

**C++ PROGRAMMING LAB**

**PRACTICAL FILE**

**Submitted to: Submitted by:**

**Dr. Sapna Zain Md. Sahil**

Enrollment no. = 2021-301-109

Section = B

BCA 2nd year 3rd semester

Assistant Professor

Deptt. of Computer Science &amp; Engg.

Jamia Hamdard, New Delhi, India

**Department Of Computer Science and Engineering**

**School of Engineering Sciences and Technology**

**Jamia Hamdard, New Delhi -110062**

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | | | |
| **S.NO** | **PROGRAM** | **PAGE NO.** | **SIGNATURE OF TEACHER** |
| 1. | Write a program in C++ to accept student name, department and course and display the details. | **4** |  |
| **2.** | Write a program in C++ to accept the age of the person and display whether the person is eligible to vote. | **5** |  |
| **3.** | Write a program in C++ to accept the weekday and display the day name using an accept switch case statement. | **6** |  |
| **4.** | Write a program in C++ to accept student name and percentage and display the grade under the following condition  Percentage Grade  >90 A+  >80 A  >70 B+  >60 B  >50 C+  >40 C  >30 D  >20 F | **7** |  |
| **5.** | Write a program in C++ to accept the destination and display the ticket charges: -  Destination Ticket Charges  Bombay 12000  Nagpur 10000  Pune 9000  Aurangabad 12000 | **8** |  |
| **6.** | Write a program in C++ to accept a number and display the calculated sum of individual digits. | **10** |  |
| **7.** | Write a program in C++ to accept a number and display its factorial. | **11** |  |
| **8.** | Write a program in C++ to display Fibonacci series: -- 0,1,1,2,3,5,8,13,21......101. | **12** |  |
| **9.** | Write a program in C++ to accept and number and display whether it is palindrome or not. | **12** |  |
| **10.** | Write a program in C++ to display the sum of even and odd numbers within the range 10 -100. | **13** |  |
| **11.** | Write a C++ program to display names, roll no and grades of 3 students who have appeared in examination. declare the class of name, roll no and grade. create an array of class objects. read and display the contents of array. | **15** |  |
| **12.** | Write a c++ program to declare struct, initialize and display contents of member variables. | **18** |  |
| **13.** | Write a C++ program to declare a class, declare pointer to class, initialize and display contents of class member. | **19** |  |
| **14.** | Write a C++ program to declare a class, declare pointer to class, initialize and display contents of class member. | **21** |  |
| **15.** | Write a C++ program to read the data of an employee and compute net salary of each employee (DA=52%of basic and income tax (IT)=30% of gross salary. | **23** |  |
| **16.** | Write a Program using class to process Shopping List for a Departmental Store. The list includes details such as the Code No and Price of each item and perform the operations like Adding, Deleting Items to the list and Printing the Total value of an Order. | **26** |  |
| **17.** | Write a program to maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result. | **29** |  |
| **18.** | Write a program to design a class representing complex numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading. | **31** |  |
| **19.** | Write a C++ Program to illustrate default constructor, parameterized constructor and copy constructors. | **33** |  |
| **20.** | Write a program to demonstrate the i) Operator Overloading ii) Function Overloading. | **37** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **21.** | Write a C++ program to show the single inheritance usig public derivation. |  |  |
| **22.** | Write a C++ program which show the multiple inheritance. |  |  |
| **23.** | Write a program to show the hierarchical inheritance. |  |  |
| **24.** | Write a program to show the simple hybrid inheritance. |  |  |
| **25.** | Program to show hybrid inheritance using virtual base classes. |  |  |
| **26.** | Write a C++ program to find factorial of a number using recursion. |  |  |
| **27.** | Write a C++ program using friend function used to print the length of a box. |  |  |
| **28.** | Write a C++ program to demonstrate when the function is friendly to two classes. |  |  |
| **29.** | Write a C++ program using a friend class. |  |  |
| **30.** | Write a C++ program to show the use of virtual function. |  |  |
| **31.** | Create two classes DM and DB which stores the value of distance, DM stores distances in metres and centimetres and DB in feet and inches. Write a program that can read values for the class objects and one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be  DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimetres depending on  the object on display. |  |  |
| **32.** | A book shop maintains the inventory of books that are  being sold at the shop. The list includes details such as  author, title, price, publisher and stock position. Whenever  a customer wants a book, the sales person inputs a title and author and the system searches the list and displays whether it is available or not. If it is then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed, otherwise the message “Required copies not in stock “is displayed. |  |  |
| **33.** | Create a base class shape (). Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add the base class, a member function get\_data() to initialise base class data members and another member function display\_area() as virtual function and redefine this function in the derived classes to suit their requirements .Using these three classes,design a program that will accept dimensions of a triangle or a rectangle interactively and display the area . |  |  |
| **34.** | Write a C++ program to design a student class representing student roll-no, test class (derived class of student)representing the scores of the student in various subjects and sports class. The sports and test class should be inherited by a result class having the functionality to add the scores and display the final result for a student. |  |  |
| **35.** | Write a Program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha & beta. |  |  |

**Q1 Write a program in C++  to accept student name, department and course and display the details.**

**Program:**

#include<iostream>

using namespace std;

int main()

{

char name[25],course[10],department[20];

cout<<"Enter the name:-";

cin>>name;

cout<<"Enter Course Name:-";

cin>>course;

cout<<"Enter Deparment Name:-";

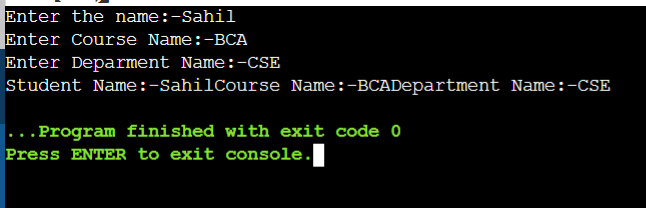
cin>>department;

cout<<"Student Name:-"<<name<<"Course Name:-"<<course<<"Department Name:-"<<department;

return 0;

}

**OUTPUT:**

****

**Q2: Write a program in C++ to accept the age of the person and display whether the person is eligible to vote.**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int age;

cout<<"Enter your Age:";

cin>>age;

if(0>age || age>100)

cout<<"\nEnter a Valid Input:\n";

else if(age>=18)

cout<<"\nEligible for Vote:\n";

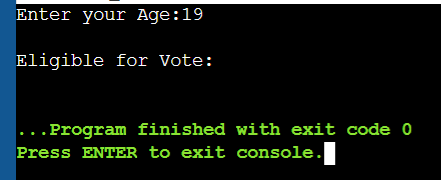
else

cout<<"\nNot Eligible for Vote:\n";

return 0;

}

**OUTPUT:**

****

**Q3: Write a program in C++ to accept the weekday and display the day name using an accept switch case statement.**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int num;

cout<<"Enter a Number Between 1 - 7 :";

cin>>num;

switch(num)

