**C++ LAB PROGRAMS**

1. Write C++ program to design a class called Bank Account. Include following data members like name

of the depositor, account number and balance. Use following member functions a) to initialize values b)

deposit an amount c) to withdraw an amount d) to display name and balance. (Class and object creation).

#include <iostream>

#include<iomanip>

using namespace std;

class BankAccount

{

char name[25];

int acc\_no;

float balance;

public:

void initialize()

{

cout<<"Enter name of the account holder:";

cin.getline(name,20);

cout<<"Enter account number:";

cin>>acc\_no;

cout<<"Enter current account balance:";

cin>>balance;

}

void deposit()

{

float cre;

cout<<"Enter the amount to be deposited:";

cin>>cre;

balance+=cre;

cout<<"Rs."<<cre<<" successfully debited to your account.\n";

}

void withdraw()

{

float deb;

cout<<"Enter the amount to be withdrawed:";

cin>>deb;

if(deb>balance)

cout<<"Sorry. Insufficient balance.\n";

else{

balance-=deb;

cout<<"Rs."<<deb<<" has been debited from your account.\n";

}

}

void display()

{

cout<<"\nAccount details\n";

cout<<"Name of the account holder : "<<name<<endl;

cout<<"Account No. : "<<acc\_no<<endl;

cout<<"Balance : ";

cout<<fixed<<setprecision(2)<<balance<<endl;

}

};

int main()

{

BankAccount B;

int ch;

B.initialize();

B.display();

cout<<"\nDo you want to deposit/Withdraw:\n\t>1.Deposit\n\t>2.Withdraw\n"<<endl;

cin>>ch;

if(ch==1)

B.deposit();

else if(ch==2)

B.withdraw();

else

cout<<"Invalid choice.\n";

B.display();

return 0;

}

=====================================================================================

2. Write a C++ program to find voulme of a Box. Create a Box class with three data members.

Demonstrate Constructor Overloading .

#include <iostream>

using namespace std;

class Box

{

float length,breadth,height,volume;

public:

Box()

{

length=breadth=height=0;

}

Box(float x)

{

length=breadth=height=x;

}

Box(float x,float y,float z)

{

length=x;

breadth=y;

height=z;

}

void compute()

{

volume=length\*breadth\*height;

}

void display()

{

cout<<"Volume of the box is "<<volume<<endl;

}

};

int main()

{

Box B1;

Box B2(5.0);

Box B3(2.0,3.0,5.0);

B1.compute();

B2.compute();

B3.compute();

B1.display();

B2.display();

B3.display();

return 0;

}

=====================================================================================

3. Given that an EMPLOYEE class contains following members. Data members: Eno, Ename and salary

Member functions: to read the data , to print data members. Write a C++ program to read the data of N

employees and display details of each employee.(use Array of objects concept).

#include <iostream>

#include<conio.h>

using namespace std;

class employee

{

int Eno;

char Ename[25];

float salary;

public:

void getdata()

{

cout<<"Enter employee number:";

cin>>Eno;

cout<<"Enter employee name:";

cin.ignore(1,'\n');

cin.getline(Ename,20);

cout<<"Enter salaray:";

cin>>salary;

}

void display()

{

cout<<"Employee Number:"<<Eno<<endl;

cout<<"Employee Name:"<<Ename<<endl;

cout<<"Salary:"<<salary<<endl<<endl;

}

};

int main()

{

int n;

cout<<"Enter the total number of employees:";

cin>>n;

employee E[n];

cout<<"Enter employee details:\n";

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

{

E[i].getdata();

}

clrscr();

cout<<"EMPLOYEE DETAILS:\n";

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

{

E[i].display();

}

return 0;

}

=====================================================================================

4. Write a C++ program with two classes A and B with one integer data member in each class. Write

member functions to read and display, place a friend function in these classes which takes the data

members of these classes and computes maximum of two data members. Demonstrate using main

function.

#include <iostream>

using namespace std;

class B;

class A

{

int a;

public:

void get\_a()

{

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

cin>>a;

}

friend int max(A,B);

};

class B

{

int b;

public:

void get\_b()

{

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

cin>>b;

}

friend int max(A,B);

};

int max(A ob1,B ob2)

{

if(ob1.a>ob2.b)

return ob1.a;

else

return ob2.b;

}

int main()

{

int m;

A ob1;

B ob2;

ob1.get\_a();

ob2.get\_b();

m=max(ob1,ob2);

cout<<"The maximum value is "<<m<<endl;

return 0;

}

=====================================================================================

5. Write a C++ program for the diagram using Hybrid inheritance. Use your own data members and

member functions to display student details.

