

Basic Object Oriented Programming

LAB MANUAL

Department

Of

Computer Engineering

# Neotech Faculty of Diploma Engineering

**Name: - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Branch: - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Division: - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Roll No: - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Year: - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NEOTECH FACULTY OF DIPLOMA ENGINEERING**

**Basic Object Oriented Programming(4311603)**



Neotech Faculty Of

Diploma Engineering

Virod, Vadodara.

CERTIFICATE

This is to certify that

Mr./Ms. Enrollment No. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of 2nd Semester Diploma course in

has satisfactorily completed his/her termwork in

within four walls of institute during the year 2022.

Place:

Date:

Subject Teacher Head of Department

**Subject: Basic Object Oriented Programming**

## Subject Code : 4311603

**LIST OF EXPERIMENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **NO.** | **TITLE OF EXPERIMENT** |  | **Date** | **Sign** |
| 1. | Create your account on github and save simple C++  Program in github |  |  |  |
| 2. | Develop minimum 5 program cin and cout |  |  |  |
| 3. | Develop program using scope resolution operator, simple manipulators , and enumeration |  |  |  |
| 4. | Develop programs using call by reference and return by reference, default arguments, constant arguments, inline and function overloading. |  |  |  |
| 5. | Develop programs using structures. |  |  |  |
| 6. | Define minimum 5 different classes such as student, distance, shape, employee, account, inventory, vector, movie-ticket booking, time, point, etc. with data member & member functions. Also Develop programs to test those classes functionality. |  |  |  |
| 7. | Develop Programs using array of objects and static member function |  |  |  |
| 8. | Develop programs to pass object as an argument and Returning object. |  |  |  |
| 9. | Develop programs using friend function and Friend class. |  |  |  |
| 10. | Apply the concepts of constructors and destructors in the Programs developed in unit-2 and test those programs. |  |  |  |
| 11. | Develop programs using single, multilevel, multiple Inheritance |  |  |  |
| 12. | Develop programs using Constructors in base and derive classes |  |  |  |
| 13. | Develop a program to show use of this pointer |  |  |  |
| 14. | Develop a program using runtime polymorphism |  |  |  |
| 15. | Develop at least 2 programs using file operations |  |  |  |

Q.2 ) Develop minimum 5 program using cin and cout.