{

case 1:

cout<<"Monday"<<endl;

break;

case 2:

cout<<"Tuesday"<<endl;

break;

case 3:

cout<<"Wednesday"<<endl;

break;

case 4:

cout<<"Thursday"<<endl;

break;

case 5:

cout<<"Friday"<<endl;

break;

case 6:

cout<<"Saturday"<<endl;

break;

case 7:

cout<<"Sunday"<<endl;

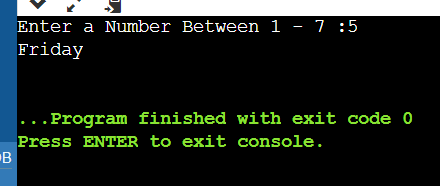
break;

}

return 0;

}

**Output:**

****

**Q4: Write a program in C++ to accept student name and percentage and display the grade under the following condition-  
Percentage            Grade  
 >90                  A+  
 >80                      A  
 >70                    B+  
 >60                      B  
 >50                      C+  
 >40                      C  
 >30                      D  
>20                      F**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int per;

char name[20];

cout<<"Enter the name of the Student:-";

cin>>name;

cout<<"Enter Percentage:-";

cin>>per;

if(per>100 || 0>per)

cout<<"\nEnter a Valid option\n";

else if(per>90)

cout<<name<<" Get A+";

else if(per>80)

cout<<name<<" Get A";

else if(per>70)

cout<<name<<" Get B+";

else if(per>60)

cout<<name<<" Get B";

else if(per>50)

cout<<name<<" Get C+";

else if(per>40)

cout<<name<<" Get C";

else if(per>30)

cout<<name<<" Get D";

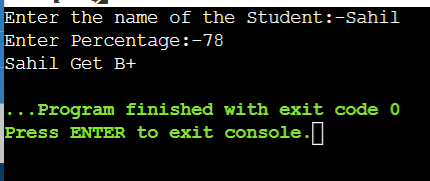
else if(per>20)

cout<<name<<" Get f";

return 0;

}

**Output:**

****

**Q5: Write a program in C++ to accept the destination and display the ticket charges:-  
Destination           Ticket Charges**

**Bombay     12000 Nagpur                      10000 Pune                        9000 Aurangabad              12000**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int choice;

cout<<"Where do you want to go\n -----Menu----- \n 1 for Bombay\n 2 for Nagpur\n 3 for Pune\n 4 for Aurangabad\n Choice:---";

cin>>choice;

if(choice==1)

cout<<"Bombay - 12,000 INR";

else if(choice==2)

cout<<"Nagpur - 10,000 INR";

else if(choice==3)

cout<<"Pune - 9,000 INR";

else if(choice==4)

cout<<"Aurangabad - 12,000 INR";

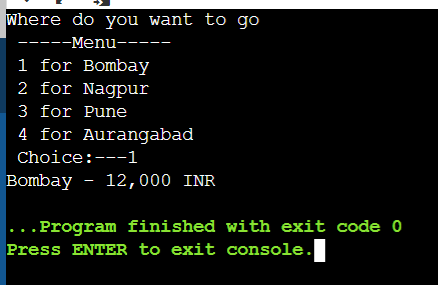
else

cout<<"---Enter a Valid Option---";

return 0;

}

**Output:**

****

**Q6: Write a program in C++ to accept a number and display the calculated sum of individual digits.**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int x,sum=0,y;

cout<<"Enter a number: ";

cin>>y;

while(y>0)

{

x=y%10;

sum=sum+x;

y=y/10;

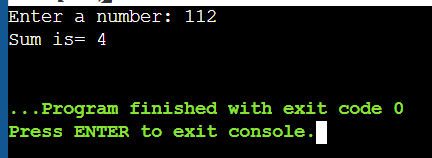
}

cout<<"Sum is= "<<sum<<endl;

return 0;

}

**Output:**



**Q7: Write a program in C++ to accept a number and display its factorial.**

**Program:**

#include <iostream>

using namespace std;

int main()

{

int i,factorial=1,num;

cout<<"Enter any Number: ";

cin>>num;

for(i=1;i<=num;i++)

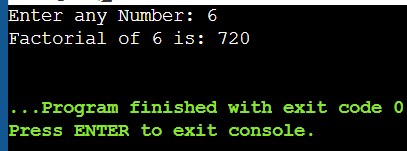
factorial=factorial\*i;

cout<<"Factorial of " <<num<<" is: "<<factorial<<endl;

return 0;

}

**Output:**

****

**Q8: Write a program in C++ to display fibonacci series :--  0,1,1,2,3,5,8,13,21......101.**

**Program:**

#include <iostream>

using namespace std;

int main() {

int n, t1 = 0, t2 = 1, nextTerm = 0;

cout << "Enter the number of terms: ";

cin >> n;

cout << "Fibonacci Series: ";

for (int i = 1; i <= n; ++i)

{

if(i == 1) {

cout << t1 << ", ";

continue;

}

if(i == 2) {

cout << t2 << ", ";

continue;

}

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

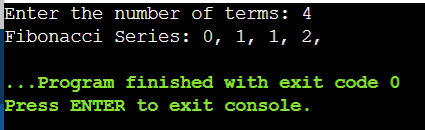
cout << nextTerm << ", ";

}

return 0;

}

**Output:**

****

**Q9: Write a program in C++ to accept and number and display whether it is palindrome or not.**

**Program:**

#include <iostream>

using namespace std;

int main()

{

int n,r,sum=0,temp;

cout<<"Enter the Number=";

cin>>n;

temp=n;

while(n>0)

{

r=n%10;

sum=(sum\*10)+r;

n=n/10;

}

if(temp==sum)

cout<<"Number is Palindrome.";

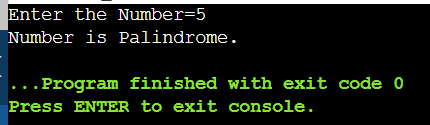
else

cout<<"Number is not Palindrome.";

return 0;

}

**Output:**

****

**Q10: Write a program in C++ to display the sum of even and odd numbers within the range 10 -100.**

**Program:**

#include<iostream>

using namespace std;

int main()

{

int number, minimum, maximum, evenSum = 0, oddSum = 0;

cout << "\n Please Enter the Minimum Limit for Even & Odd Numbers = ";

cin >> minimum;

cout << "\n Please Enter the Maximum Limit for Even & Odd Numbers = ";

cin >> maximum;

number=minimum;

for(minimum = 1; number <= maximum; number++)

{

if ( number % 2 == 0 )

{

evenSum = evenSum + number;

}

else

{

oddSum = oddSum + number;

}

}

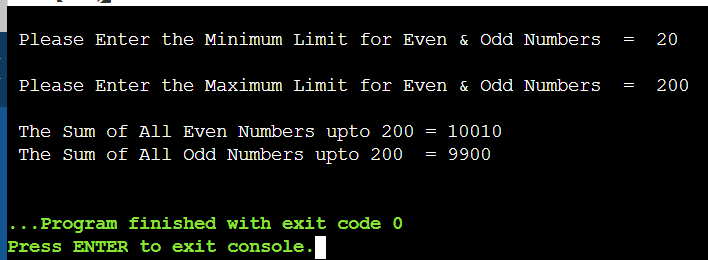
cout << "\n The Sum of All Even Numbers upto " << maximum << " = " << evenSum;

cout << "\n The Sum of All Odd Numbers upto " << maximum << " = " << oddSum<<endl;

return 0;

