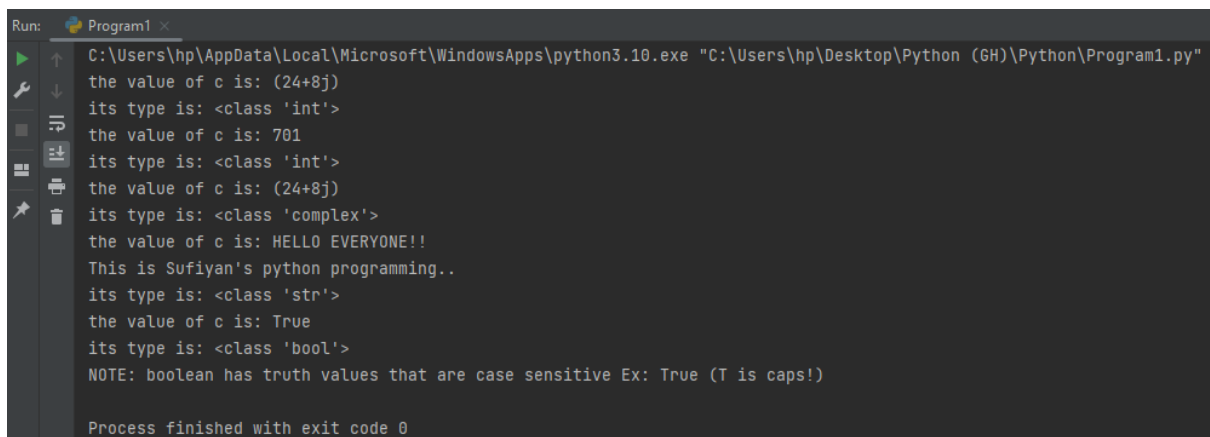


Code:

```
i = 7
c = 24+8j
f = 701
s = 'HELLO EVERYONE!!\nThis is Sufiyan\'s python programming..'
# NOTE: boolean has truth values that are case-sensitive Ex: True (T is caps!)
b = True
print("the value of c is:", c, '\nits type is:', type(i))
print("the value of c is:", f, '\nits type is:', type(f))
print("the value of c is:", c, '\nits type is:', type(c))
print("the value of c is:", s, '\nits type is:', type(s))
print("the value of c is:", b, '\nits type is:', type(b))
print('NOTE: boolean has truth values that are case sensitive Ex: True (T is caps!))')
```

Output:



The screenshot shows a Windows command prompt window titled "Run: Program1". The command executed is "C:\Users\hp\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\hp\Desktop\Python (GH)\Python\Program1.py"". The output of the program is as follows:

```
the value of c is: (24+8j)
its type is: <class 'int'>
the value of c is: 701
its type is: <class 'int'>
the value of c is: (24+8j)
its type is: <class 'complex'>
the value of c is: HELLO EVERYONE!!
This is Sufiyan's python programming..
its type is: <class 'str'>
the value of c is: True
its type is: <class 'bool'>
NOTE: boolean has truth values that are case sensitive Ex: True (T is caps!)

Process finished with exit code 0
```

Code:

Classes, Objects & Inheritance

person is a base class

class Person:

def __init__(self, n, a):

self.name = n

self.age = a

employee is the class derived from person using single inheritance

class Employee(Person):

def __init__(self, n, a, d, s):

Person.__init__(self, n, a)

self.designation = d

self.salary = s

def show(self):

print("Employee Details: ")

print(" Name: ", self.name, "\n Age:", self.age, "\n Designation:", self.designation, "\n Salary:", self.salary)

class Student:

def __init__(self, id_, rno):

self.studentId = id_

self.room_no = rno

resident is a class derived from person and student using multiple inheritance

class Resident(Person, Student):

def __init__(self, n, a, id_, rno):

Person.__init__(self, n, a)

Student.__init__(self, id_, rno)

def show(self):

print("Resident Details:")

print(" Name:", self.name, "\n Age: ", self.age, "\n Id:", self.studentId, "\n Room no.:", self.room_no)

