

## **1. Write a Python program to Show Multilevel Inheritance.**

### **Program Code :**

```
class Mca:

    def course1(self):

        print("PG Course: MCA")

class Bca(Mca):

    def course2(self):

        print(" UG Course:BCA")

class Bsc(Bca):

    def course3(self):

        print("Second UG Course: BSc")


c = Bsc()

c.course1()

c.course2()

c.course3()
```

### **OUTPUT :**

PG Course: MCA

UG Course:BCA

Second UG Course: BSc

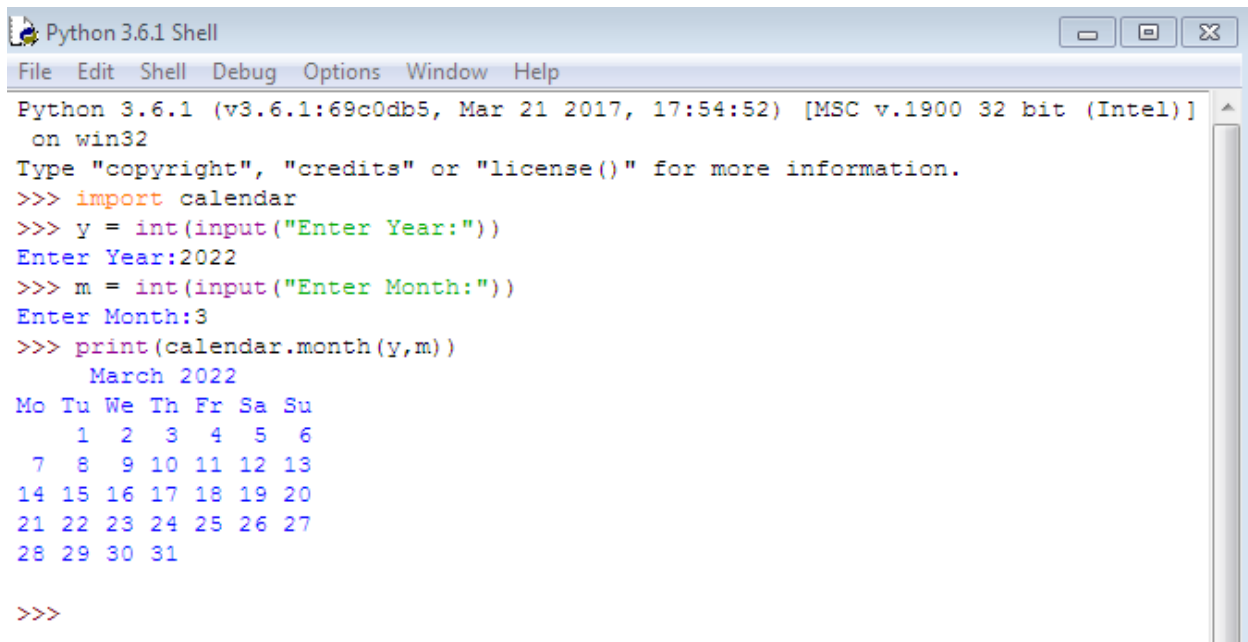
## 2. Write a Python program to display Calendar by providing the Year entered by user.

### CODE :

```
>>> import calendar
>>> y = int(input("Enter year: "))
Enter year: 2022
>>> m = int(input("Enter Month:"))
Enter Month:3
>>> print(calendar.month(y,m))
```

### OUTPUT:

```
March 2022
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

A screenshot of a Python 3.6.1 Shell window. The window has a title bar that says "Python 3.6.1 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area of the window shows the following text:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> import calendar
>>> y = int(input("Enter Year:"))
Enter Year:2022
>>> m = int(input("Enter Month:"))
Enter Month:3
>>> print(calendar.month(y,m))
March 2022
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

>>>
```

### **3. Write a Program in Python to Show Method Overriding.**

#### **CODE :**

```
class Demo1:
    def Dispaly(self):
        print("I am from Parent Class")

class Demo2(Demo1):
    def Display(self):
        print("I am from Child Class")

a = Demo2()
a.Display()
```

