**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Object Oriented Programming Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 3rd**

**Submission Date: 08 February, 2022**

**Submitted to: Engr. Rafi-Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 10 & 11**

**Introduction to Abstract & Concrete Classes**

**&**

**Polymorphism and Abstract Base Class**

**Objectives:**

In this lab we will be discussing about Polymorphism in detail. One of the most important concepts in object-oriented programming is that of inheritance. Polymorphism is a mechanism that allows you to implement a function in different ways. Polymorphism is by far the most important and widely used concept in Object-Oriented programming.

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

**Virtual Members:**

A virtual member is a member of a base class that we can redefine in its derived classes. To declare a member as virtual we must use the keyword virtual.

**Polymorphic Class:**

A class that declares or inherits a virtual function is called a **polymorphic class.**

**Abstract Base Classes:**

An abstract base class has one or more **pure virtual** member functions.

**Lab Tasks:**

**Task # 01:**

Declare a class “Fruits” with any two data Members and two Member functions. Derive 2 class Mango and Orange from this Fruits class with one data Member and Member function. Create two Pointer objects of Fruits class and convert one to Mango Class Pointer Object and One to Orange Class Pointer Object through explicit Cast.

**Code:**

**#include <iostream>**

**using namespace std;**

**class Fruits**

**{**

**protected:**

**int quantity;**

**int price=0;**

**public:**

**void set\_funct (int qnt)**

**{**

**quantity = qnt;**

**}**

**};**

**class Mango: public Fruits**

**{**

**public:**

**int display()**

**{**

**cout<<"Quantity of Mango is: "<<quantity<<endl;**

**price = (quantity\*100);**

**cout<<"Price of Mango is: "<<price<<endl;**

**}**

**};**

**class Orange: public Fruits**

**{**

**public:**

**int display()**

**{**

**cout<<"Quantity of Orange is : "<<quantity<<endl;**

**price = (quantity\*70);**

**cout<<"Price of Orange is: "<<price<<endl;**

**}**

**};**

**int main ()**

**{**

**Mango m1;**

**Orange o1;**

**int quant1,quant2;**

**Fruits \* f1 = &m1;**

**Fruits \* f2 = &o1;**

**cout<<"Enter the Quantity of Mangoes you want! (In KG): ";**

**cin>>quant1;**

**cout<<"Enter the Quantity of Oranges you want! (In Dozen): ";**

**cin>>quant2;**

**f1->set\_funct(quant1);**

**f2->set\_funct(quant2);**

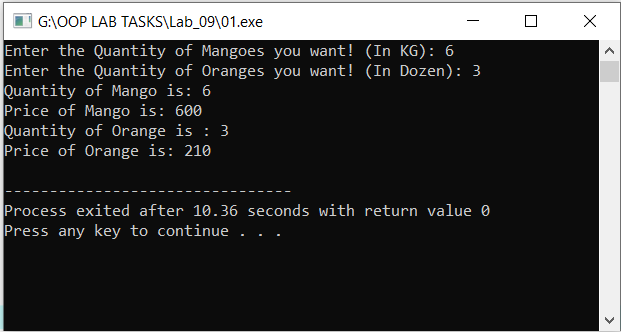
**m1.display();**

**o1.display();**

**return 0;**

**}**

**Output:**

****

**Task #02:**

Write a class “Parent” with data members name,age,dob and one “GrandParent” class with data members first\_name, address. Make one pointer type of each class. And cast the pointer type of Parent into GrandParent.

**Code:**

**#include <iostream>**

**using namespace std;**

**class Parent**

**{**

**protected:**

**string name;**

**int age,dob;**

**public:**

**void setup(string n, int a, int d)**

**{**

**name= n;**

**age=a;**

**dob=d;**

**}**

**void display()**

**{**

**cout<<"Name: "<<name<<endl;**

**cout<<"Age: "<<age<<endl;**

**cout<<"Date of Birth: "<<dob<<endl;**

**}**

**};**

**class GrandParent**

**{**

**public :**

**string first\_name, address;**

**void setup(string fn, string ad)**

**{**

**first\_name=fn;**

**address= ad;**

**}**

**void display()**

**{**

**cout<<"First Name: "<<first\_name<<endl;**

**cout<<"Address: "<<address<<endl;**

**}**

**};**

**int main ()**

**{**

**Parent p1;**

**GrandParent gp1;**

**Parent \*ptr\_p1= &p1;**

**GrandParent \*ptr\_p2= &gp1;**

**ptr\_p1->setup("Ahmed",19,2002);**

**ptr\_p2->setup("Khalid","Taxila");**

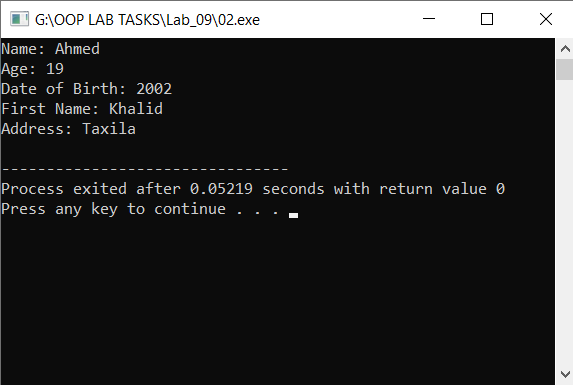
**p1.display();**

**gp1.display();**

**return 0;**

**}**

**Output:**

****

**Task #03:**

Write a C++ program which contain **Base Class** and **Derived Class (Cylinder)**. Base Class has two data members Radius, Height and a member function **CalculateArea(h, r).** Derived class override the function **CalculateArea(h, r)** and print the output using the Base class pointer and Derived class object. The output must be printed through overridden function in derived class.