Program1

🡪Addition of two number

**#include<iostream.h>**

**#include<conio.h>**

**int main()**

**{**

**int num1,num2,num3;**

**cout<<"Enter the number 1 : ";**

**cin>>num1;**

**cout<<"Enter the number 2 : ";**

**cin>>num2;**

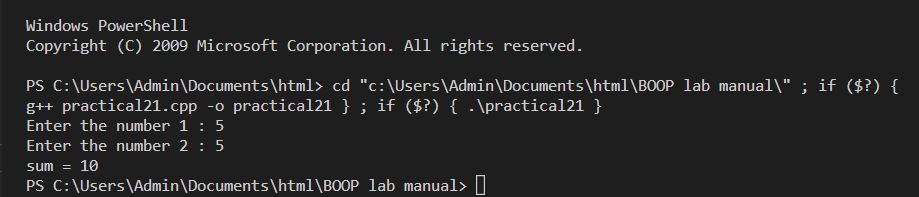
**num3=num1+num2;**

**cout<<"sum = "<<num3<<endl;**

**return 0;**

**}**

Output



Program2

🡪Subtraction of two number

**#include<iostream.h>**

**#include<conio.h>**

**int main()**

**{**

**int num1,num2,num3;**

**cout<<"Enter the number 1 : ";**

**cin>>num1;**

**cout<<"Enter the number 2 : ";**

**cin>>num2;**

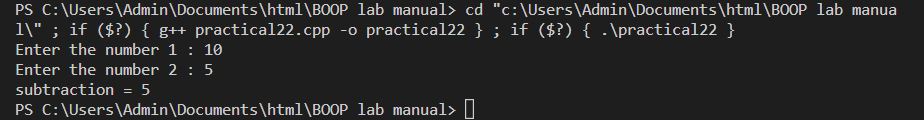
**num3=num1-num2;**

**cout<<"subtraction = "<<num3<<endl;**

**return 0;**

**}**

Output



Program 3

🡪Multiplication of two number

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**int main()**

**{**

**int num1,num2,num3;**

**cout<<"Enter the number 1 : ";**

**cin>>num1;**

**cout<<"Enter the number 2 : ";**

**cin>>num2;**

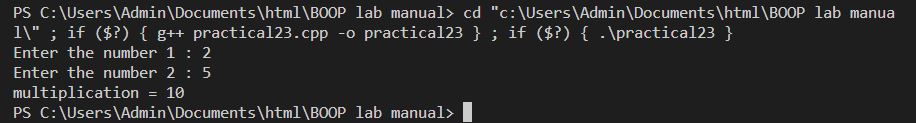
**num3=num1\*num2;**

**cout<<"multiplication = "<<num3<<endl;**

**return 0;**

**}**

Output



Program 4

🡪Division of two number

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**int main()**

**{**

**int num1,num2,num3;**

**cout<<"Enter the number 1 : ";**

**cin>>num1;**

**cout<<"Enter the number 2 : ";**

**cin>>num2;**

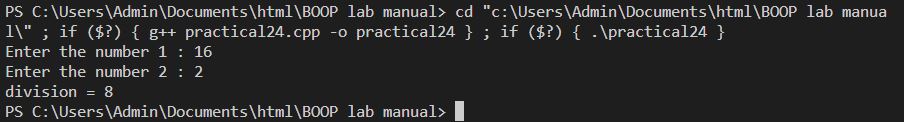
**num3=num1/num2;**

**cout<<"division = "<<num3<<endl;**

**return 0;**

**}**

Output



Program 5

🡪Program to print the number

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**int main()**

**{**

**int number;**

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

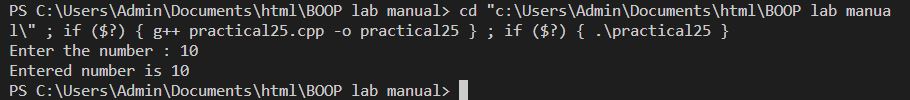
**cin>>number;**

**cout<<"Enter number is "<<number<<endl;**

**return 0;**

**}**

Output



Q.3 )Develop program using scope resolution operator ,simple manipulators ,and enumeration.

Program 1

🡪 scope resolution operator

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**int num = 100;**

**int main()**

**{**

**int num = 50;**

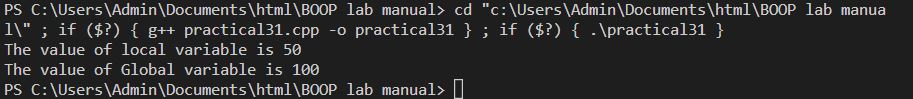
**cout<<"The value of local variable is "<<num<<endl;**

**cout<<"The value of Global variable is "<<::num<<endl;**

**return 0;**

**}**

Output

****

Program 2

🡪 Enumeration operator

**#include<iostream>**

**using namespace std;**

**enum weak{sunday,monday,tuesday,wednesday,thursday,friday,saturday};**

**int main()**

**{**

**weak today;**

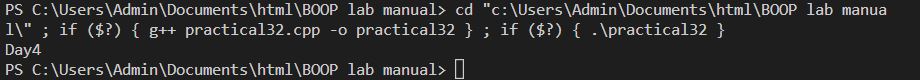
**today = wednesday;**

**cout<<"Day"<<today<<endl;**

**return 0;**

**}**

Output

****

Program 3

🡪Manipulator

**#include<iostream>**

**#include<conio.h>**

**#include<iomanip>**

**using namespace std;**

**int main()**

**{**

**int num1=1,num2=20,num3=300;**

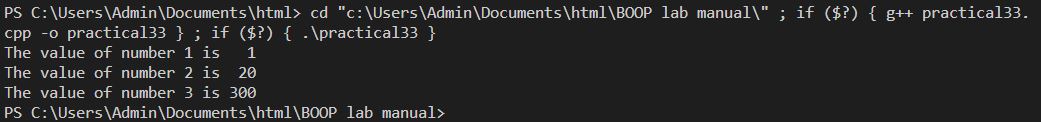
**cout<<"The value of number 1 is"<<setw(4)<<num1<<endl;**

**cout<<"The value of number 2 is"<<setw(4)<<num2<<endl;**

**cout<<"The value of number 3 is"<<setw(4)<<num3<<endl;**

**}**

Output



Q.4 )Develop program using call by reference and return by reference ,default arguments ,constant arguments, inline and function overloading.

🡪Call reference

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**void swap(int &,int &);**

**int main()**

**{**

**int num1,num2;**

**cout<<"Enter the number 1"<<endl;**

**cin>>num1;**

**cout<<"Enter the number 2"<<endl;**

**cin>>num2;**

**swap(num1,num2);**

**cout<<"The swap number 1 is "<<num1<<endl;**

**cout<<"The swap number 2 is "<<num2<<endl;**

**return 0;**

**}**

**void swap (int &var1,int &var2)**

**{**

**int temp;**

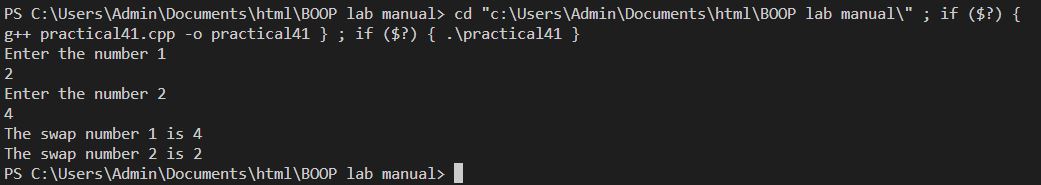
**temp=var1;**

**var1=var2;**

**var2=temp;**

**}**

Output

****

🡪Return by reference

**#include<iostream>**

**using namespace std;**

**int num;**

**int& test();**

**int main()**

**{**

**test()=5;**

**cout<<num<<endl;**

**return 0;**

**}**

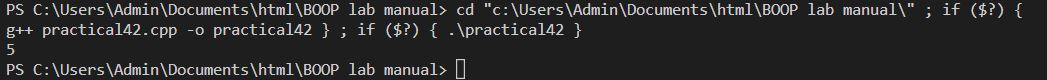
**int& test()**

**{**

**return num;**

**}**

**Output:**



🡪Default argument

**#include<iostream>**

**using namespace std;**

**int fun(int a, int b = 7)**

**{**

**cout<<a<<endl;**

**cout<<b<<endl;**

**}**

**int main()**

**{**

**cout<<"function call with two argument"<<endl;**

**fun(14,15);**

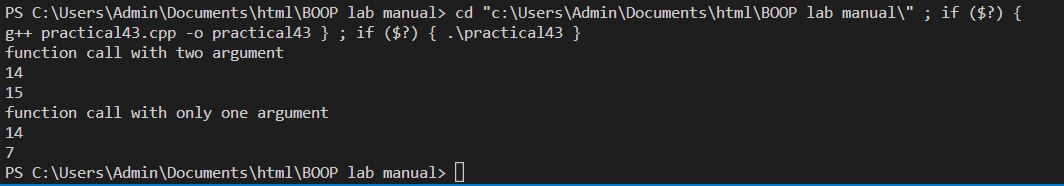
**cout<<"function call with only one argument"<<endl;**