}

**Output:**

****

**11.Write a C++ program to display names, roll no and grades of 3 students who have appeared in examination. declare the class of name, roll no and grade. create an array of class objects. read and display the contents of array.**

Program:

#include<iostream>

using namespace std;

class student

{

char name[50];

int rollno;

int total;

public:

void readDetails(void); //member function to get students Details

void GradeCal(void); //member function to display voidGrade

void printDetails(void); //member function to print students print Details

};

//member function definition outside the class

void student :: readDetails(void)

{

cout<<"Enter name: ";

cin>>name;

cout<<"\n Enter Roll no. :";

cin>>rollno;

cout<<"\n Enter the total marks (out to 500): ";

cin>>total;

}

void student :: GradeCal(void)

{

float percentage;

percentage = (float)total/500.0\*100.0;

cout<<percentage<<"%\n";

if(percentage>=75.0){

cout<<"Grade : A \n";

}

else if(percentage>=60.0 && percentage<75.0){

cout<<"Grade : B \n";

}

else if(percentage>=40.0 && percentage<60){

cout<<"Grade : C \n";

}

else{

cout<<"Grade :D \n";

}

}

//member function defination outside the class

void student :: printDetails(void)

{

cout<<"Student Details: \n";

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

cout<<"Rollnumber: "<<rollno<<endl;

cout<<"Total: "<<total<<endl;

}

int main()

{

student std[10]; //array of objects creation

int n,i;

cout<<"Enter total nubmer of students: ";

cin>>n;

for(i=0; i<n; i++){

cout<<"Enter the details of students: "<<i+1<<endl;

std[i].readDetails();

}

for(i=0; i<n; i++){

cout<<"Details od student "<<i+1<<endl;

std[i].printDetails();

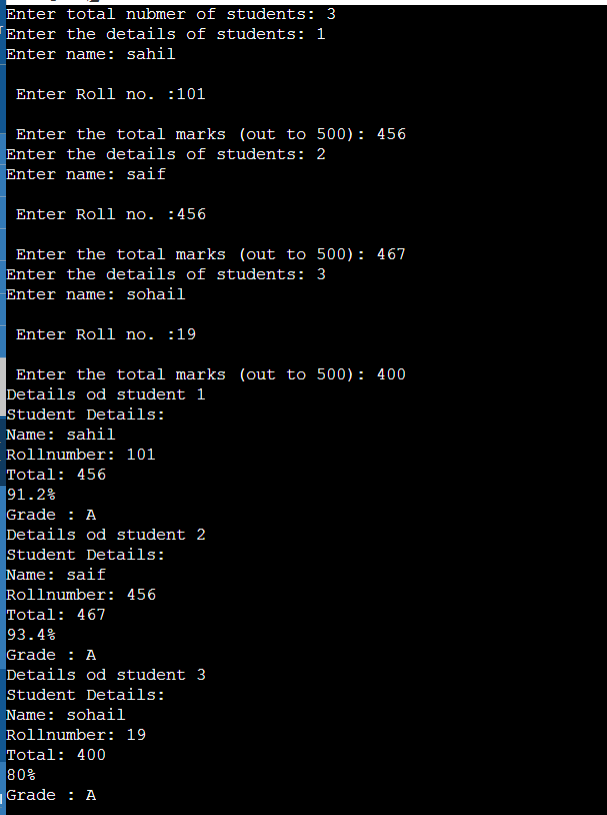
std[i].GradeCal();

}

return 0;

}

Output:



**12. Write a c++ program to declare struct, initialize and display contents of member variables.**

Program:

#include<iostream>

using namespace std;

struct college\_info{

char college\_name[20];

int college\_code;

char dept\_name[50];

int intake;

};

int main(){

struct college\_info college;

cout<<"\n \*\*\*\*\*\*\*\*\*Enter the college information\*\*\*\*\*\*\*\*\n";

cout<<"Enter the name of college: ";

cin>>college.college\_name;

cout<<"Enter the collage code: ";

cin>>college.college\_code;

cout<<"Name of the department: ";

cin>>college.dept\_name;

cout<<"Enter the Department intake: ";

cin>>college.intake;

cout<<"\n\n\*\*\*\*\*\*\*\*College Information\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"Name of the college: "<<college.college\_name<<endl;

cout<<"College Code: "<<college.college\_code<<endl;

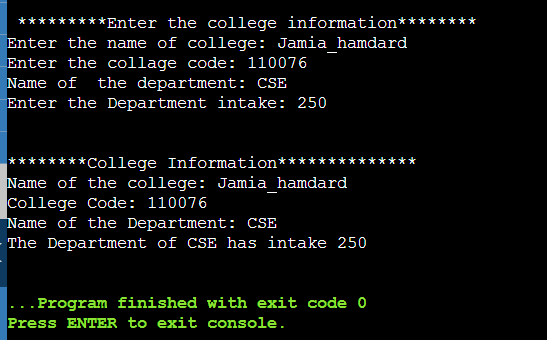
cout<<"Name of the Department: "<<college.dept\_name<<endl;

cout<<"The Department of "<<college.dept\_name<<" has intake "<<college.intake<<endl;

return 0;

}

**Output:**

****

**13.** **Write a C++ program to declare a class, declare pointer to class, initialize and display contents of class member.**

Program:

#include <iostream>

using namespace std;

class RectangleTest{

public:

int length, breadth;

public:

void initialize(int len, int bre){

length = len;

breadth = bre;

}

int getArea(){

return length\*breadth;

}

void display(){

int area = getArea();

cout<<"\n\*\*\* Rectangle Information \*\*\*\n";

cout<<"Length = "<<length;

cout<<"\nBreadth = "<<breadth;

cout<<"\nArea = "<<area;

cout<<"\n-----------------------------\n";

}

};

int main()

{

RectangleTest rect, \*class\_ptr;

class\_ptr = &rect;

class\_ptr->initialize(10,5);

class\_ptr->display()

class\_ptr->length = 2;

class\_ptr->breadth = 3;

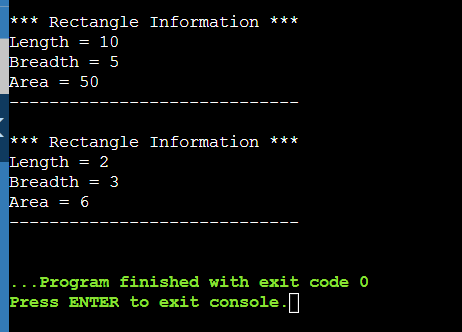
class\_ptr->initialize(class\_ptr->length,class\_ptr->breadth);

class\_ptr->display();

return 0;

}

**Output:**



**14. Given that an employee class contains following members, data members, employee number, employee name, basic, DA, IT, net salary and print data members.**

Program:

#include<iostream>

using namespace std;

class employee

{

int emp\_number;

char emp\_name[20];

float emp\_basic;

float emp\_da;

float emp\_it;

float emp\_net\_salary;

public:

void get\_emp\_details();

float find\_net\_salary(float basic, float da, float it);

void show\_emp\_details();