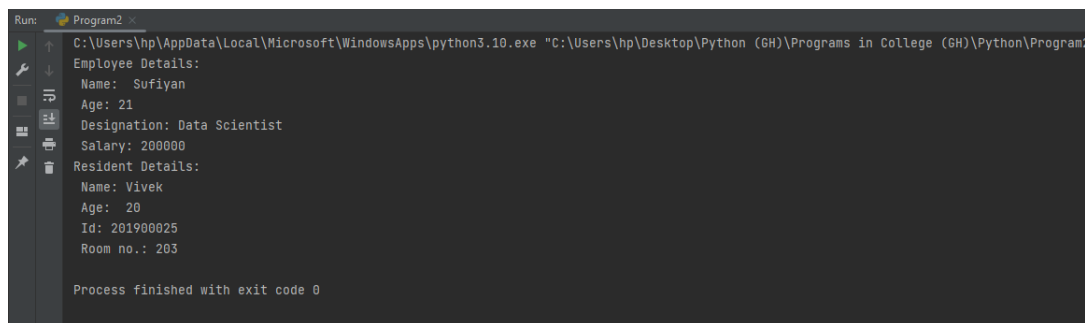
Creating objects of employee and resident classes

```
e1 = Employee("Sufiyan", 21, "Data Scientist", 200000)
r1 = Resident("Vivek", 20, 201900025, 203)
e1.show()
r1.show()
```

Exception Handling

```
try:
    number1, number2 = eval(input("Enter two numbers separated by a comma:"))
    result = number1 / number2
    print("Result is",result)
except ZeroDivisionError:
    print("Division by Zero")
except SyntaxError:
    print("A comma may be Missing in the Input")
except RuntimeError:
    print("May be Meaningless ")
except:
    print("Something Wrong in the Input")
else:
    print("No Exceptions")
finally:
    print("Finally Clause is Executed ")
```

Outputs:



```
Run: Program2
C:\Users\hp\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\hp\Desktop\Python (GH)\Programs in College (GH)\Python\Program2
Employee Details:
Name: Sufiyan
Age: 21
Designation: Data Scientist
Salary: 200000
Resident Details:
Name: Vivek
Age: 20
Id: 201900025
Room no.: 203

Process finished with exit code 0
```

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c
:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program2b.py"
Enter two numbers separated by a comma: 9 1
A comma may be Missing in the Input
Finally Clause is Executed
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c
:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program2b.py"
Enter two numbers separated by a comma: 8, 0
Division by Zero
Finally Clause is Executed
```

Code:

#Code1

```
file1 = open("myfile.txt", "w")
L = ["This is Calcutta \n", "This is China \n", "This is France"]
file1.writelines(L)
file1.close()
# Append-adds at last
file1 = open("myfile.txt", "a") # append mode
file1.write("Today \n")
file1.close()
file1 = open("myfile.txt", "r")
print("Output of Readlines after appending")
print(file1.read())
print()
file1.close()
# Write-Overwrites
file1 = open("myfile.txt", "w")
# write mode
file1.write("Tomorrow \n")
file1.close()
file1 = open("myfile.txt", "r")
print("Output of Readlines after writing")
print(file1.read())
print()
file1.close()
```

#Code2

```
file = open("myfile.txt", "r")
print("The contents of the file: ")
print(file.read())
file.close()
file = open("myfile.txt", "r")
lines = 0
words = 0
symbols = 0
for line in file:
    lines += 1
    words += len(line.split())
    symbols += len(line.strip('\n'))
print("\nDetails")
print("Lines:", lines)
print("Words:", words)
print("Symbols:", symbols)
```

```
file.close()
```

#Code3

import OS module

```
import os
```

Get the list of all files and directories

```
path = "C://Users//USER//Desktop//Sufi Folder"
```

```
dir_list = os.listdir(path)
```

```
print("Files and directories in '", path, "' :")
```

prints all files

```
print(dir_list)
```

Outputs:

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c
:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program3a.py"
Output of Readlines after appending
This is Calcutta
This is China
This is FranceToday

Output of Readlines after writing
Tomorrow
```