#include <iostream>

using namespace std;

class student

{

protected:

int reg\_no;

char name[25];

public:

void get\_student()

{

cout<<"Enter Register Number:";

cin>>reg\_no;

cout<<"Enter name:";

cin.ignore(1,'\n');

cin.getline(name,20);

}

};

class test:public student

{

protected:

int m[6];

public:

void get\_marks()

{

cout<<"Enter 6 subject marks:\n";

for(int i=0;i<6;i++)

{

cin>>m[i];

}

}

};

class sports

{

protected:

int sports\_marks;

public:

void get\_sports\_mark()

{

cout<<"Enter sports marks of student:";

cin>>sports\_marks;

}

};

class result:public test,public sports

{

int total;

float perc;

public:

result(){total=0; perc=0.0;}

void compute()

{

for(int i=0;i<6;i++)

total+=m[i];

total=total+sports\_marks;

perc=total/6.0;

}

void display()

{

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

cout<<"Registration No:"<<reg\_no<<endl;

cout<<"Subject Marks:\n";

for(int i=0;i<6;i++)

{

cout<<" >Subject "<<i+1<<":"<<m[i]<<endl;

}

cout<<"Sports Marks:"<<sports\_marks<<endl;

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

cout<<"Percentage:"<<perc<<endl;

if(perc>=35)

cout<<"Result=PASS";

else

cout<<"Result=FAIL";

}

};

int main()

{

result R;

R.get\_student();

R.get\_marks();

R.get\_sports\_mark();

R.compute();

R.display();

return 0;

}

=====================================================================================

6. Write a C++ program to demonstrate multiple inheritance concepts.

#include <iostream>

using namespace std;

class complex\_1

{

protected:

int real1,img1;

public:

void get\_complex\_1()

{

cout<<"Enter Complex No-1:"<<endl;

cout<<"Enter real part : ";

cin>>real1;

cout<<"Enter imaginary part : ";

cin>>img1;

}

void display()

{

cout<<"Complex No-1 : "<<real1;

if(img1>0) cout<<"+";

cout<<img1<<"i"<<endl;

}

};

class complex\_2

{

protected:

int real2,img2;

public:

void get\_complex\_2()

{

cout<<"Enter Complex No-2:"<<endl;

cout<<"Enter real part : ";

cin>>real2;

cout<<"Enter imaginary part : ";

cin>>img2;

}

void display()

{

cout<<"Complex No-2 : "<<real2;

if(img2>0) cout<<"+";

cout<<img2<<"i"<<endl;

}

};

class sum:public complex\_1,public complex\_2

{

int real, img;

public:

void add()

{

real=real1+real2;

img=img1+img2;

}

void display()

{

complex\_1::display();

complex\_2::display();

cout<<"Sum : "<<real;

if(img>0) cout<<"+";

cout<<img<<"i";

}

};

int main()

{

sum s;

s.get\_complex\_1();

cout<<endl;

s.get\_complex\_2();

s.add();

cout<<endl;

s.display();

return 0;

}

=====================================================================================

7. Write a C++ program to demonstrate static data member and static member function.

#include <iostream>

#include<string.h>

using namespace std;

class enroll

{

char name[25];

int mobile;

static int count;

public:

void get\_details()

{

cout<<"Enter participant name:";

cin.ignore(1,'\n');

cin.getline(name,20);

cout<<"Enter phone number:";

cin>>mobile;

}

static void count\_number()

{

count++;

}

static void display()

{

cout<<"\nTotal enrollments are "<<count<<endl;

}

};

int enroll::count;

int main()

{

int n;

cout<<"Enter the maximum number of participants to be enrolled:";

cin>>n;

cout<<endl;

enroll E[n];

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

{

E[i].get\_details();

enroll::count\_number();

cout<<"\nDo you want to enroll new name? Types Yes/No\n";

char s[5];

cin>>s;

if(strcmpi(s,"Yes")!=0)

break;

}

enroll::display();

return 0;

}

=====================================================================================

8. Write a C++ program with class Time with data members that represents hours and minutes. Include

appropriate member functions to compute time in hours and minutes . (Use of objects as arugemnts).

#include <iostream>

using namespace std;

class time

{

int hours,minutes;

public:

void get\_time()

{

cout<<"Enter hours:";

cin>>hours;

cout<<"Enter Minutes:";

cin>>minutes;

cout<<endl;

}

void show\_time()

{

cout<<hours<<"hours "<<minutes<<"minutes"<<endl;

}

friend void total\_time(time,time);

};

void total\_time(time ob1,time ob2)

{

int h,m;

cout<<"Time 1 : ";

ob1.show\_time();

cout<<"Time 2 : ";

ob2.show\_time();

m=ob1.minutes+ob2.minutes;

h=ob1.hours+ob2.hours;

if(m>=60)

{

h=h+(m/60);

m=m%60;

}

cout<<"Total time : "<<h<<"hours "<<m<<"minutes"<<endl;

}

int main()

{

time t1,t2;

t1.get\_time();

t2.get\_time();

total\_time(t1,t2);

return 0;

}

=====================================================================================

9. Write a C++ program to demonstrate to uses of constructors in derived class concept. (Any inheritance

you can use but constructors in base class should have at least one parameter).