};

void employee :: get\_emp\_details()

{

cout<<"Enter the Employee Number; "<<endl;

cin>>emp\_number;

cout<<"Enter the employee Name: "<<endl;

cin>>emp\_name;

cout<<"Enter the employee Basic: "<<endl;

cin>>emp\_basic;

cout<<"Enter the employee DA: "<<endl;

cin>>emp\_da;

cout<<"Enter the employree Income Tax: "<<endl;

cin>>emp\_it;

}

float employee :: find\_net\_salary(float basic, float da, float it)

{

return(basic+da)-it;

}

void employee :: show\_emp\_details()

{

cout<<"\n\n\*\*\*\*\*\*\*Details of Employees\*\*\*\*\*\*\*\*\n\n";

cout<<"Employee Name : " <<emp\_name<<endl;

cout<<"Employee Number : " <<emp\_number<<endl;

cout<<"Basic Salary : " <<emp\_basic<<endl;

cout<<"Employee DA : " <<emp\_da<<endl;

cout<<"Income Tax : " <<emp\_it<<endl;

cout<<"Net Salary : " <<find\_net\_salary(emp\_basic, emp\_da, emp\_it)<<endl;

cout<<"\n\n------------------------------------------------------\n\n";

}

int main()

{

employee emp;

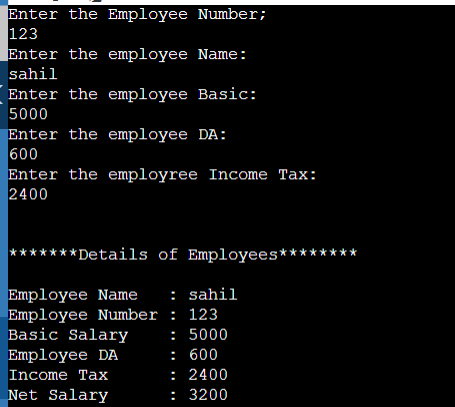
emp.get\_emp\_details();

emp.show\_emp\_details();

return 0;

}

**Output:**

****

**15.Write a C++ program to read the data of an employee and compute net salary of each employee (DA=52%of basic and income tax (IT)=30% of gross salary.**

Program:

#include<iostream>

using namespace std;

class employee

{

char emp\_name[20];

int emp\_num;

float basic, da, it, gross\_salary, net\_salary;

public:

void read\_emp\_details(int count)

{

cout<<"\n\n \*\*\*\*\*\*Enter Employees "<<count<<" Details\*\*\*\*\*\*\*\n";

cout<<"Enter Employee number: ";

cin>>emp\_num;

cout<<"Enter the Employee name: ";

cin>>emp\_name;

cout<<"Enter The Basic Salary: ";

cin>>basic;

cout<<"\n -----Employee "<<count<<" Details are saved------\n";

}

float find\_net\_salary()

{

da = basic \* 0.50;

gross\_salary = basic + da;

it = gross\_salary \* 0.30;

net\_salary = (basic + da) - it;

return net\_salary;

}

void display\_emp\_details(int count)

{

cout<<"\n\n\*\*\*\*\*\*Employee "<<count<<" Details\*\*\*\*\*\*\*\*\*\n";

cout<<"\n Employee Number : "<<emp\_num;

cout<<"\n Employee Name : "<<emp\_name;

cout<<"\n Net Salary : "<<net\_salary;

cout<<"\n------------------------------------\n\n";

}

};

int main()

{

employee emp[100];

int number\_of\_emp, count;

cout<<"\n Please enter the number of Employees(max 100): ";

cin>>number\_of\_emp;

for(count=0; count<number\_of\_emp; count++)

{

emp[count].read\_emp\_details(count+1);

}

for(count=0; count<number\_of\_emp; count++)

{

emp[count].find\_net\_salary();

}

for(count=0; count<number\_of\_emp; count++)

{

emp[count].display\_emp\_details(count+1);

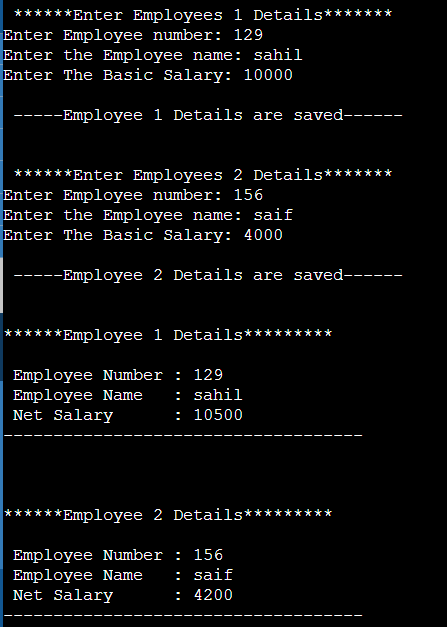
}

cout<<"\n Press any key to close!!";

return 0;

}

**Output:**



**16. Write a Program using class to process Shopping List for a Departmental Store. The list includes details such as the Code No and Price of each item and perform the operations like Adding, Deleting Items to the list and Printing the Total value of an Order.**

Program:

#include<iostream>

using namespace std;

const int m = 50;

class items

{

int itemCode[m];

float itemPrice[m];

int count;

public:

void CNT(void){

count = 0;

}

void getitem(void);

void displaySum(void);

void remove(void);

void displayItems(void);

};

void items :: getitem(void)

{

cout<<"Enter Item Code: ";

cin>>itemCode[count];

cout<<"Enter item cost: ";

cin>>itemPrice[count];

count++;

}

void items :: displaySum(void)

{

float sum = 0;

for(int i=0; i<count; i++){

sum = sum + itemPrice[i];

cout<<"\n Total value: "<<sum<<endl;

}

}

void items :: remove(void)

{

int a;

cout<<"Enter item Code: ";

cin>>a;

for(int i=0; i<count; i++){

if(itemCode[i] == a){

itemPrice[i] = 0;

}

}

}

void items :: displayItems(void)

{

cout<<"\n COde \t Price \n ";

for(int i=0; i<count; i++)

{

cout<<itemCode[i]<<"\t"<<itemPrice[i]<<"\n ";

}

cout<<"\n";

}

int main()

{

items order;

order.CNT();

int x;

do{

cout<<"\n You can do the following: "<<" Enter the appropriate number \n";

cout<<"\n1. Add an item: ";

cout<<"\n2. Display total value ";

cout<<"\n3. Delete an item ";

cout<<"\n4. Delete all items ";

cout<<"\n Quit ";

cout<<"\n What is your option: ";

cin>>x;

switch(x)

{

case 1:

order.getitem();

break;

case 2:

order.displaySum();

break;

case 3:

order.remove();

break;

case 4:

order.displayItems();

break;

case 5:

break;

case 6:

cout<<"Error in Input: try again\n";