**fun(14);**

**return 0;**

**}**

**Output:**



🡪constant argument

**#include<iostream>**

**using namespace std;**

**const int PI = 3.14;**

**int function(int x, int y)**

**{**

**return y\*x\*x;**

**}**

**int main()**

**{**

**int r, area;**

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

**cin>>r;**

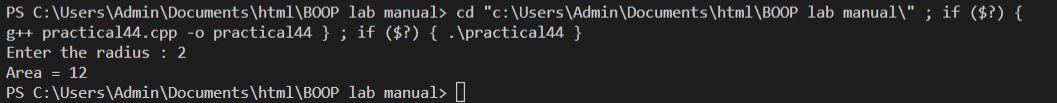
**area = function(r,PI);**

**cout<<"Area = "<<area<<endl;**

**return 0;**

**}**

**Output:**



🡪inline function

**#include<iostream>**

**using namespace std;**

**inline int fun(int x, int y)**

**{**

**return x + y;**

**}**

**int main()**

**{**

**int a ,b,c;**

**cout<<"Enter the two number:"<<endl;**

**cin>>a>>b;**

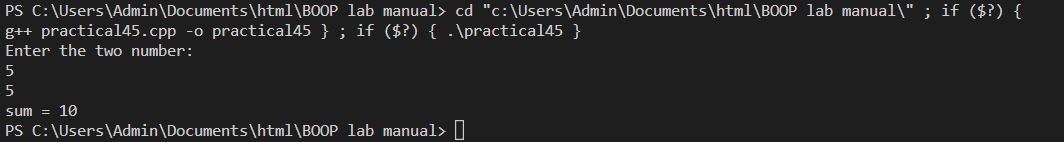
**c=fun(a,b);**

**cout<<"sum = "<<c<<endl;**

**return 0;**

**}**

**Output:**



🡪function overloading

**#include<iostream>**

**using namespace std;**

**int function1(int x ,int y)**

**{**

**return x+y;**

**}**

**float function1(float x ,float y)**

**{**

**return x\*y;**

**}**

**int main()**

**{**

**int a,b;**

**cout<<"Enter the numbers : "<<endl;**

**cin>>a>>b;**

**cout<<"Sum = "<<function1(a,b)<<endl;**

**float i,j;**

**cout<<"Enter the numbers : "<<endl;**

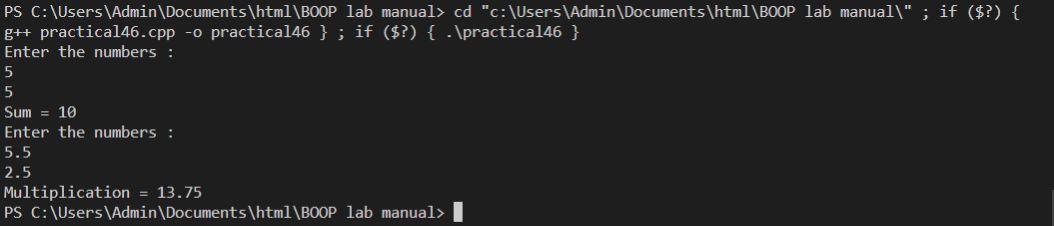
**cin>>i>>j;**

**cout<<"Multiplication = "<<function1(i,j)<<endl;**

**return 0;**

**}**

Output:



Q.5 ) Develop program using structure.

🡪Structure

**#include<iostream>**

**#include<string.h>**

**using namespace std;**

**struct student**

**{**

**int Enroll;**

**char name[20];**

**int age;**

**int sem;**

**};**

**int main()**

**{**

**struct student a1;**

**cout<<"Enter the Enrollement number : ";**

**cin>>a1.Enroll;**

**cout<<"Enter the name of the student : ";**

**cin>>a1.name;**

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

**cin>>a1.age;**

**cout<<"Enter the semester of the student : ";**

**cin>>a1.sem;**

**cout<<endl;**

**cout<<"-------The entered information are--------"**

**cout<<"The name of the student is : "<<a1.name<<endl;**

**cout<<"The Enrollement number of the student is : "<<a1.Enroll<<endl;**

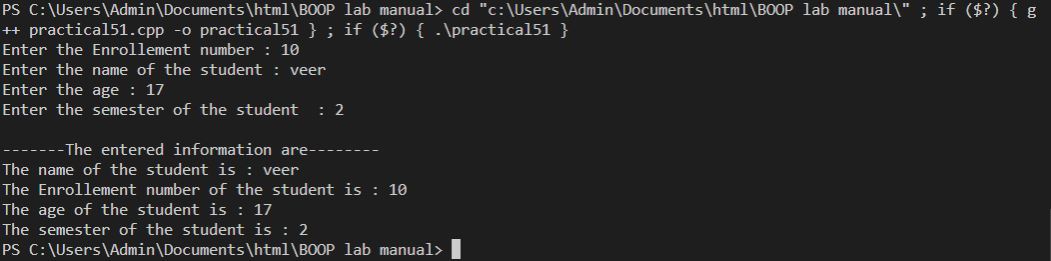
**cout<<"The age of the student is : "<<a1.age<<endl;**

