**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: 4th February, 2022**

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

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

**Instructor Signature**

**Lab No. 08**

**Class Inheritance**

**Objectives:**

In this lab we will be discussing about Class Inheritance in detail. One of the most important concepts in object-oriented programming is that of inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the baseclass, and the new class is referred to as the derivedclass.

**Tools/Software Required:**

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

**Introduction:**

**Class Inheritance**

A class can be derived from more than one classes, which means it can inherit data and functions from multiple base classes. To define a derived class, we use a class derivation list to specify the base class(es). A class derivation list names one or more base classes and has the form:

**class derived-class: access-specifier base-class**

Where access-specifier is one of **public, protected,** or **private**, and base-class is the name of a previously defined class. If the access-specifier is not used, then it is private by default.

**Type of Inheritance:**

When deriving a class from a base class, the base class may be inherited through **public, protected** or **private** inheritance.

**Public Inheritance:**

When deriving a class from a **public** base class, **public** members of the base class become **public** members of the derived class and **protected** members of the base class become **protected** members of the derived class. A base class's **private** members are never accessible directly from a derived class, but can be accessed through calls to the **public** and **protected** members of the base class.

**Protected Inheritance:**

When deriving from a **protected** base class, **public** and **protected** members of the base class become **protected** members of the derived class.

**Private Inheritance:**

When deriving from a **private** base class, **public** and **protected** members of the base class become **private** members of the derived class.

**Multiple Inheritances:**

A C++ class can inherit members from more than one class and here is the extended syntax: **class derived-class: access baseA, access baseB**....

**Lab Tasks:**

**Task # 01:**

Write a program for **multiple inheritance** which has base classes **BasicInfo** and **DeptInfo** and derived class **Employee.** Derived class employee get basic info from the basic classes using inheritance.

**BasicInfo:** data members **name, empId, gender** and member function **getBasicInfo().**

**DeptInfo:** data members **deptName, assignedWork, time2complete** and function **getDeptInfo()**.

**Employee:** only member functions **getEmployeeInfo()** and **PrintEmployeeInfo()**

**Code:**

**#include <iostream>**

**using namespace std;**

**// Base class BasicInfo**

**class BasicInfo**

**{**

**public:**

**protected:**

**string Name, gender;**

**int empId;**

**void getBasicInfo(void)**

**{**

**cout<<"Enter employee's basic info:"<<endl;**

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

**cin>>Name;**

**cout<<"Enter Emp. Id: ";**

**cin>>empId;**

**cout<<"Enter Gender: ";**

**cin>>gender;**

**}**

**};**

**class DeptInfo**

**{**

**protected:**

**string depart\_name, assignedWork;**

**int time2complete;**

**void getDeptInfo(void)**

**{**

**cout<<"Enter employee's department info:"<<endl;**

**cout<<"Enter Department name: ";**

**cin>>depart\_name;**

**cout<<"Enter Assigned work: ";**

**cin>>assignedWork;**

**cout<<"Enter time in hours to complete work: ";**

**cin>>time2complete;**

**cout<<endl;**

**}**

**};**

**// Derived class**

**class Employee: public BasicInfo, public DeptInfo**

**{**

**public:**

**void getEmployeeInfo()**

**{**

**getBasicInfo();**

**getDeptInfo();**

**}**

**void PrintEmployeeInfo(void)**

**{**

**cout << "Employee's Information is: "<<endl;**

**cout << "BASIC INFORMATION..."<<endl;**

**cout << "Employee Name is: "<<Name<<endl;**

**cout << "Employee ID is: " <<empId<<endl;**

**cout << "Gender: "<<gender<<endl<<endl;**

**cout << "DEPARTMENT INFORMATION..."<<endl;**

**cout << "Department Name is: "<<depart\_name<< endl;**

**cout << "Assigned Work: "<< assignedWork<<endl;**

**cout << "Time to complete work: "<< time2complete<<endl;**

**}**

**};**

**int main()**

**{**

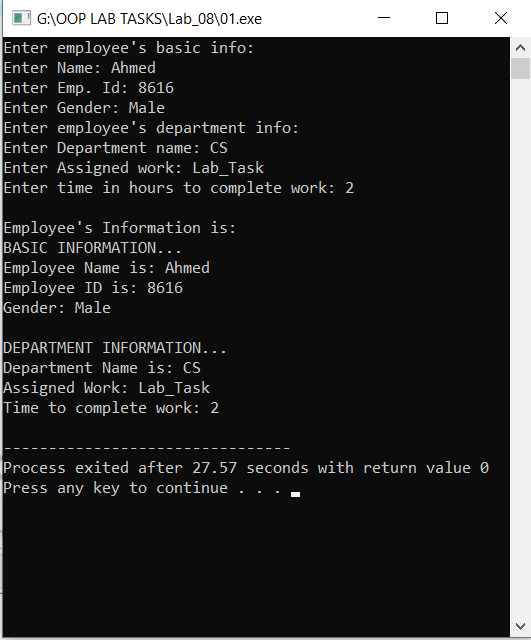
**Employee E;**

**E.getEmployeeInfo();**

**E.PrintEmployeeInfo();**

**}**

**Output:**

****

**Task #02:**

Assume that you are working with a film streaming company as a C++ developer. The company wants you to write C++ program which initialize a list of movies (see the table below). The list consists of 5 movies where each movie is represented by 3 fields: movie name, genre and IMDB ratings. Your code must provide the following functionalities:

1. User can search movies by name

2. User can find highest rated movie along with its details

3. User can see the list of all movies

Your program must draw a menu for above options where the user is repeatedly asked for his choice until the user selects exit option from the menu.