}

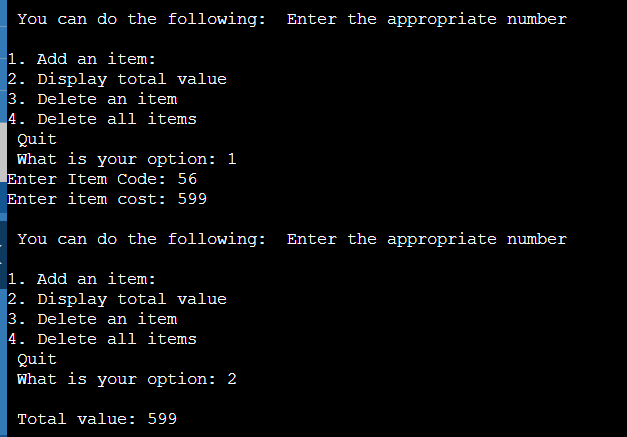
}

while(x != 5);

return 0;

}

Output:



**17. Write a program to maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result.**

Program:

#include<iostream>

using namespace std;

class Person

{

char name[20];

int age;

public:

void getData()

{

cout<<"Enter Name : ";

cin>>name;

cout<<"Enter Age : ";

cin>>age;

}

void putData()

{

cout<<"\n\n Details About Elder Person as follows: ";

cout<<"\n Name = "<<name<<endl;

cout<<" Age = "<<age<<endl;

}

Person&Compare(Person &p1)

{

if(p1.age>this->age)

return p1;

return \*this;

}

};

int main()

{

Person a,b,c;

a.getData();

b.getData();

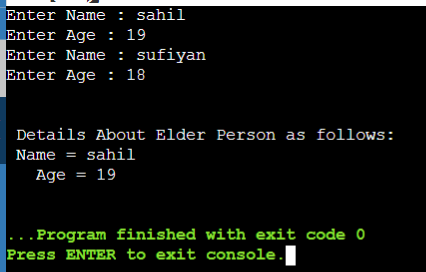
c = a.Compare(b);

c.putData();

return 0;

}

Output:



**18.Write a program to design a class representing complex numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading**.

Program:  
#include<iostream>

using namespace std;

class Complex

{

private:

int real, imag;

public:

Complex ()

{

}

Complex(int r, int i)

{

real = r;

imag = i;

}

Complex operator \*(Complex b)

{

Complex c;

c.real = this -> real \* b.real;

c.imag = this -> imag \* b.imag;

return c;

}

Complex operator +(Complex b)

{

Complex c;

c.real = this -> real + b.real;

c.imag = this -> imag + b.imag;

return c;

}

void display()

{

cout<<"\n"<< real<<"\n" <<imag<<endl;

}

};

int main()

{

Complex c1(12, 10), c2(4, 5), c3, c4;

c3 = c1+c2;

c4 = c1\*c2;

cout<<"THe addition of two complex number is : ";

c3.display();

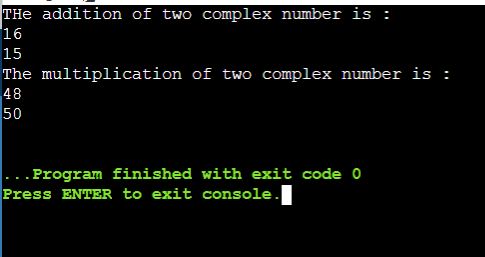
cout<<"The multiplication of two complex number is : ";

c4.display();

return 0;

}

**Output:**

****

**19.Write a C++ Program to illustrate default constructor, parameterized constructor and copy constructors.**

**DEFAULT CONSTRUCTOR:**

Program:

#include<iostream>

using namespace std;

class student

{

int roll;

float marks;

public:

student()

{

cout<<"Enter the Roll Number: ";

cin>>roll;

cout<<"Enter the Marks: ";

cin>>marks;

}

void display()

{

cout<<"Roll no. = "<<roll<<endl;

cout<<"Marks = "<<marks<<endl;

}

};

int main()

{

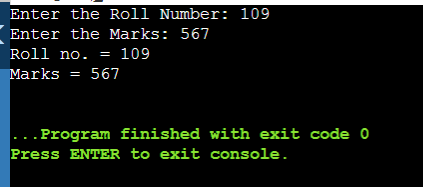
student s1;

s1.display();

return 0;

}

Output:



**PARAMETARIZED CONSTRUCTOR:**

Program:

#include<iostream>

using namespace std;

class pointer

{

private:

int x, y;

public:

pointer(int x1, int y1)

{

x = x1;

y = y1;

}

int get\_x()

{

return x;

}

int get\_y()

{

return y;

}

};

int main()

{

pointer p1(10, 15);

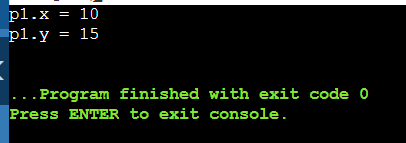
cout<<"p1.x = "<<p1.get\_x()<<endl;

cout<<"p1.y = "<<p1.get\_y()<<endl;

return 0;

}

**Output:**



**COPY CONSTRUCTOR:**

Program:

#include <iostream>

using namespace std;

class Wall {

private:

double length;

double height;

public:

Wall(double len, double hgt) {

length = len;

height = hgt;

}

Wall(Wall &obj) {

length = obj.length;

height = obj.height;

}

double calculateArea() {

return length \* height;

}

};

int main() {

Wall wall1(10.5, 8.6);

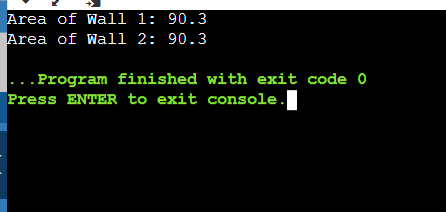
Wall wall2 = wall1;

cout << "Area of Wall 1: " << wall1.calculateArea() << endl;

cout << "Area of Wall 2: " << wall2.calculateArea();

return 0;

}



**20. Write a program to demonstrate the i) Operator Overloading ii) Function Overloading.**

**i) Operator Overloading**

Program:

#include <iostream>

using namespace std;

class Test

{

private:

int num;

public:

Test(): num(8){}

void operator ++()

{

num = num+2;

}

void Print()

{

cout<<"The Count is: "<<num;

}

};

int main()

{

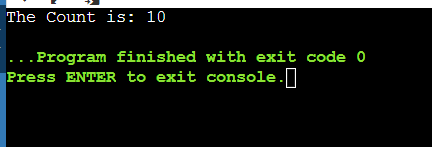
Test tt;

++tt; // calling of a function "void operator ++()"

tt.Print();

return 0;

}

****

**ii) Function Overloading.**

Program:

#include <iostream>

using namespace std;

class Calculation

{

public:

static int add(int a,int b)

{

return a + b;

}

static int add(int a, int b, int c)

{

return a + b + c;

}

};

int main(void) {

Calculation C; // class object declaration.