**cout<<"The semester of the student is : "<<a1.sem<<endl;**

**return 0;**

**}**

Output



Q.6 ) Define minimum 5 different classes such as student, distance, shape, employee, account, inventory, vector, movie-ticket booking, time, point, etc. with data member & member functions. Also Develop programs to test those classes functionality

* program 1

🡪

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**class student**

**{**

**private:**

**int enrollement\_number;**

**char name[20];**

**char field[10];**

**int phone\_number;**

**int student\_number;**

**char student\_email[40];**

**public:**

**void getinfo(void)**

**{**

**cout<<"------Enter the student details------";**

**cout<<endl;**

**cout<<endl;**

**cout<<"Enter the student name"<<endl;**

**cin>>name;**

**cout<<"Enter the student enrollement number"<<endl;**

**cin>>enrollement\_number;**

**cout<<"Enter the student Field"<<endl;**

**cin>>field;**

**cout<<"Enter the parents phone number"<<endl;**

**cin>>phone\_number;**

**cout<<"Enter the student phone number"<<endl;**

**cin>>student\_number;**

**cout<<"Enter the student Email "<<endl;**

**cin>> student\_email;**

**}**

**void printinfo(void)**

**{**

**cout<<"Informations of the student"<<name;**

**cout<<endl;**

**cout<<endl;**

**cout<<"student full name :"<<name<<endl;**

**cout<<"student enrollement :"<<enrollement\_number<<endl;**

**cout<<"student Filed :"<<field <<endl;**

**cout<<"phone number :"<< phone\_number <<endl;**

**cout<<"student phone number :"<< student\_number <<endl;**

**cout<<"student Email :"<< student\_email <<endl;**

**}**

**};**

**int main()**

**{**

**student veer;**

**veer.getinfo();**

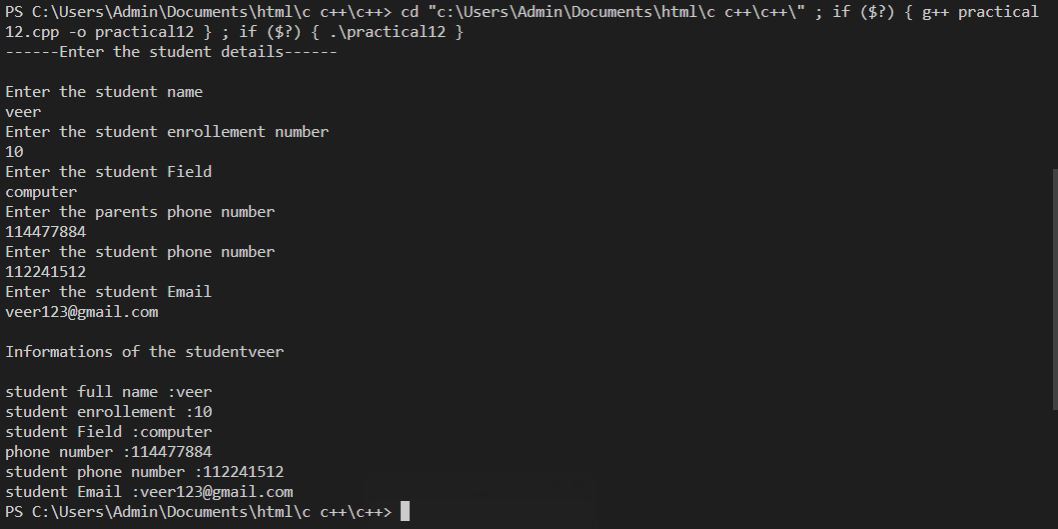
**cout<<endl;**

**veer.printinfo();**

**return 0;**

**}**

**Output**

****

🡪program 2

**#include<iostream>**

**#include<conio.h>**

**using namespace std;**

**class employ**

**{**

**private:**

**int id;**

**char employName[20];**

**int number;**

**char address[40];**

**int salary;**

**public:**

**void getinfo(void);**

**void printinfo(void);**

**};**

**void employ :: getinfo(void)**

**{**

**cout<<"Enter the employ information"<<endl;**

**cout<<" Enter the Employ ID : ";**

**cin>>id;**

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

**cin>>employName;**

**cout<<" Enter the employ phone number: ";**

**cin>>number;**

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