**Note:** In this program, You must print your name as a message in the overridden function too.

**Code:**

**#include <iostream>**

**using namespace std;**

**class Cylinder**

**{**

**protected:**

**int radius=0, height=0, Area=0;**

**const float pi= 3.14;**

**public:**

**void Calculate(int h, int r)**

**{**

**height=h;**

**radius=r;**

**}**

**};**

**class SmallCylinder: public Cylinder**

**{**

**public:**

**void Calculate()**

**{**

**Area =(2\*pi\*radius\*height)+(2\*pi\*radius\*radius);**

**cout<<"Area of Cyclinder is: "<<Area;**

**}**

**};**

**int main ()**

**{**

**SmallCylinder s1;**

**int num1,num2;**

**Cylinder \* C1 = &s1;**

**cout<<"Enter the Radius: ";**

**cin>>num1;**

**cout<<"Enter the Height: ";**

**cin>>num2;**

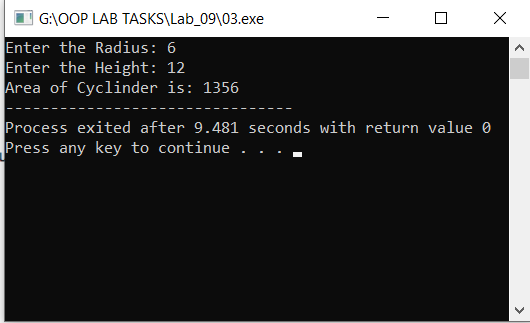
**C1->Calculate(num1, num2);**

**s1.Calculate();**

**return 0;**

**}**

**Output:**

****

**Task #04:**

Write a C++ program which has **Abstract Base Class (Students)** containing Pure Virtual function **ShowInfo()** and data members name, reg-no and home-address**. Derived Class** show these info using the **ShowInfo()** function.

**Code:**

**#include <iostream>**

**using namespace std;**

**class Student**

**{**

**protected:**

**int id=0;**

**string reg\_no, name, address;**

**public:**

**virtual void ShowInfo(string n, string r, int i, string ad)**

**{**

**name=n;**

**reg\_no=r;**

**id=i;**

**address=ad;**

**}**

**};**

**class SmallStudent: public Student**

**{**

**public:**

**void ShowInfo()**

**{**

**cout<<endl<<"Name of the student is: "<<name<<endl;**

**cout<<"Reg\_No of the student is: "<<reg\_no<<endl;**

**cout<<"Id of the student is: "<<id<<endl;**

**cout<<"Address of the student is: "<<address;**

**}**

**};**

**int main ()**

**{**

**SmallStudent s;**

**string name,reg,address;**

**int id;**

**Student \*S1 = &s;**

**cout<<"Enter the Name: ";**

**cin>>name;**

**cout<<"Enter the Reg\_No: ";**

**cin>>reg;**

**cout<<"Enter the id: ";**

**cin>>id;**

**cout<<"Enter the Address: ";**

**cin>>address;**

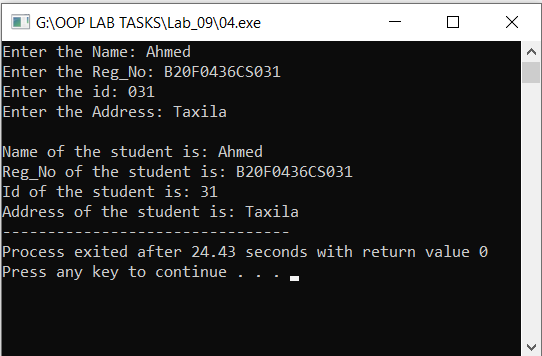
**S1->ShowInfo(name, reg, id, address);**

**s.ShowInfo();**

**return 0;**

**}**

**Output:**

****

**Results & Observations:**

In this Lab I’ve learned about the concept Polymorphism & also I understand the concept of Abstract Class that how can we can we make class an abstract class by using virtual function in the class.

**(1)** In the first task, I’ve used class “Fruits” with two data Members like Quantity & Price and two Member functions.Further I derive two more classese Mango and Orange from this Fruits class with one data Member and Member function like display(). And then I create two Pointer objects of Fruits class like:

**Fruits \* f1 = &m1;**

**Fruits \* f2 = &o1;**

and then convert one to Mango Class Pointer Object **m1** and One to Orange Class Pointer Object **o1** through explicit Cast.

**(2)** In the second task, I’ve used class **“Parent”** with data members **name,age,dob** and one **“GrandParent**” class with data members **first\_name, address**.

Furthermore I created one pointer type of each class like this:

**Parent \*ptr\_p1= &p1;**

& cast the pointer type of Parent into GrandParent.

**GrandParent gp1;**

**GrandParent \*ptr\_p2= &gp1;**

**(3)** In the third task, I’ve used **Base Class** and **Derived Class (Cylinder)**. Base Class has two data members **Radius, Height** and a member function **CalculateArea(h, r).** Further I derived a class that override the function **CalculateArea(h, r)** and print the output by using the Base class pointer:

**SmallCylinder s1;**

**Cylinder \* C1 = &s1;**

and Derived class object

**C1->Calculate(num1, num2);**

Here the output printed through overridden function in derived class.

**s1.Calculate();**

**(4)** In the third task, I’ve used Abstract Base Class (**Students**)containing Pure Virtual function **ShowInfo()** and data members **name, reg-no and home-address.** And also I’vecreated derived Class

**SmallStudent s;**

thatshow these info by using the **ShowInfo()** function.

**Student \*S1 = &s;**

**S1->ShowInfo(name, reg, id, address);**

**s.ShowInfo();**