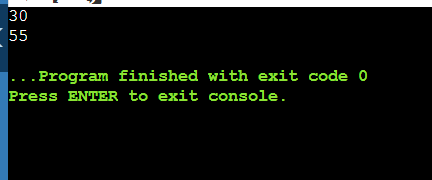
cout<<C.add(10, 20)<<endl;

cout<<C.add(12, 20, 23);

return 0;

}

Output:



**20. Write a C++ program to show the single inheritance usig public derivation**.

Program:

#include<iostream>

using namespace std;

class A{

int a=4;

int b=5;

public:

int mul()

{

int c=a\*b;

return c;

}

};

class B : private A{

public:

void display()

{

int result = mul();

cout<<"Multiplication of A and B is : "<<result<<endl;

}

};

int main(){

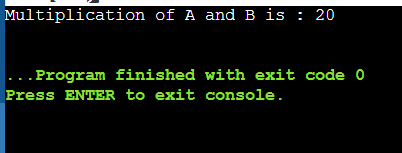
B b;

b.display();

return 0;

}

**Output:**

****

**22.Write a C++ program which show the multiple inheritance**.

Program:  
#include<iostream>

using namespace std;

class A{

protected:

int a;

public:

void get\_a(int n){

a=n;

}

};

class B{

protected:

int b;

public:

void get\_b(int n){

b=n;

}

};

class C : public A, public B{

public:

void display()

{

cout<<"THe value of A is : "<<a<<endl;

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

cout<<"THe addition of A and B is : "<<a+b<<endl;

}

};

int main(){

C c;

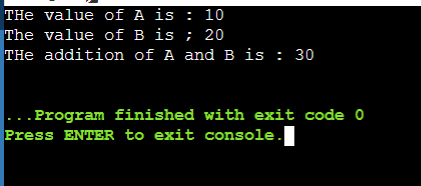
c.get\_a(10);

c.get\_b(20);

c.display();

}

Output:



**23.Write a program to show the hierarchical inheritance.**

**Program:**

#include<iostream>

using namespace std;

class Shape{

public:

int a;

int b;

void get\_data(int n, int m)

{

a=n;

b=m;

}

};

class Rectrangle : public Shape{

public:

int rec\_area()

{

int result= a\*b;

return result;

}

};

class triangle : public Shape

{

public:

int triangle\_area()

{

float result = 0.5\*a\*b;

return result;

}

};

int main(){

Rectrangle r;

triangle t;

int length,breadth,base,height;

cout<<"Enter the length of rectrangle: ";

cin>>length;

cout<<"Enter the breadth of rectrangle: ";

cin>>breadth;

r.get\_data(length, breadth);

int m = r.rec\_area();

cout<<"Area of rectrangle is : "<<m<<endl;

cout<<"Enter the base of triangle: ";

cin>>base;

cout<<"Enter the height of triangle: ";

cin>>height;

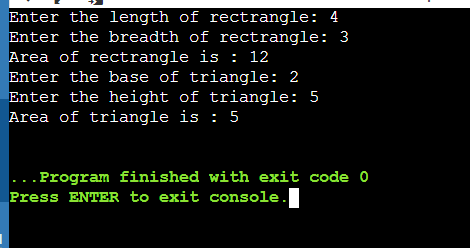
t.get\_data(base, height);

float n = t.triangle\_area();

cout<<"Area of triangle is : "<<n<<endl;

return 0;

}

**Output:  
**

**24.Write a program to show the simple hybrid inheritance.**

**Program:**

#include<iostream>

using namespace std;

class A{

protected:

int a;

public:

void get\_a()

{

cout<<"Enter the value of A: ";

cin>>a;

}

};

class B : public A{

protected:

int b;

public:

void get\_b(){

cout<<"Enter the value of B : ";

cin>>b;

}

};

class C{

protected:

int c;

public:

void get\_c()

{

cout<<"Enter the value of C : ";

cin>>c;

}

};

class D : public B, public C{

protected:

int d;

public:

void mul()

{

get\_a();

get\_b();

get\_c();

cout<<"Multiplication of A,B,C is : "<<a\*b\*c<<endl;

}

} ;

int main()

{

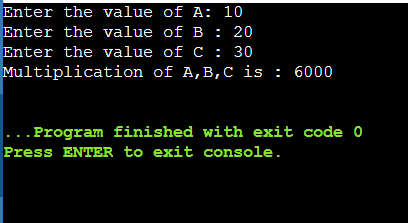
D d;

d.mul();

return 0;

}

**Output:**

****

**25.Program to show hybrid inheritance using virtual base classes.**

**Program:**

#include<iostream>

using namespace std;

class students{

protected:

int roll\_no;

public:

void set\_number(int a){

roll\_no=a;

}

void print\_number(void){

cout<<"Your roll number is : "<<roll\_no<<endl;

}

};

class test : virtual public students{

protected:

float maths,physics,bio;

public:

void set\_marks(float m1, float m2, float m3){

maths=m1;

physics=m2;

bio=m3;

}

void print\_marks(void){

cout<<"Your result is here: "<<endl;

cout<<"Maths: "<<maths<<endl;

cout<<"Physics: "<<physics<<endl;

cout<<"Bio : "<<bio<<endl;

}

};

class sports : virtual public students{

protected:

float score;

public:

void set\_score(float sc){

score=sc;

}

void print\_score(void){

cout<<"Your Pt score is : "<<score<<endl;

}

};

class result :virtual public test ,virtual public sports{

private:

float total;

public:

void display(void){

total = maths+physics+bio+score;

print\_number();

print\_marks();

print\_score();

cout<<"Your total score is: "<<total<<endl;

}

};

int main(){

result sahil;

sahil.set\_number(450);

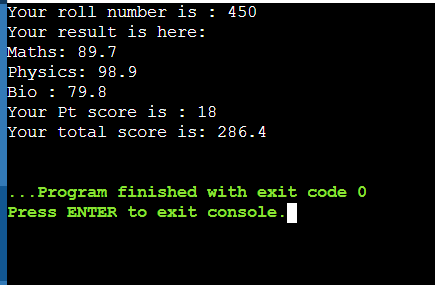
sahil.set\_marks(89.7, 98.9, 79.8);

sahil.set\_score(18);

sahil.display();

return 0;

}  
**Output:**

****

**26.Write a C++ program to find factorial of a number using recursion.**

**Program:**

#include<iostream>

using namespace std;

int factorial(int n){

if(n>1){

return n\*factorial(n-1);

}

else{

return 1;

}

}

int main(){

int n;

cout<<"Enter the positive integer: ";

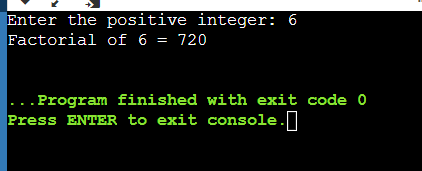
cin>>n;

cout<<"Factorial of "<<n<<" = "<<factorial(n)<<endl;

return 0;

}

**Output:**

****

**27. Write a C++ program using friend function used to print the length of a box.**

**Program:**

#include<iostream>

using namespace std;

class Box{

private:

int length;

public:

Box():length(0){}

friend int printLength(Box);

};

int printLength(Box b){

b.length += 10;

return b.length;

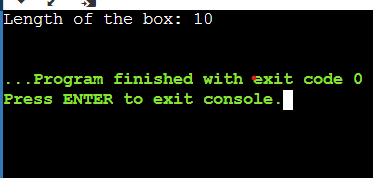
}

int main(){

Box b;

cout<<"Length of the box: "<<printLength(b)<<endl;

