**Multiple and Multilevel Inheritance**

**LAB #** **06**

**Fall 2019**

**CSE208L Object Oriented Programming Lab**

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Sumayyea Salahuddin**

January 6, 2020

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Objectives of the Lab:**

Objectives of the lab are to:

# Understand the concept of multiple and multilevel inheritance.

# Write two level inherited classes.

* Write a class inherited from multiple base classes.
* Write multi-file programs covering inheritance.

# Activity # 01

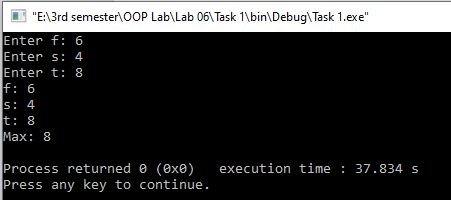
**Title:**

Make a Parent class for First, inherit class Second from First and class Third from Second.

**In C++**

**Source code: Output:**

#include <iostream>



using namespace std;

class First

{

protected:

int f;

public:

void f\_input()

{

cout<<"Enter f: ";

cin>>f;

}

};

class Second: public First

{

protected:

int s;

public:

void s\_input()

{

f\_input();

cout<<"Enter s: ";

cin>>s;

}

};

class Third: public Second

{

protected:

int t;

public:

void t\_input()

{

s\_input();

cout<<"Enter t: ";

cin>>t;

}

void Max()

{

if(f>s && f>t)

cout<<"Max: "<<f<<endl;

else if (s>f && s>t)

cout<<"Max: "<<s<<endl;

else if (t>f && t>s)

cout<<"Max: "<<t<<endl;

}

void Show()

{

cout<<"f: "<<f<<endl;

cout<<"s: "<<s<<endl;

cout<<"t: "<<t<<endl;

}

};

int main()

{

Third t1;

t1.t\_input();

t1.Show();

t1.Max();

return 0;

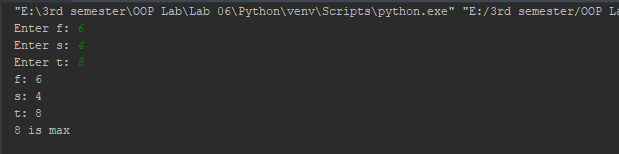
}

**In Python**

**Source code:**

class First:  
 def in1(self):  
 self.f=input("Enter f: ")  
  
class Second(First):  
 def in1(self):  
 self.f= input("Enter f: ")  
 self.s=input("Enter s: ")  
  
class Third(Second):  
 def in1(self):  
 self.f=input("Enter f: ")  
 self.s = input("Enter s: ")  
 self.t=input("Enter t: ")  
 def max(self):  
 if (self.f>self.s and self.f>self.t):  
 print(self.f+" is max ")  
 elif (self.s>self.f and self.s>self.t):  
 print(self.s+" is max ")  
 elif (self.t>self.f and self.t>self.s):  
 print(self.t+" is max ")  
 def show(self):  
 print("f: {}".format(self.f))  
 print("s: {}".format(self.s))  
 print("t: {}".format(self.t))  
  
t1 = Third()  
t1.in1()  
t1.show()  
t1.max()

**Output:**



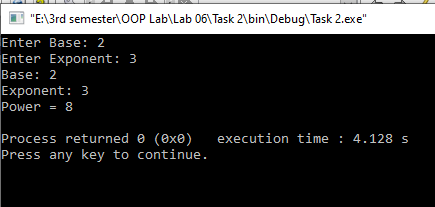
# Activity # 02

**Title:**

Make a class Base and Exponent and inherit class Power from Base and Exponent.

**In C++**

**Source code: Output:**

#include <iostream>

using namespace std;

class Base

{

protected:

int ba;

public:

void input\_base()

{

cout<<"Enter Base: ";

cin>>ba;

}

int show\_base()

{

return ba;

}

};

class Exponent

{

protected:

int exp;

public:

void input\_exp()

{

cout<<"Enter Exponent: ";

cin>>exp;

}

int show\_exp()

{

return exp;

}

};

class Power: public Base, Exponent

{

protected:

int po;

public:

Power():po(1){}

void in1()

{

input\_base();

input\_exp();

}

void show1()

{

cout<<"Base: "<<show\_base()<<endl;

cout<<"Exponent: "<<show\_exp()<<endl;

for(int i=0;i<show\_exp();i++)

{

po\*=show\_base();

}

cout<<"Power = "<<po<<endl;

}

};

int main()

{

Power p1;

p1.in1();

p1.show1();

return 0;

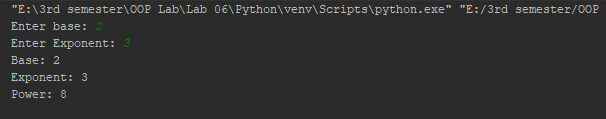
}

**In Python**

**Source code:**

class Base:  
 def input\_base(self):  
 self.ba=int(input("Enter base: "))  
 def show\_base(self):  
 print("Base: {}".format(self.ba))  
class Exponent:  
 def input\_exp(self):  
 self.exp=int(input("Enter Exponent: "))  
 def show\_exp(self):  
 print("Exponent: {}".format(self.exp))  
class Power(Base,Exponent):  
 def \_\_init\_\_(self):  
 self.po=1  
 def in1(self):  
 self.input\_base()  
 self.input\_exp()  
 def show1(self):  
 self.show\_base()  
 self.show\_exp()  
 for i in range(self.exp):  
 self.po\*=self.ba  
 print("Power: {}".format(self.po))  
p1 = Power()  
p1.in1()  
p1.show1()