#### **OUTPUT:**

I am from Child Class

#### **4. Write a Python Program that implement Thread.**

##### **CODE:**

```
from time import sleep
from threading import Thread
```

```
class Hello(Thread):
    def run(self):
        for i in range(100):
            print("Hello")
            sleep(1)
```

```
class Hi(Thread):
    def run(self):
        for i in range(100):
            print("Hi")
            sleep(1)
```

```
t1 = Hello()
t2 = Hi()
```

```
t1.start()
sleep(0.2)
t2.start()
```

```
t1.join()
t2.join()
print("Good");
```

##### **OUTPUT:**

```
== RESTART: C:\Users\u\AppData\Local\Programs\Python\Python36-32\Thread1.py ==
Hello
Hi
Hello
Hi
Hello
Hi
Hello
```

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

Hello

Hi

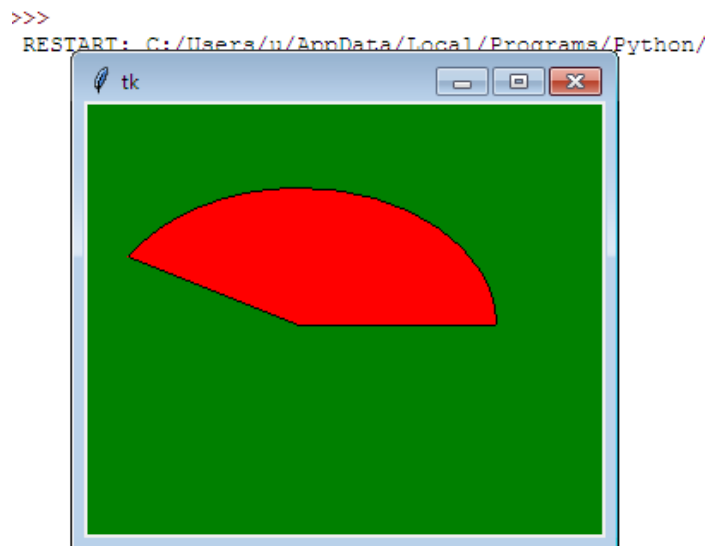
Hello

**5. Write a Python Program to draw the “filled arc” using TKinter Module.**

**CODE :**

```
import tkinter
top = tkinter.Tk()
C = tkinter.Canvas(top,bg="green",height=250,width=300)
coord = 10,50,240,210
arc = C.create_arc(coord,start=0,extent=150, fill="red")
C.pack()
top.mainloop()
```

**OUTPUT:**



## **6. Write Python program to create Menus and Submenus using Tkinter.**

```
from tkinter import *

def donothing():
    filewin = Toplevel(root)
    button = Button(filewin, text="Do nothing button")
    button.pack()

root = Tk()
menubar = Menu(root)
filemenu = Menu(menubar, tearoff=0)
filemenu.add_command(label="New", command=donothing)
filemenu.add_command(label="Open", command=donothing)
filemenu.add_command(label="Save", command=donothing)
filemenu.add_command(label="Save as...", command=donothing)
filemenu.add_command(label="Close", command=donothing)

filemenu.add_separator()

filemenu.add_command(label="Exit", command=root.quit)
menubar.add_cascade(label="File", menu=filemenu)
editmenu = Menu(menubar, tearoff=0)
editmenu.add_command(label="Undo", command=donothing)

editmenu.add_separator()

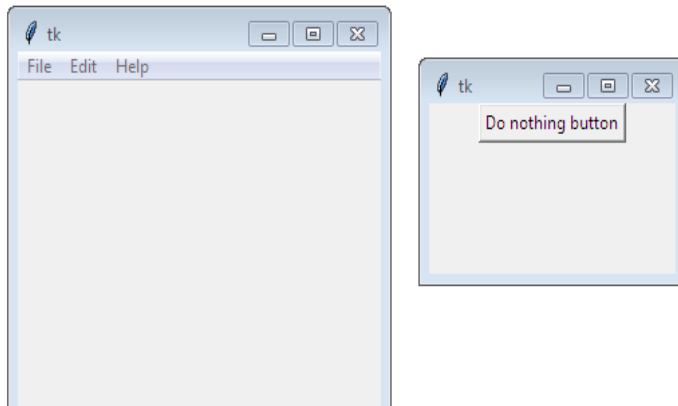
editmenu.add_command(label="Cut", command=donothing)
editmenu.add_command(label="Copy", command=donothing)
editmenu.add_command(label="Paste", command=donothing)
editmenu.add_command(label="Delete", command=donothing)
editmenu.add_command(label="Select All", command=donothing)

menubar.add_cascade(label="Edit", menu=editmenu)
```

```
helpmenu = Menu(menubar, tearoff=0)
helpmenu.add_command(label="Help Index", command=donothing)
helpmenu.add_command(label="About...", command=donothing)
menubar.add_cascade(label="Help", menu=helpmenu)
```

```
root.config(menu=menubar)
root.mainloop()
```

```
>>>
RESTART: C:/Users/u/AppData/Local/Programs/Python/Python36-32/MenuTkinter.py
```