**cin>>address;**

**cout<<" Enter the salary of the employ: ";**

**cin>>salary;**

**}**

**void employ :: printinfo(void)**

**{**

**cout<<"The information of the Employ are "<<endl;**

**cout<<"The Employ ID is : "<<id<<endl;**

**cout<<"The Employ name is : "<<employName<<endl;**

**cout<<"The Employ phone number is : "<<number<<endl;**

**cout<<"The Employ Address is : "<<address<<endl;**

**cout<<"The Employ salary is : "<<salary<<endl;**

**}**

**int main()**

**{**

**employ veer;**

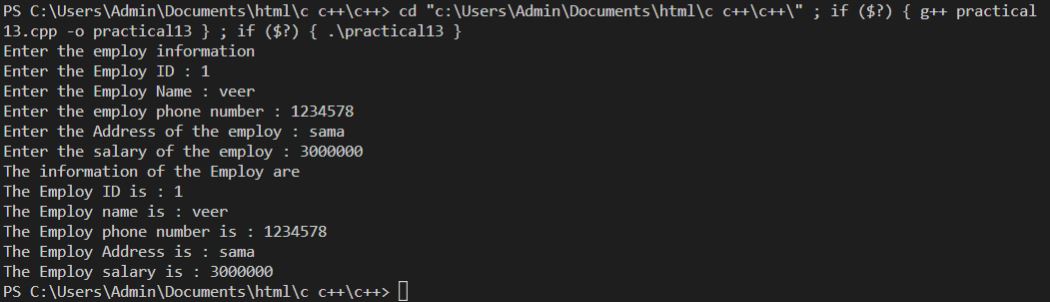
**veer.getinfo();**

**veer.printinfo();**

**return 0;**

**}**

**Output**



🡪program 3

**#include<iostream>**

**using namespace std;**

**class shape**

**{**

**private:**

**int side;**

**int height;**

**int width;**

**char sh\_name[20];**

**public:**

**void getInfo(void)**

**{**

**cout<<"Shape name : ";**

**cin>>sh\_name;**

**cout << "Enter the number of sides : ";**

**cin >> side;**

**cout << "Enter the height of shape : ";**

**cin >> height;**

**cout << "Enter the width of shape : ";**

**cin >> width;**

**}**

**void printInfo(void)**

**{**

**cout << "\_\_\_\_\_Entered information are\_\_\_\_\_\_ " << endl;**

**cout<<"Shape name : "<<sh\_name<<endl;**

**cout << "The number of sides of the shape are : " << side<< endl;**

**cout << "The height of the shape are : " << height<<"cm" << endl;**

**cout << "The width of the shape are : " << width <<"cm"<< endl;**

**}**

**};**

**int main()**

**{**

**shape a;**

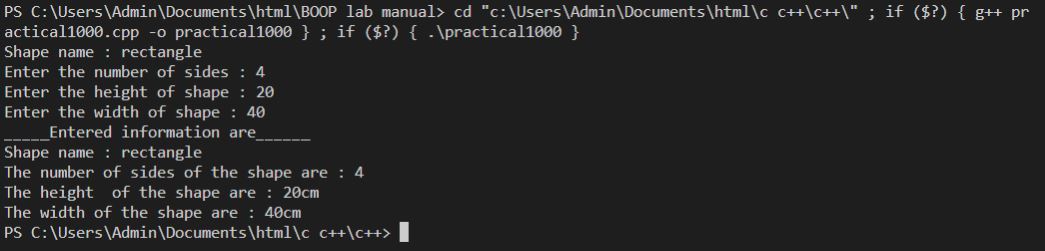
**a.getInfo();**

**a.printInfo();**

**return 0;**

**}**

**Output**

****

**🡪program 4**

**#include<iostream>**

**using namespace std;**

**class movie**

**{**

**private:**

**int num\_tic;**

**char movie[20];**

**char name[20];**

**int row;**

**char email[30];**

**int startseatnu;**

**int endseatnum;**

**public:**

**void getInfo(void)**

**{**

**cout<<"Enter the information";**

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

**cin>>name;**

**cout<<"Enter the name of the movie : ";**

**cin>>movie;**

**cout<<"Enter the number of ticket: ";**

**cin>>num\_tic;**

**cout<<"Enter the row of seat : ";**

**cin>>row;**

**cout<<"Enter the starting and ending seat"<<endl;**

**cin>>startseatnu;**

**cin>>endseatnum;**

**cout<<"Enter the Email number";**

**cin>>email;**

**}**

**void Printinfo(void)**

**{**

**cout<<endl;**

**cout<<"The enter inforamtion are "<<endl;**

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