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c
:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program3b.py"
The contents of the file:
Tomorrow

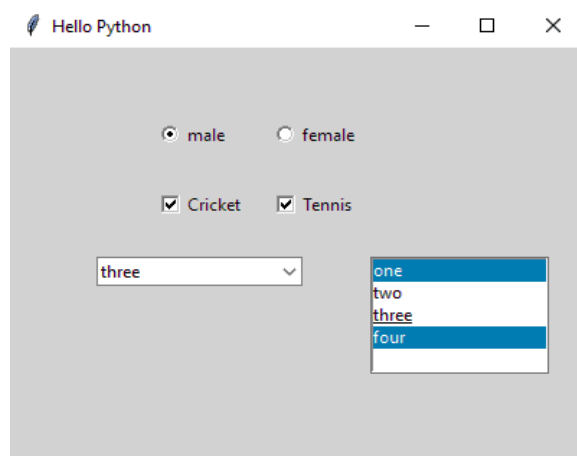
Details
Lines: 1
Words: 1
Symbols: 9
```

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c
:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program3c.py"
Files and directories in ' C://Users//USER//Desktop//Sufi Folder ' :
['Android Development', 'Books', 'College Works', 'Other', 'Programming Languages (GH)', 'Software & Setup Files']
```

Code:

```
from tkinter import *
from tkinter.ttk import Combobox
window=Tk()
var = StringVar()
var.set("one")
data=("one", "two", "three", "four")
cb=Combobox(window, values=data)
cb.place(x=60, y=150)
lb=Listbox(window, height=5, selectmode='multiple')
for num in data: lb.insert(END,num)
lb.place(x=250, y=150)
v0=IntVar()
v0.set(1)
r1=Radiobutton(window, text="male", variable=v0,value=1)
r2=Radiobutton(window, text="female", variable=v0,value=2)
r1.place(x=100,y=50)
r2.place(x=180, y=50)
v1 = IntVar()
v2 = IntVar()
C1 = Checkbutton(window, text = "Cricket", variable = v1)
C2 = Checkbutton(window, text = "Tennis", variable = v2)
C1.place(x=100, y=100)
C2.place(x=180, y=100)
window.title('Hello Python')
window.geometry("400x300+10+10")
window.mainloop()
```

Output:



Code:

```
#Code1
def isEmpty(stk):
# checks whether the stack is empty or not
    if stk==[]:
        return True
    else:
        return False
def Push(stk,item): # Allow additions to the stack
    stk.append(item)
    top = len(stk)-1
def Pop(stk):
    if isEmpty(stk): # verifies whether the stack is empty or not
        print("Underflow")
    else:
# Allow deletions from the stack
        item=stk.pop()
        if len(stk)==0:
            top = None
        else:
            top=len(stk)
        print("Popped item is "+str(item))
def Display(stk):
    if isEmpty(stk):
        print("Stack is empty")
    else:
        top=len(stk)-1
        print("Elements in the stack are: ")
        for i in range(top,-1,-1):
            print (str(stk[i]))
# executable code
stk=[]
top=None
Push(stk,1)
Push(stk,2)
Push(stk,3)
Push(stk,4)
print("Stack before popping an element:")
Display(stk)
Pop(stk)
print("\nStack after popping an element:")
Display(stk)
```

#Code2

#Adding elements to queue at the rear end

def enqueue(data):

queue.insert(0,data)

#Removing the front element from the queue

def dequeue():

if len(queue)>0:

return queue.pop()

return ("Queue Empty!")