**Code:**

**#include <iostream>**

**using namespace std;**

**class List**

**{**

**private:**

**string movie\_name[5];**

**string genre[5];**

**float IMDB[5];**

**public:**

**movie()**

**{}**

**void Table(string mn[], string g[], float imdb[])**

**{**

**for(int i=0;i<5;i++)**

**{**

**movie\_name[i]=mn[i];**

**genre[i]=g[i];**

**IMDB[i]=imdb[i];**

**}**

**}**

**// For Searching Movie by its Name**

**int search(string movie)**

**{**

**for(int i=0;i<5;i++)**

**{**

**if(movie\_name[i]==movie)**

**{**

**cout<<"Movie Found\n\n";**

**cout <<"\t"<<movie\_name[i]<<"\t"<<genre[i]<<"\t"<<IMDB[i];**

**return 0;**

**}**

**}**

**cout<<"\n\nMovie NOT Found";**

**}**

**// For finding Highest Rate of movie**

**void High\_Rating()**

**{**

**int high=IMDB[0];**

**int j;**

**for(int i=1;i<5;i++)**

**{**

**if(IMDB[i]>high)**

**{**

**j=i;**

**high=IMDB[i];**

**}**

**}**

**cout << "\t" << movie\_name[j] << "\t" << genre[j] << "\t" << IMDB[j];**

**}**

**void Display()**

**{**

**cout << "\n\n\tMovie\t\t\tGenre\t\t\tIMDB";**

**for(int i=0;i<5;i++)**

**{**

**cout <<"\n\t" << movie\_name[i] <<"\t\t" <<genre[i] <<"\t\t" << IMDB[i];**

**}**

**}**

**};**

**int main()**

**{**

**List l;**

**string mn[5] = {"Inception", "Peter\_Pan", "The\_Dark\_Knight", "Desperado", "Pulp\_Fiction"};**

**string g[5] = {"Action ", "Adventure", "Crime ", "Thriller", "Crime "};**

**float f[5] = {8.8,7.3,9,7.2,8.9};**

**string movie;**

**l.Table(mn,g,f);**

**int num;**

**cout<<"\t\tMovie and its Ratings\n\n";**

**do**

**{**

**cout<<"\n\n1. Search Movie by name\n2. Find Highest Rated Movie\n3. List all Movies\nSelect option : ";**

**cin>>num;**

**switch(num)**

**{**

**case 1:**

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

**cin>>movie;**

**l.search(movie);**

**break;**

**case 2:**

**l.High\_Rating();**

**break;**

**case 3:**

**l.Display();**

**break;**

**default:**

**l.Display();**

**break;**

**}**

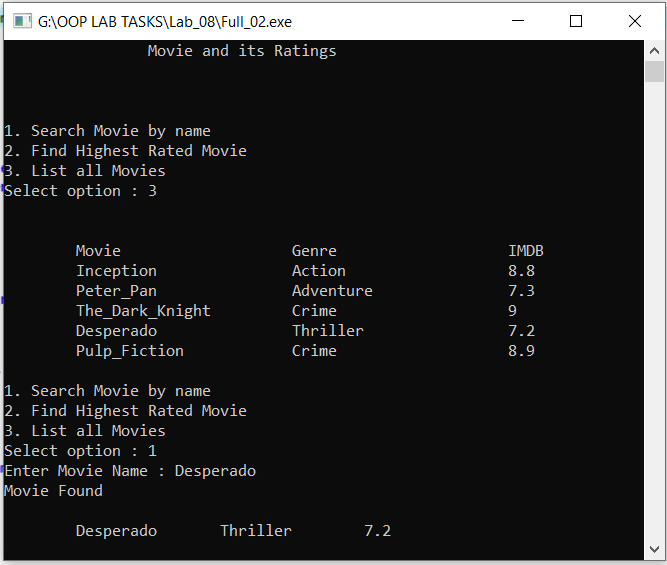
**}**

**while(num!=-1);**

**return 0;**

**}**

**Output:**

****

**Results & Observations:**

In this Lab I’ve learned about the concept Class Inheritance & also I understand the concept of Multiple Inheritance that how can we use Multiple Inheritance in the class. As well as I learned about Public, Private and Protected Inheritance. In the first task, I’ve used multiple inheritancewhich has base classes BasicInfo and DeptInfoand derived class Employee**.** Derived class employee get basic info from the basic classes using inheritance. And in the second, I’ve used class List and some functions that’s actually searching movie by name or by it’s highest rate. In the end I’ve used Switch statement to ask user to perform which task.