**cout<<"Movie name : "<<movie<<endl;**

**cout<<"Number of ticket : "<<num\_tic<<endl;**

**cout<<"Seat row : "<<row<<endl;**

**cout<<"The seats are from "<<startseatnu<<"to"<<endseatnum<<endl;**

**cout<<"Email : "<<email<<endl;**

**}**

**};**

**int main()**

**{**

**movie a1;**

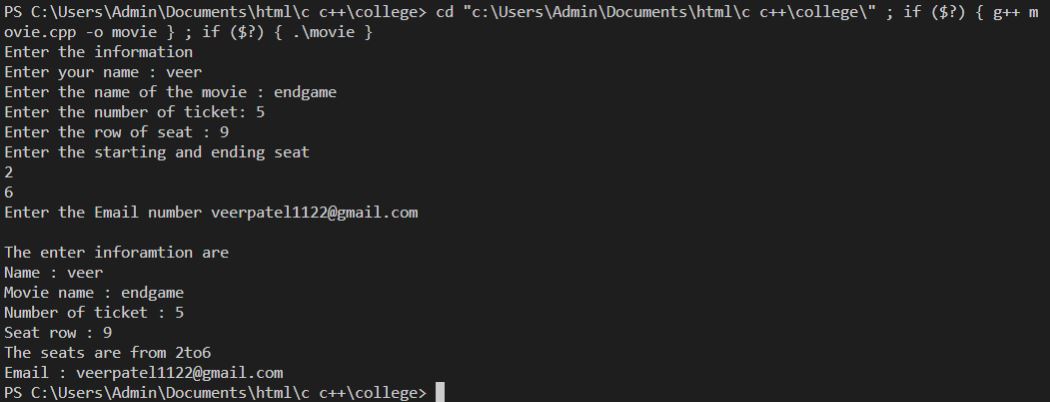
**a1.getInfo();**

**a1.Printinfo();**

**return 0;**

**}**

**Output**

****

**🡪program 5**

**#include<iostream>**

**using namespace std;**

**class account**

**{**

**private:**

**int ac\_number;**

**char ac\_owner[30];**

**int ifsc\_code;**

**int ac\_balance = 100000;**

**int pin;**

**public:**

**void getInfo(void)**

**{**

**cout<<"Enter the account information"<<endl;**

**cout<<"Enter the account number : ";**

**cin>>ac\_number;**

**cout<<"Enter the account owner name : ";**

**cin>>ac\_owner;**

**cout<<"Entert the IFSC code : ";**

**cin>>ifsc\_code;**

**cout<<"Enter the four digit pin : ";**

**cin>>pin;**

**cout<<endl;**

**}**

**void printInfo(void)**

**{**

**cout<<"The information you enter is as follows "<<endl;**

**cout<<"Account number :"<<ac\_number <<endl;**

**cout<<"Account owner :"<<ac\_owner <<endl;**

**cout<<"Account IFSC code :"<<ifsc\_code <<endl;**

**cout<<"Account balance :"<<ac\_balance <<endl;**

**}**

**};**

**int main()**

**{**

**account a1;**

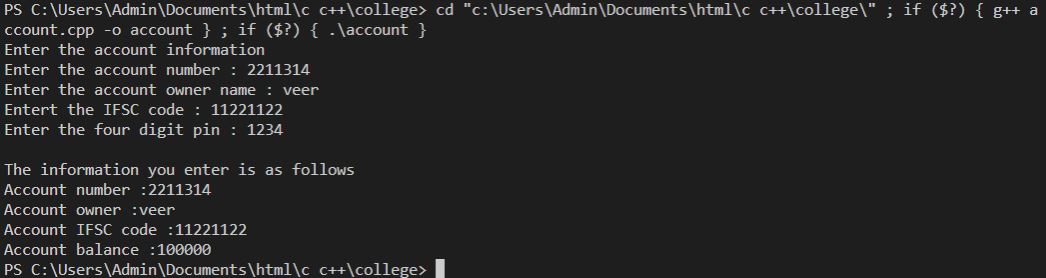
**a1.getInfo();**

**a1.printInfo();**

**return 0;**

**}**

**Output:**

****

Q.7 ) Develop a program using array of objects and static member function

🡪

**#include<iostream>**

**using namespace std;**

**class student**

**{**

**private:**

**static int count;**

**char name[20];**

**int enroll\_num;**

**int sem;**

**public:**

**void getInfo(void)**

**{**

**count++;**

**cout << endl << "-------Enter the information of student count" << count << "-------" << endl << endl;**

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

**cin>>name;**

**cout << "Enter the student Enrollement number : ";**

**cin >> enroll\_num;**

**cout << "Enter the student semester : ";**

**cin >> sem;**

**}**

**};**

**int student::count = 0;**

**int main()**

**{**

**student a[3];**

**int i;**

**for ( i = 0; i < 3; i++)**

**{**

**a[i].getInfo();**

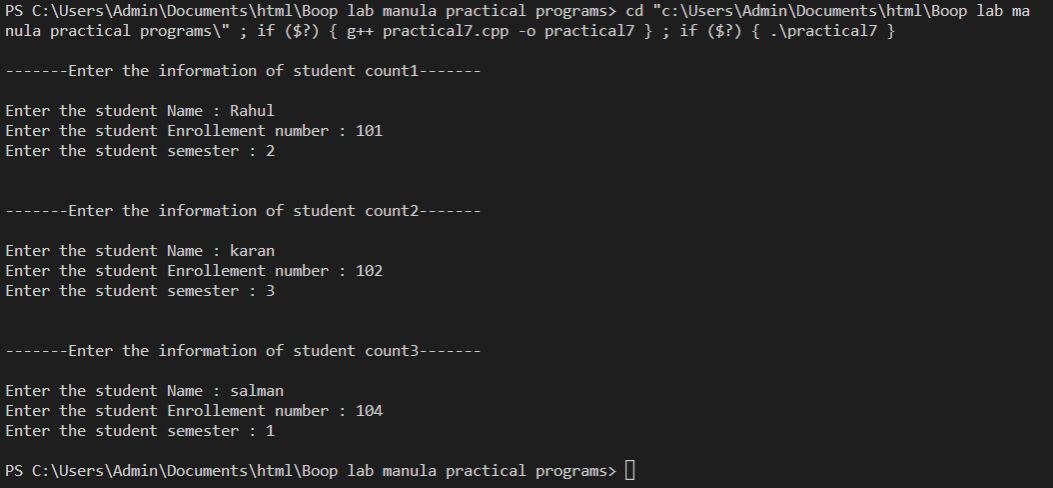
**cout<<endl;**

**}**

**return 0;**

**}**

**Output:**

****

Q.8/9 )