**Output:**



# Activity # 03

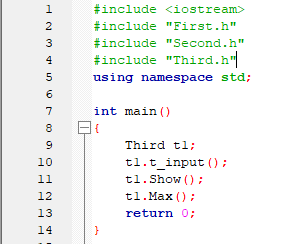
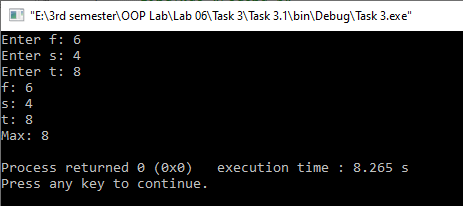
**Title:**

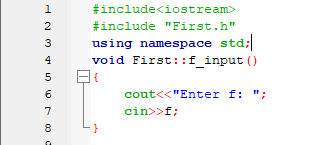
Redo Activity 1 and Activity 2 using multi-file Programming.

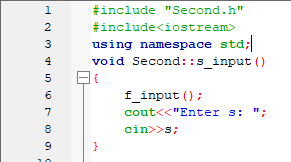
**Activity 1(Multi-File):**

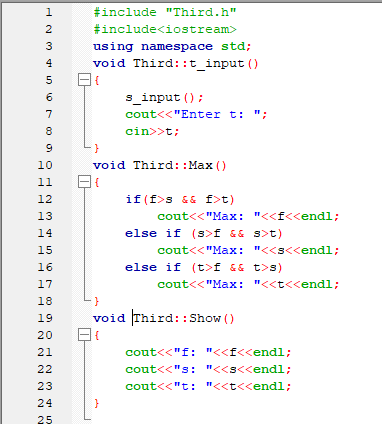
**In C++**

**Source code: Output:**









**In Python**

**Source code:**

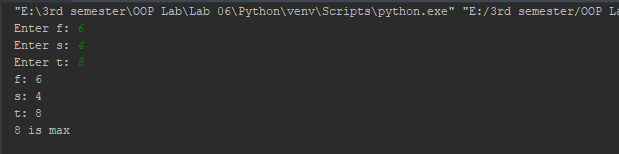
Main:

from task1classes import \*  
  
t1 = Third()  
t1.in1()  
t1.show()  
t1.max()

Classes:

class First:  
 def in1(self):  
 self.f=input("Enter f: ")  
  
class Second(First):  
 def in1(self):  
 self.f= input("Enter f: ")  
 self.s=input("Enter s: ")  
  
class Third(Second):  
 def in1(self):  
 self.f=input("Enter f: ")  
 self.s = input("Enter s: ")  
 self.t=input("Enter t: ")  
 def max(self):  
 if (self.f>self.s and self.f>self.t):  
 print(self.f+" is max ")  
 elif (self.s>self.f and self.s>self.t):  
 print(self.s+" is max ")  
 elif (self.t>self.f and self.t>self.s):  
 print(self.t+" is max ")  
 def show(self):  
 print("f: {}".format(self.f))  
 print("s: {}".format(self.s))  
 print("t: {}".format(self.t))

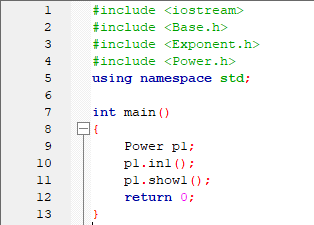
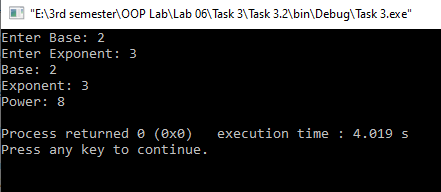
**Output:**

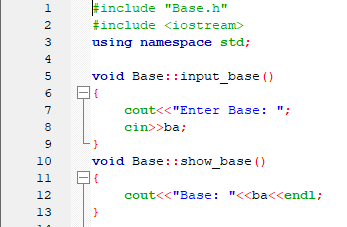


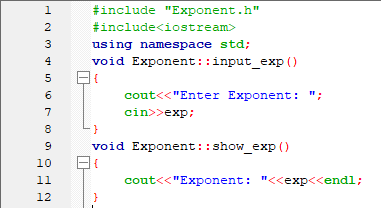
**Activity 2 (Multi-file):**

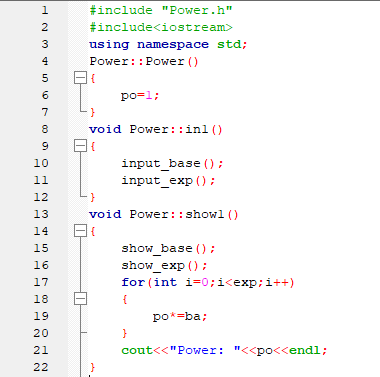
**In C++**

**Source code: Output:**









**In Python**

**Source code:**

Main:

from task2classes import \*  
p1 = Power()  
p1.in1()  
p1.show1()

Classes:

class Base:  
 def input\_base(self):  
 self.ba=int(input("Enter base: "))  
 def show\_base(self):  
 print("Base: {}".format(self.ba))  
class Exponent:  
 def input\_exp(self):  
 self.exp=int(input("Enter Exponent: "))  
 def show\_exp(self):  
 print("Exponent: {}".format(self.exp))  
class Power(Base,Exponent):  
 def \_\_init\_\_(self):  
 self.po=1  
 def in1(self):  
 self.input\_base()  
 self.input\_exp()  
 def show1(self):  
 self.show\_base()  
 self.show\_exp()  
 for i in range(self.exp):  
 self.po\*=self.ba  
 print("Power: {}".format(self.po))

**Output:**