#To display the elements of the queue

def display():

print("Elements on queue are:");

for i in range(len(queue)):

print(queue[i])

executable code

queue=[]

enqueue(5)

enqueue(6)

enqueue(9)

enqueue(5)

enqueue(3)

print("Queue before popping element: ")

display()

print("\nPopped Element is: "+str(dequeue()))

print("Queue after popping element: ")

display()

#Code3

importing module

import collections

initialising a deque() of arbitrary length

linked_lst = collections.deque()

filling deque() with elements

linked_lst.append('first')

linked_lst.append('second')

linked_lst.append('third')

print("Elements in the linked_list:")

print(linked_lst)

adding element at an arbitrary position

linked_lst.insert(1, 'fourth')


```
print("\nElements in the linked_list:")
print(linked_list)
```

```
# deleting the last element
linked_list.pop()
```

```
print("\nElements in the linked_list:")
print(linked_list)
```

```
# removing a specific element
linked_list.remove('fourth')
```

```
print("\nElements in the linked_list:")
print(linked_list)
```

Outputs:

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program5.py"
Stack before popping an element:
Elements in the stack are:
4
3
2
1
Popped item is 4

Stack after popping an element:
Elements in the stack are:
3
2
1
```

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program5b.py"
Queue before popping element:
Elements on queue are:
3
5
9
6
5

Popped Element is: 5
Queue after popping element:
Elements on queue are:
3
5
9
6
```

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program5c.py"
Elements in the linked_list:
deque(['first', 'second', 'third'])

Elements in the linked_list:
deque(['first', 'fourth', 'second', 'third'])

Elements in the linked_list:
deque(['first', 'fourth', 'second'])

Elements in the linked_list:
deque(['first', 'second'])
```

Code:

#Code1

```
import sqlite3
db=sqlite3.connect('database.db')
try:
    cur=db.cursor()
    cur.execute("""CREATE TABLE book (
    BookID INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT (20) NOT NULL,
    author TEXT (30),
    publisher TEXT (20));""")
    print ('Table Created Successfully')
except:
    print ('Error in Operation')
    db.rollback()
    db.close()
```

#Code2

```
import sqlite3
db=sqlite3.connect('database.db')
qry="insert into book (title, author, publisher) values('Internet Programming', 'Arya More', 'Sandip Publications'),('Machine Learning', 'Sufiyan Chougule', 'Arif Publications');"
try:
    cur=db.cursor()
    cur.execute(qry)
    db.commit()
    print ("Two Records Added Successfully")
except:
    print ("Error in operation")
    db.rollback()
    db.close()
```

#Code3

```
import sqlite3
db=sqlite3.connect('database.db')
sql="SELECT * from book;"
cur=db.cursor()
cur.execute(sql)
while True:
    record=cur.fetchone()
    if record==None:
        break
    print (record)
```

```
db.close()
```

#Code4

```
import sqlite3
```

```
db=sqlite3.connect('database.db')
```

```
qry="update book set title = 'Microprocessors' where author = 'Arya More'"
```

```
try:
```

```
    cur=db.cursor()
```

```
    cur.execute(qry)
```

```
    db.commit()
```

```
    print("Record Updated Successfully")
```

```
except:
```

```
    print("Error in Operation")
```

```
    db.rollback()
```

```
    db.close()
```

#Code5

```
import sqlite3
```

```
db = sqlite3.connect( 'database.db')
```

```
qry= " DELETE from book where publisher='Sandip Publications'"
```

```
try:
```

```
    cur=db.cursor()
```

```
    cur.execute(qry)
```

```
    db.commit()
```

```
    print(" Record Deleted Successfully")
```

```
except:
```

```
    print(" Error in Operation")
```

```
    db.rollback()
```

```
    db.close()
```

Output:

```
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6a.py"
Table Created Successfully
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6b.py"
Two Records Added Successfully
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6c.py"
(1, 'Internet Programming', 'Arya More', 'Sandip Publications')
(2, 'Machine Learning', 'Sufiyan Chougule', 'Arif Publications')
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6d.py"
Record Updated Successfully
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6c.py"
(1, 'Microprocessors', 'Arya More', 'Sandip Publications')
(2, 'Machine Learning', 'Sufiyan Chougule', 'Arif Publications')
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6e.py"
Record Deleted Successfully
PS C:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python> python -u "c:\Users\USER\Desktop\Sufi Folder\College Works\Important\Practicals & Projects (GH)\Programs in College\Python\Program6c.py"
(2, 'Machine Learning', 'Sufiyan Chougule', 'Arif Publications')
```