🡪

**#include<iostream>**

**using namespace std;**

**class s2;**

**class s1**

**{**

**int a;**

**public:**

**int setinfo(int x)**

**{**

**a=x;**

**}**

**void printinfo(void)**

**{**

**cout<<"The value of a is "<<a<<endl;**

**}**

**friend void swap(s1 &,s2 &);**

**};**

**class s2**

**{**

**int b;**

**public:**

**int setinfo(int x)**

**{**

**b=x;**

**}**

**void printinfo(void)**

**{**

**cout<<"The value of b is "<<b<<endl;**

**}**

**friend void swap(s1 &,s2 &);**

**};**

**void swap(s1 & o1, s2 & o2)**

**{**

**int temp;**

**temp = o1.a;**

**o1.a = o2.b;**

**o2.b = temp;**

**}**

**int main()**

**{**

**s1 a1;**

**s2 a2;**

**a1.setinfo(3);**

**a1.printinfo();**

**a2.setinfo(5);**

**a2.printinfo();**

**swap(a1,a2);**

**cout<<"Values after swaping"<<endl;**

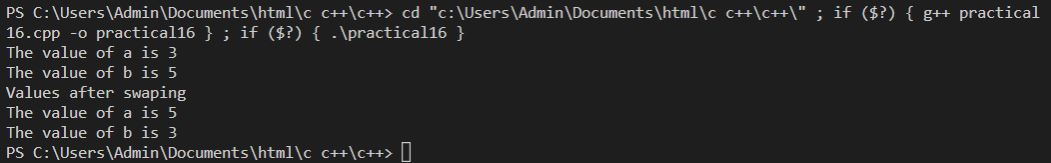
**a1.printinfo();**

**a2.printinfo();**

**return 0;**

**}**

**Output:**

****

Q10)Apply the concepts of constructors and destructors in the Programs developed in unit-2 and test those programs.