} **Output:**

****

**28.Write a C++ program to demonstrate when the function is friendly to two classes.**

**Program:**

#include<iostream>

using namespace std;

class B;

class A{

int x;

public:

void set\_data(int i){

x=i;

}

friend void min(A,B);

};

class B{

int y;

public:

void set\_data(int i){

y=i;

}

friend void min(A,B);

};

void min(A a, B b){

if(a.x<=b.y){

cout<<a.x<<endl;

}

else{

cout<<b.y<<endl;

}

}

int main(){

A a;

B b;

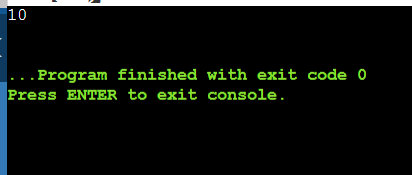
a.setdata(10);

b.set\_data(20);

min(a,b);

return 0;

} **Output:**

****

**29.Write a C++ program using a friend class.**

**Program:**

#include<iostream>

using namespace std;

class A{

int x=5;

friend class B;

};

class B{

public:

void display(A &a){

cout<<"The value of x is: "<<a.x<<endl;

}

};

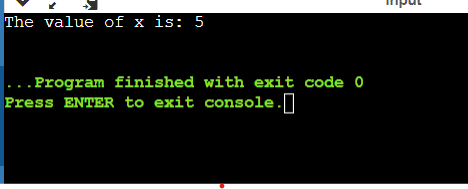
int main(){

A a;

B b;

b.display(a);

return 0;

}  
**Output:  
**

**30. Write a C++ program to show the use of virtual function.**

**Program:**

#include<iostream>

using namespace std;

class Base{

public:

virtual void show()=0;

};

class derived:public Base{

public:

void show()

{

cout<<"Derived class is derived from the base class: "<<endl;

}

};

int main(){

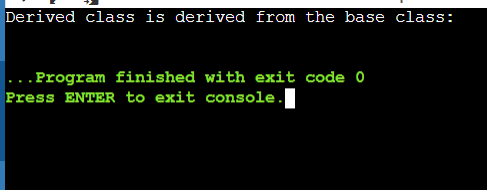
Base \*bptr;

derived d;

bptr = &d;

bptr-> show();

return 0;

}  
**Output:  
**

**31.Create two classes DM and DB which stores the value of distance, DM stores distances in metres and centimetres and DB in feet and inches. Write a program that can read values for the class objects and one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimetres depending on the object on display.**

**Program:**

#include<iostream>

using namespace std;

class DM{

public:

double meter,centimeter;

};

class DB{

public:

double feet,inches;

friend void add(DM,DB);

};

void add(DM dm, DB db)

{

double d1,d2;

cout<<"\n Enter the ditance in meter and centimeter: ";

cin>>dm.meter>>dm.centimeter;

cout<<"\n Enter the distance in feet and inches: ";

cin>>db.feet>>db.inches;

d1=dm.meter+(db.feet)/3.281;

d2=dm.centimeter+(db.inches)\*2.54;

cout<<"\n Meter+feet = "<<d1<<"meter";

cout<<"\n Centimeter + inches = "<<d2<<"centimeter";

}

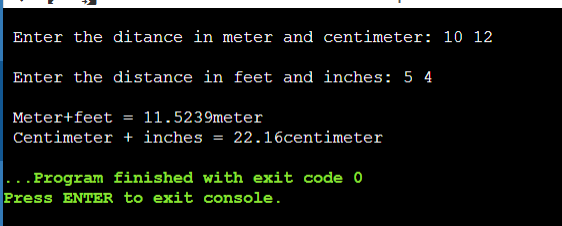
int main(){

DM dm;

DB db;

add(dm,db);

return 0;

}  
**Output:  
**

A book shop maintains the inventory of books that are

being sold at the shop. The list includes details such as

author, title, price, publisher and stock position. Whenever

**32. A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever** **a customer wants a book, the sales person inputs a title and author and the system searches the list and displays whether it is available or not. If it is then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed, otherwise the message “Required copies not in stock “is displayed.**

**Program:**

#include <iostream>

#include <string.h>

#include <stdlib.h>

using namespace std;

class book

{

private:

char \*author, \*title, \*publisher;

float \*price;

int \*stock;

public:

book()

{

author = new char[20];

title = new char[20];

publisher = new char[20];

price = new float;

stock = new int;

}

void feeddata();

void editdata();

void showdata();

int search(char[], char[]);

void buybook();

};

void book::feeddata()

{

cin.ignore();

cout << "\nEnter Author Name: ";

cin.getline(author, 20);

cout << "Enter Title Name: ";

cin.getline(title, 20);

cout << "Enter Publisher Name: ";

cin.getline(publisher, 20);

cout << "Enter Price: ";

cin >> \*price;

cout << "Enter Stock Position: ";

cin >> \*stock;

}

void book::editdata()

{

cout << "\nEnter Author Name: ";

cin.getline(author, 20);

cout << "Enter Title Name: ";

cin.getline(title, 20);

cout << "Enter Publisher Name: ";

cin.getline(publisher, 20);

cout << "Enter Price: ";

cin >> \*price;

cout << "Enter Stock Position: ";

cin >> \*stock;

}

void book::showdata()

{

cout << "\nAuthor Name: " << author;

cout << "\nTitle Name: " << title;

cout << "\nPublisher Name: " << publisher;

cout << "\nPrice: " << \*price;

cout << "\nStock Position: " << \*stock;

}

int book::search(char tbuy[20], char abuy[20])

{

if (strcmp(tbuy, title) == 0 && strcmp(abuy, author) == 0)

return 1;

else

return 0;

}

void book::buybook()

{

int count;

cout << "\nEnter Number Of Books to buy: ";

cin >> count;

if (count <= \*stock)

{

\*stock = \*stock - count;

cout << "\nBooks Bought Sucessfully";

cout << "\nAmount: Rs. " << (\*price) \* count;

}

else

cout << "\nRequired Copies not in Stock";

}

int main()

{

book \*B[20];

int i = 0, r, t, choice;

char titlebuy[20], authorbuy[20];

while (1)

{

cout << "\n\n\t\tMENU"

<< "\n1. Entry of New Book"

<< "\n2. Buy Book"

<< "\n3. Search For Book"

<< "\n4. Edit Details Of Book"

<< "\n5. Exit"

<< "\n\nEnter your Choice: ";

cin >> choice;

switch (choice)

{

case 1:

B[i] = new book;

B[i]->feeddata();

i++;

break;

case 2:

cin.ignore();

cout << "\nEnter Title Of Book: ";

cin.getline(titlebuy, 20);

cout << "Enter Author Of Book: ";

cin.getline(authorbuy, 20);

for (t = 0; t < i; t++)

{

if (B[t]->search(titlebuy, authorbuy))

{

B[t]->buybook();

break;

