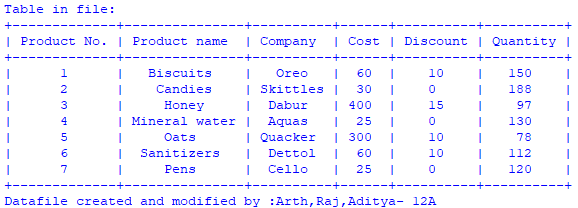


After Execution Of Program:

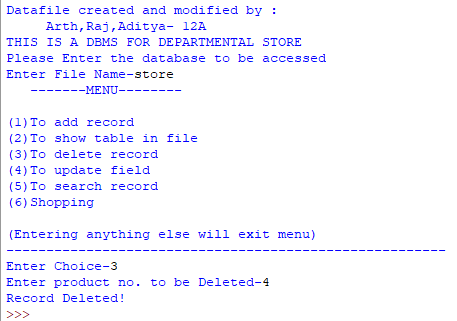


**2.** Deleting Record

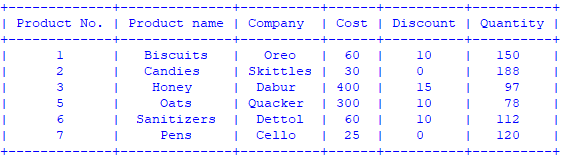
Before Execution Of Program:



Input:

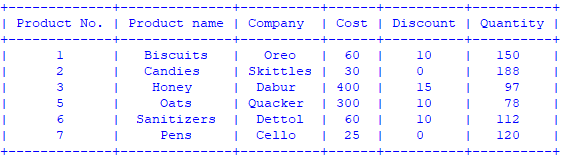


After Execution Of Program:

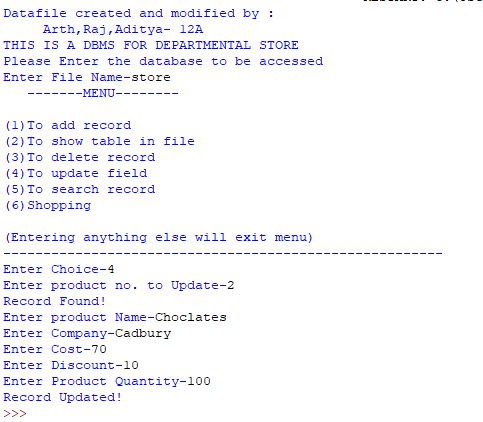


**3.** Updating Record

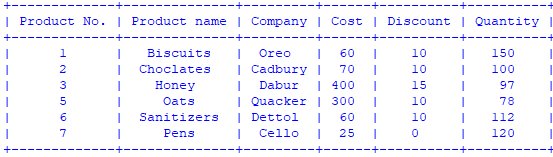
Before Execution Of Program:



Input:

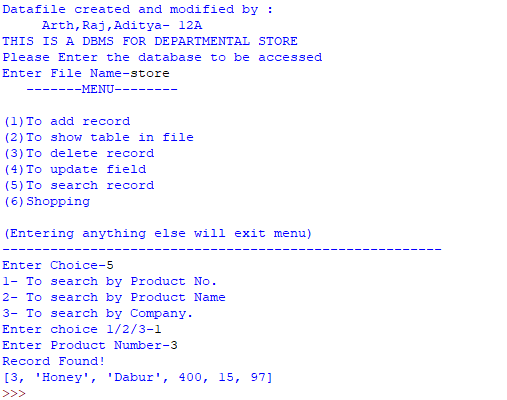


After Execution Of Program:

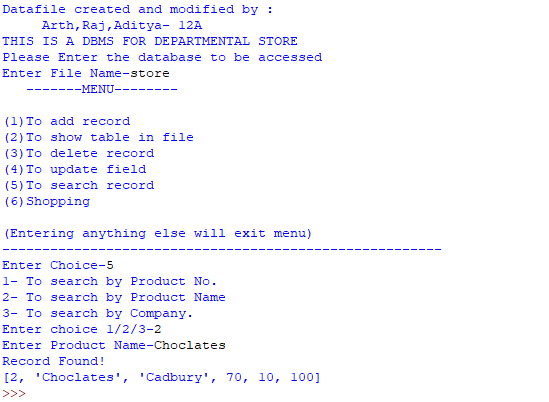


**4.** Searching Record

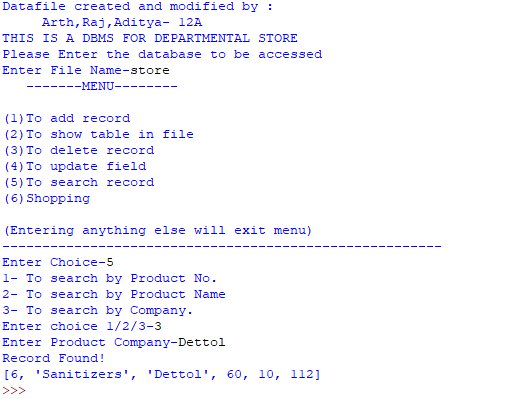
* Searching By Product No.:



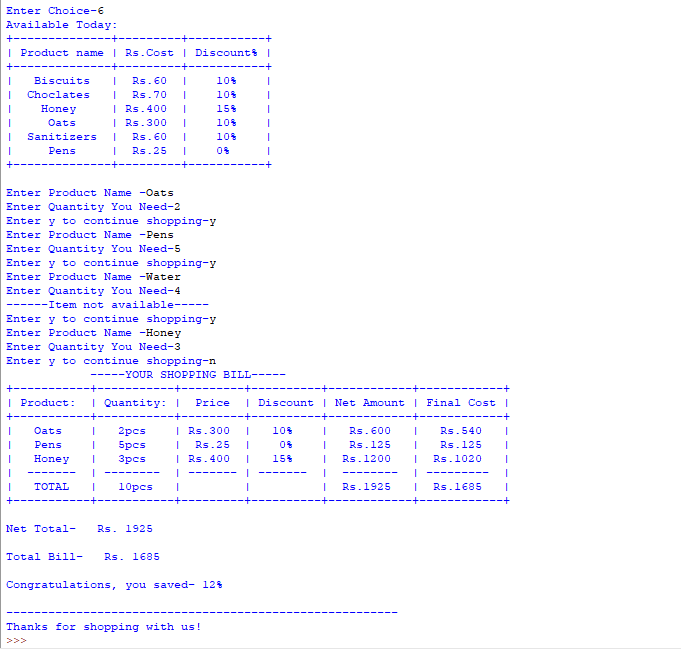
* Searching By Product Name:



* Searching By Product Company:



**5.** Bill Generation



**BIBLIOGRAPHY**

1. Computer science with Python - Class 12 by Sumita Arora.
2. Website: [https://www.w3resource.com](https://www.w3resource.com/)
3. Website: www.geeksforgeeks.org