**🡪**

**#include<iostream>**

**using namespace std;**

**class base**

**{**

**public:**

**base()**

**{**

**cout<<"-------Constructor is called-------"<<endl;**

**}**

**~base()**

**{**

**cout<<"-------destructor is called-------"<<endl;**

**}**

**};**

**int main()**

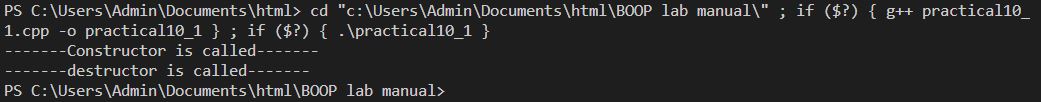
**{**

**base a;**

**return 0;**

**}**

**Output:**

****

**Q11)** Develop programs using single, multilevel, multiple Inheritance

**🡪 single level**

**#include<iostream>**

**using namespace std;**

**class base**

**{**

**int num1;**

**public:**

**int num2;**

**void setdata()**

**{**

**num1 = 10;**

**num2 = 30;**

**}**

**void getn()**

**{**

**cout<<"enter";**

**cin>>num1;**

**}**

**int getnum()**

**{**

**return num1;**

**}**

**};**

**class derived : public base**

**{**

**int num3;**

**public:**

**void process()**

**{**

**num3 = num2 + getnum();**

**}**

**void display()**

**{**

**cout << "------------The numbers are----------- "<<endl;**

**cout << "the num1 is " << getnum() << endl;**

**cout << "the num2 is " << num2 << endl;**

**cout << "the num3 is " << num3 << endl;**

**}**

**};**

**int main()**

**{**

**derived a;**

**a.setdata();**

**a.getn();**

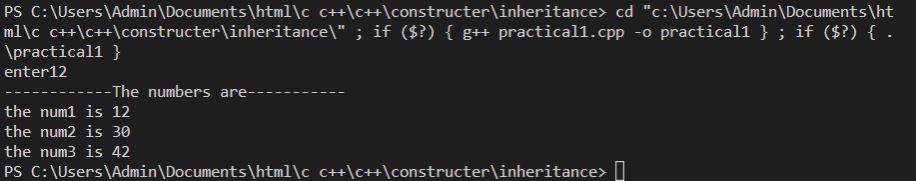
**a.process();**

**a.display();**

**return 0;**

**}**

**Output:**

****

**🡪Multi level**

**#include<iostream>**

**using namespace std;**

**class num1**

**{**

**int num\_1;**

**public:**

**void getNum1()**

**{**

**cout << "enter the number 1 : ";**

**cin >> num\_1;**

**}**

**int setNum1()**

**{**

**return num\_1;**

**}**

**};**

**class num2 :public num1**

**{**

**int num\_2;**

**public:**

**void getNum2()**

**{**

**cout << "enter the number 2 : ";**

**cin >> num\_2;**

**}**

**int setNum2()**

**{**

**return num\_2;**

**}**

**};**

**class sum : public num2**

**{**

**int sum\_num;**

**public:**

**void sum\_number()**

**{**

**cout << "sum = " << setNum1()+setNum2()<<endl;**

**}**

**};**

**int main()**

**{**

**sum a1;**

**a1.getNum1();**

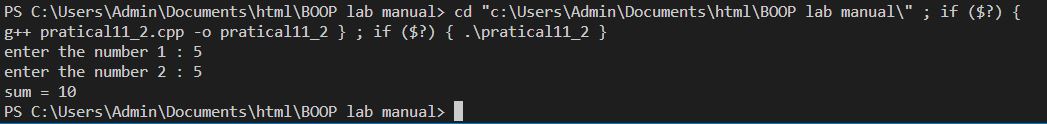
**a1.getNum2();**

**a1.sum\_number();**

**return 0;**

**}**

**Output:**

****

**🡪multiple**

**#include<iostream>**

**using namespace std;**

**class num1**

**{**

**public:**

**int num\_1;**

**void getNum1()**

**{**

**cout << "enter the number 1 : ";**

**cin >> num\_1;**

**}**

**};**

**class num2**

**{**

**public:**

**int num\_2;**

**void getNum2()**

**{**

**cout << "enter the number 2 : ";**

**cin >> num\_2;**

**}**

**};**

**class sum : public num1, public num2**

**{**

**public:**

**int sum\_num;**

**void sum\_number()**

**{**

**cout << "sum = " << num\_1 + num\_2 << endl;**

**}**

**};**

**int main()**

**{**

**sum a1;**

**a1.getNum1();**

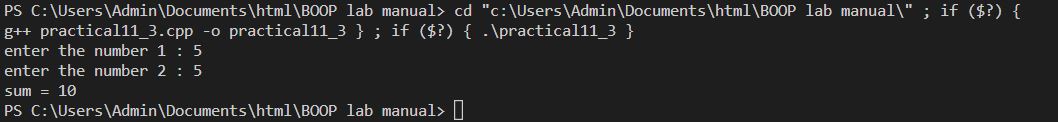
**a1.getNum2();**

**a1.sum\_number();**

**return 0;**

**}**

**Output:**

****

Q12)Develop programs using Constructors in base and derive classes

🡪

**#include<iostream>**

**using namespace std;**

**class base**

**{**

**public:**

**base()**

**{**

**cout<<"-----base class constructor is called-----"<<endl;**

**}**

**};**

**class derive : public base**

**{**

**public:**

**derive()**

**{**

**cout<<"-----derived class constructor is called-----"<<endl;**

**}**

**};**

**int main()**

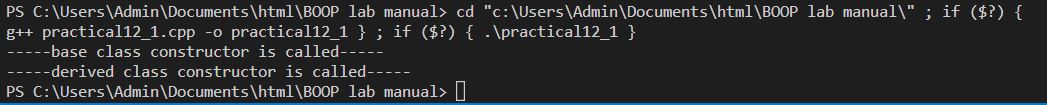
**{**

**derive a1;**

**return 0;**

**}**

**Output:**

****

**Q13)** Develop a program to show use of this pointer

**🡪**

**#include<iostream>**

**using namespace std;**

**void test(int\*, int\*);**

**int main() {**

**int a = 5, b = 5;**

**cout << "Before changing:" << endl;**

**cout << "a = " << a << endl;**

**cout << "b = " << b << endl;**

**test(&a, &b);**

**cout << "\nAfter changing" << endl;**

**cout << "a = " << a << endl;**

**cout << "b = " << b << endl;**

**return 0;**

**}**

**void test(int\* n1, int\* n2)**

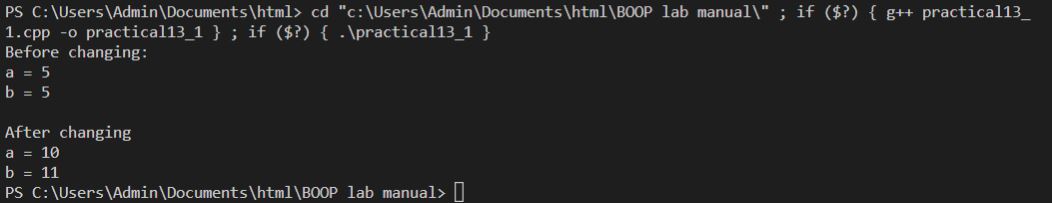
**{**

**\*n1 = 10;**

**\*n2 = 11;**

**}**

**Output:**

****

**Q14)** Develop a program using runtime polymorphism

**🡪**

**#include <iostream>**

**using namespace std;**

**class Polygon {**

**public:**

**virtual void show() {**

**cout<<"It is a polygon"<<endl;**

**}**

**};**

**class Hexagon : public Polygon {**

**public:**

**void show() {**

**cout<<"Hexagon is a 6 sided polygon"<<endl;**

**}**

**};**

**class Pentagon : public Polygon {**

**public:**

**void show() {**

**cout<<"Pentagon is a 5 sided polygon"<<endl;**

**}**

**};**

**int main() {**

**Polygon \*P;**

**Hexagon h;**

**Pentagon p;**

**P = &h;**

**P->show();**

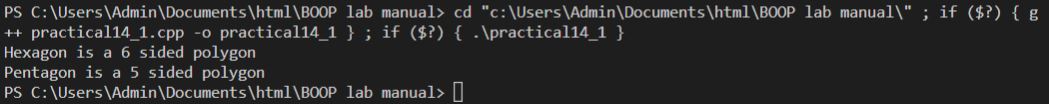
**P = &p;**

**P->show();**

**return 0;**

**}**

**Output:**

****

**Q15)** Develop at least 2 programs using file operations

**Program 1**

🡪

**#include<iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**fstream new\_file;**

**new\_file.open("new\_file", ios::out);**

**if (!new\_file)**

**{**

**cout << "File creation failed";**

**}**

**else**

**{**

**cout << "New file created"<<endl;**

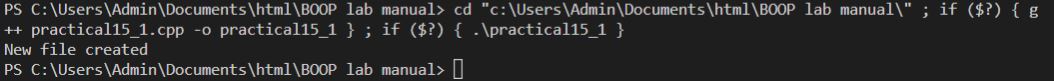
**new\_file.close(); // Step 4: Closing file**

**}**

**return 0;**

**}**

**Output:**

****

**Program 2**

**🡪**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**fstream new\_file;**

**new\_file.open("new\_file\_write.txt", ios::out);**

**if (!new\_file)**

**{**

**cout << "File creation failed";**

**}**

**else**

**{**

**cout << "New file created"<<endl;**

**new\_file << "Learning File handling"<<endl; //Writing to file**

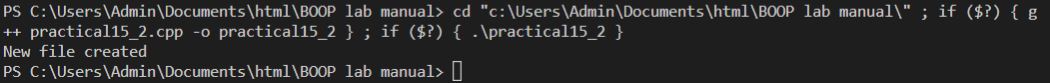
**new\_file.close();**

**}**

**return 0;**

**}**

**Output:**

****