**7. Write a Python Program to Show the concept of Exception handling.**

```
a=10
b=5
try:
    d=a/b
    print(d)
except ZeroDivisionError:
    print("Division by zero not allowed")
print("Rest of the code")
```

**OUTPUT:**

```
Division by zero not allowed
Rest of the code
```

**8. Write a Program in Python that show the use of following Built-In Functions.**

**i) append()      ii) reverse()      iii) index      iv) getattr()      v) setattr()**

**i) append()**

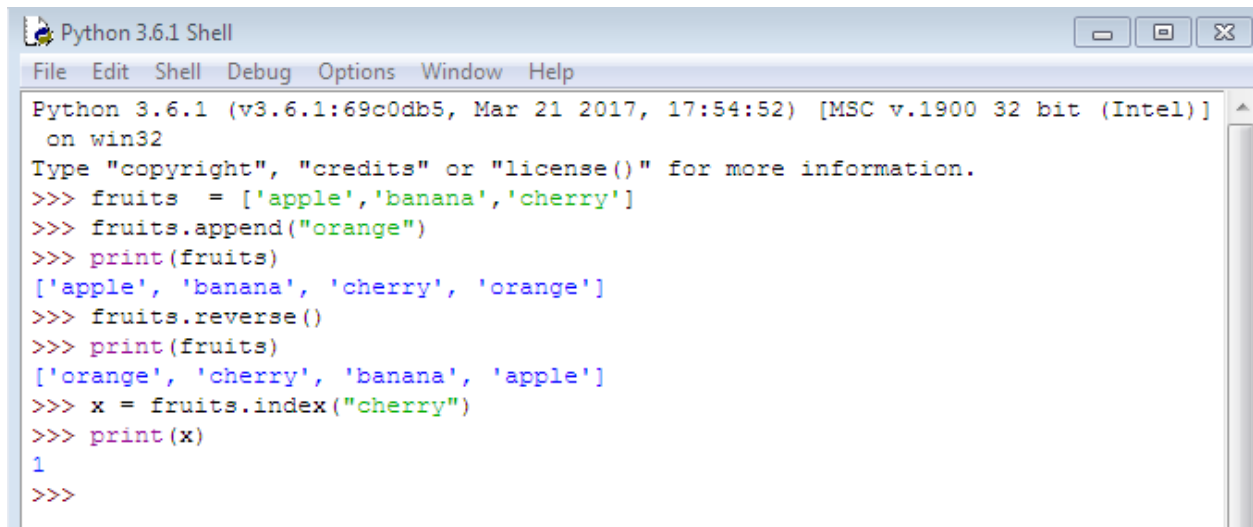
```
fruits = ['apple', 'banana', 'cherry']  
fruits.append("orange")  
print(fruits)
```