}

}

if (t == 1)

cout << "\nThis Book is Not in Stock";

break;

case 3:

cin.ignore();

cout << "\nEnter Title Of Book: ";

cin.getline(titlebuy, 20);

cout << "Enter Author Of Book: ";

cin.getline(authorbuy, 20);

for (t = 0; t < i; t++)

{

if (B[t]->search(titlebuy, authorbuy))

{

cout << "\nBook Found Successfully";

B[t]->showdata();

break;

}

}

if (t == i)

cout << "\nThis Book is Not in Stock";

break;

case 4:

cin.ignore();

cout << "\nEnter Title Of Book: ";

cin.getline(titlebuy, 20);

cout << "Enter Author Of Book: ";

cin.getline(authorbuy, 20);

for (t = 0; t < i; t++)

{

if (B[t]->search(titlebuy, authorbuy))

{

cout << "\nBook Found Successfully";

B[t]->editdata();

break;

}

}

if (t == i)

cout << "\nThis Book is Not in Stock";

break;

case 5:

exit(0);

default:

cout << "\nInvalid Choice Entered";

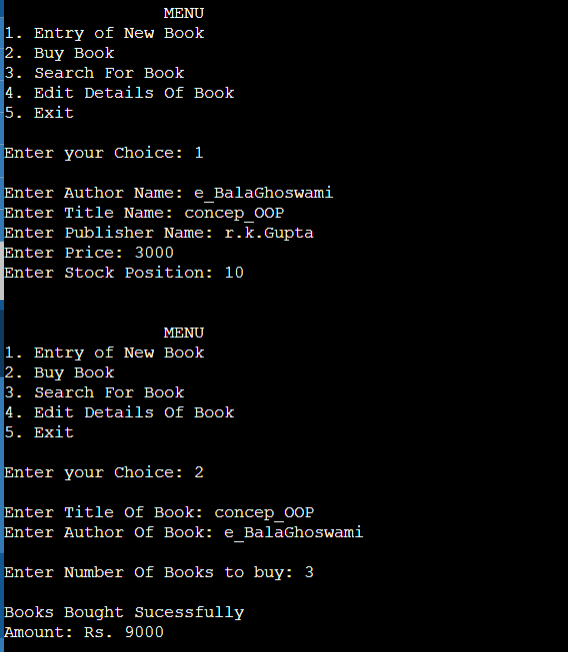
}

}

return 0;

}

**Output:**

****

**33.Create a base class shape (). Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add the base class, a member function get\_data() to initialise base class data members and another member function display\_area() as virtual function and redefine this function in the derived classes to suit their requirements .Using these three classes,design a program that will accept dimensions of a triangle or a rectangle interactively and display the area .**

**Program:**

#include<iostream>

using namespace std;

class Shape{

public:

double height, base;

Shape()

{

height=0;

base=0;

}

void get\_data()

{

cout<<"\n Enter the Height: ";

cin>>height;

cout<<"\n Enter the Base: ";

cin>>base;

}

virtual void display\_area()

{

}

};

class Triangle : public Shape

{

public:

void display\_area()

{

cout<<"\n Area of Triangle = "<<(height\*base)/2<<endl;

}

};

class Rectrangle : public Shape

{

public:

void display\_area()

{

cout<<"\n Area of Rectrangle = "<<height\*base;

}

};

int main()

{

Shape \*s;

Triangle t;

t.get\_data();

s=&t;

s->display\_area();

Rectrangle r;

r.get\_data();

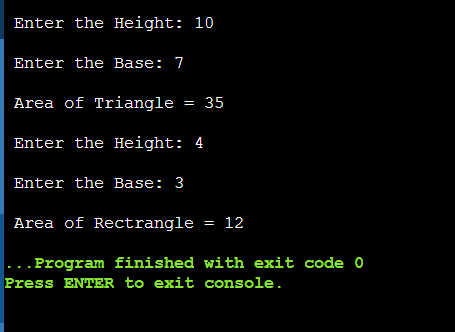
s=&r;

s->display\_area();

return 0;

}

**Output:**

****

**34.Write a C++ program to design a student class representing student roll-no, test class (derived class of student) representing the scores of the student in various subjects and sports class. The sports and test class should be inherited by a result class having the functionality to add the scores and display the final result for a student.**

**Program:**

#include<iostream>

using namespace std;

class student

{

protected:

int roll\_no;

public:

void get\_number(int a)

{

roll\_no=a;

}

void put\_number(void)

{

cout<<"Roll No. "<<roll\_no<<endl;

}

};

class test : public student

{

protected:

float part1,part2;

public:

void get\_marks(float x, float y)

{

part1=x;

part2=y;

}

void put\_marks(void)

{

cout<<"marks Obtained "<<"\n"<<"part1 = "<<part1<<"\n "<<"part2 = "<<part2<<endl;

}

};

class Sports

{

protected:

float score;

public:

void get\_score(float s)

{

score=s;

}

void put\_score(void)

{

cout<<"Sports wt. "<<score<<endl;

}

};

class result : public test, public Sports

{

float total;

public:

void display(void);

};

void result :: display(void)

{

total = part1+part2+score;

put\_number();

put\_marks();

put\_score();

cout<<"Total Score: "<<total<<endl;

}

int main()

{

result S\_1;

S\_1.get\_number(145);

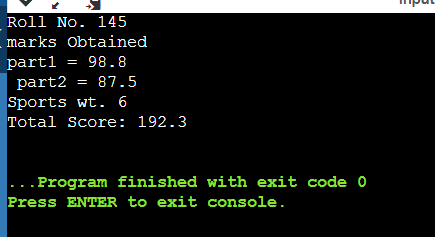
S\_1.get\_marks(98.8, 87.5);

S\_1.get\_score(6.0);

S\_1.display();

return 0;

} **Output:**

****

**35.Write a Program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha & beta.**

**Program:**

#include<iostream>

using namespace std;

class alpha

{

int x;

public:

alpha(int i)

{

x=i;

cout<<"alpha initialized \n";

}

void show\_x(void)

{

cout<<"x = "<<x<<endl;

}

};

class beta

{

float y;

public:

beta(float j)

{

y=j;

cout<<"Beta initialized "<<endl;

}

void show\_y(void)

{

cout<<"y = "<<y<<endl;

}

};

class gamma : public beta , public alpha

{

int m,n;

public:

gamma(int a, float b, int c, int d):

alpha(a) , beta(b)

{

m=c;

n=d;

cout<<"gamma initialized "<<endl;

}

void show\_mn(void)

{

cout<<"m = "<<m<<endl;

cout<<"n = "<<n<<endl;

}

};

int main()

{

gamma g(5, 10.75, 20, 30);

g.show\_x();

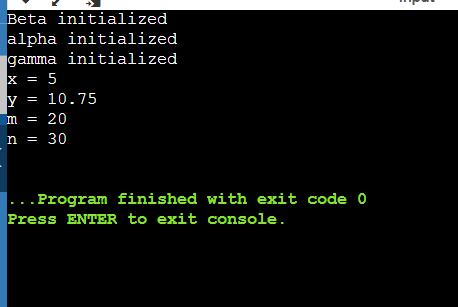
g.show\_y();

g.show\_mn();

return 0;

}

**Output:**

****