**ii) reverse()**

```
fruits = ['apple', 'banana', 'cherry']  
fruits.reverse()
```

**iii) index**

```
fruits = ['apple', 'banana', 'cherry']  
x = fruits.index("cherry")
```



```
Python 3.6.1 Shell  
File Edit Shell Debug Options Window Help  
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]  
on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> fruits = ['apple', 'banana', 'cherry']  
>>> fruits.append("orange")  
>>> print(fruits)  
['apple', 'banana', 'cherry', 'orange']  
>>> fruits.reverse()  
>>> print(fruits)  
['orange', 'cherry', 'banana', 'apple']  
>>> x = fruits.index("cherry")  
>>> print(x)  
1  
>>>
```

#### iv) `getattr()`

```
class Person:  
    name = "John"  
    age = 36  
    country = "Norway"
```

```
x = getattr(Person, 'age')
```

**OUTPUT: 36**

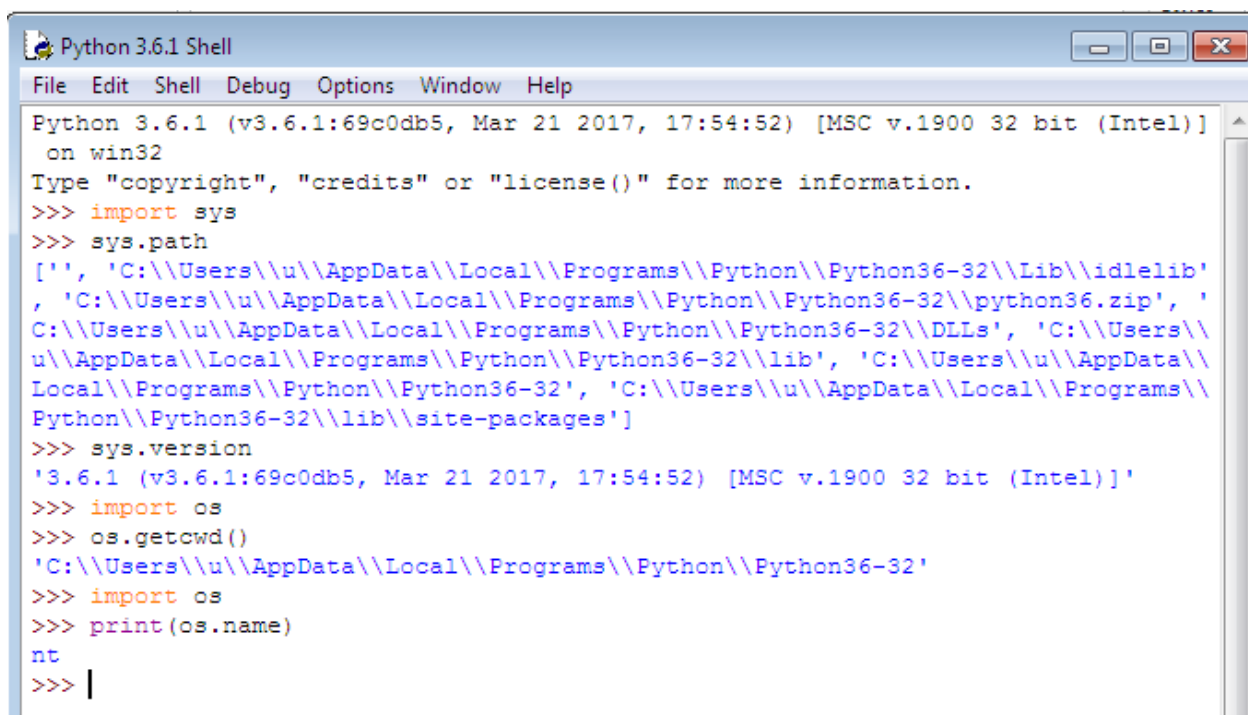
#### v) `setattr()`

```
class Person:  
    name = "John"  
    age = 36  
    country = "Norway"
```

```
setattr(Person, 'age', 40)
```

**OUTPUT: 40**

**9. Write a Python program that Show the OS name, Version of System, path and Current working directory.**

A screenshot of a 'Python 3.6.1 Shell' window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following code and output:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> import sys
>>> sys.path
['', 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\Lib\\idlelib',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\python36.zip',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\DLLs', 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\lib',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\lib\\site-packages']
>>> sys.version
'3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]'
>>> import os
>>> os.getcwd()
'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32'
>>> import os
>>> print(os.name)
nt
>>> |
```

```
>>> import sys
>>> sys.path
['', 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\Lib\\idlelib',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\python36.zip',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\DLLs',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\lib',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32',
 'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32\\lib\\site-packages']

>>> sys.version
'3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]'
```

```
>>> import os
```

```
>>> os.getcwd()
```

```
'C:\\Users\\u\\AppData\\Local\\Programs\\Python\\Python36-32'
```

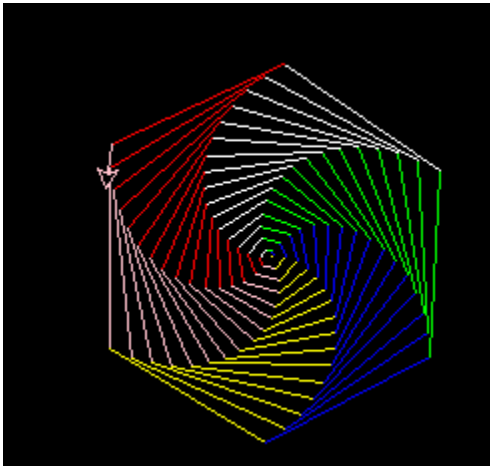
```
>>> import os
```

```
>>> print(os.name)
```

**10. Write a Python program to draw Colorful Star using Turtle module.**

```
import turtle
star = turtle.Turtle()
for i in range(100):
    star.forward(100)
    star.right(144)
    turtle.done()
```

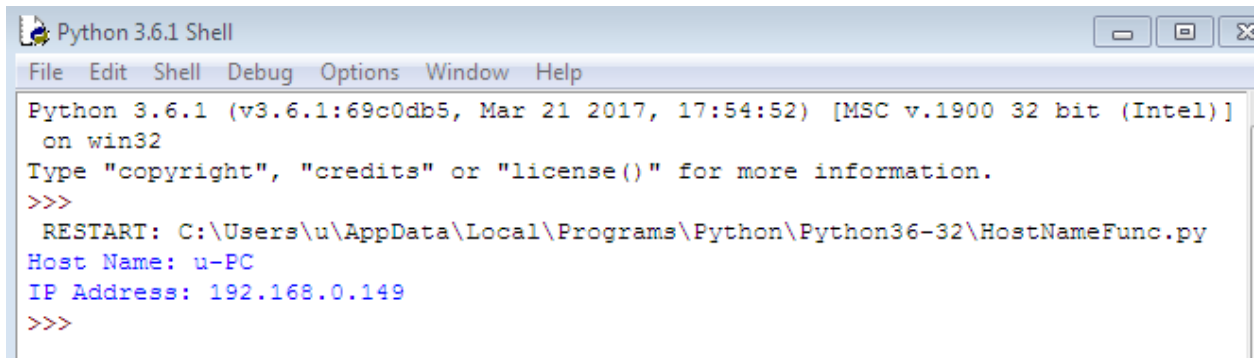
OUTPUT:



## 11. Write a Python Program that Show HostName and IP Address using Socket module.

```
import socket
def print_machine_info():
    host_name = socket.gethostname()
    ip_address = socket.gethostbyname(host_name)
    print("Host Name: %s" %host_name)
    print("IP Address: %s" %ip_address)
if __name__ == '__main__':
    print_machine_info()
```

### OUTPUT:

A screenshot of a Python 3.6.1 Shell window. The window has a title bar that says "Python 3.6.1 Shell" and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area of the window shows the output of a Python script. It starts with the Python version and build information: "Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32". Then it says "Type 'copyright', 'credits' or 'license()' for more information." followed by a prompt ">>>". The user has entered a command to restart the shell, and the output shows the file path: "RESTART: C:\Users\u\AppData\Local\Programs\Python\Python36-32\HostNameFunc.py". Finally, the script's output is displayed: "Host Name: u-PC" and "IP Address: 192.168.0.149", followed by another prompt ">>>".

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\u\AppData\Local\Programs\Python\Python36-32\HostNameFunc.py
Host Name: u-PC
IP Address: 192.168.0.149
>>>
```