#include <iostream>

#include<iomanip>

using namespace std;

class rectangle1

{

protected:

float l,b;

public:

rectangle1(float x,float y)

{

l=x;

b=y;

cout<<"Rectangle-1 initialized.\n";

}

void show()

{

cout<<"Rectangle 1 : Length="<<l<<" Breadth="<<b<<endl;

}

};

class rectangle2

{

protected:

float l,b;

public:

rectangle2(float x,float y)

{

l=x;

b=y;

cout<<"Rectangle-2 initialized.\n";

}

void show()

{

cout<<"Rectangle 2 : Length="<<l<<" Breadth="<<b<<endl;

}

};

class Area:public rectangle1,public rectangle2

{

float area1,area2,area;

public:

Area(float a,float p,float q,float r,float s):rectangle1(p,q),rectangle2(r,s)

{

area=area1=area2=a;

cout<<"Area1,area2 and area pre-initialized to "<<a<<endl;

}

void compute()

{

area1=rectangle1::l\*rectangle1::b;

area2=rectangle2::l\*rectangle2::b;

area=area1+area2;

}

void show()

{

rectangle1::show();

rectangle2::show();

cout<<"Area of rectangle-1:"<<area1<<endl;

cout<<"Area of rectangle-2:"<<area2<<endl;

cout<<"Total Area:"<<area<<endl;

}

};

int main()

{

Area A(0,3.0,2.0,5.5,2.5);

A.compute();

A.show();

return 0;

}

=====================================================================================

10. Design a C++ program to illustrate function overloading in a class.

#include <iostream>

using namespace std;

class Volume

{

float v;

public:

Volume()

{

v=0.0;

}

void volume(float x)

{

v=(x\*x\*x);

}

void volume(float x,float y)

{

v=3.142\*x\*x\*y;

}

void volume(float x,float y,float z)

{

v=x\*y\*z;

}

void display()

{

cout<<v<<endl;

}

};

int main()

{

int ch;

Volume V;

cout<<"Enter your choice\n1.Volume of Cube\n2.Volume of cylinder\n3.Volume of cuboid\n";

cin>>ch;

switch(ch)

{

case 1:

float x;

cout<<"Enter side of cube:";

cin>>x;

V.volume(x);

cout<<"Volume of cube is ";

V.display();

break;

case 2: {

float x,y;

cout<<"Enter radius:";

cin>>x;

cout<<"Enter height:";

cin>>y;

V.volume(x,y);

cout<<"Volume of cylinder is ";

V.display();

break;

}

case 3: {

float x,y,z;

cout<<"Enter length:";

cin>>x;

cout<<"Enter breadth:";

cin>>y;

cout<<"Enter height:";

cin>>z;

V.volume(x,y,z);

cout<<"Volume of cuboid is ";

V.display();

break;

}

default:cout<<"Invalid choice.";

}

return 0;

}

=====================================================================================

11. Write a C++ program to demonstrate multilevel inheritance for the below diagram.

(Assume appropriate member functions and data members).

#include <iostream>

#include<conio.h>

using namespace std;

class student

{

protected:

int reg\_no;

char name[25];

public:

void get\_data()

{

cout<<"Enter Register Number : "<<endl;

cin>>reg\_no;

cout<<"Enter Name:";

cin.ignore(1,'\n');

cin.getline(name,20);

}

};

class test:public student

{

protected:

int marks[6];

public:

void get\_marks()

{

cout<<"Enter 6 subject marks :\n";

for(int i=0; i<6;i++)

cin>>marks[i];

}

};

class result:public test

{

int total;

float perc;

public:

void compute\_res()

{

for(int i=0; i<6; i++)

total+=marks[i];

perc=total/6.0;

}

void display()

{

cout<<"\nRegister No. : "<<reg\_no<<endl;

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

cout<<"Marks:\n";

for(int i=0;i<6;i++)

cout<<" > Subject "<<i+1<<" : "<<marks[i]<<endl;

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

cout<<"Percentage : "<<perc<<endl;

if(perc>=35)

cout<<"Result = PASS";

else

cout<<"Result = FAIL";

}

};

int main()

{

result R;

R.get\_data();

R.get\_marks();

R.compute\_res();

R.display();

return 0;

}

=====================================================================================

12. Write C++ program to demonstrate copy constructor.

#include <iostream>

using namespace std;

class copy\_constructor

{

int a,b;

public:

copy\_constructor(int x,int y)

{

a=x ; b=y;

}

void display()

{

cout<<"a="<<a<<"\tb="<<b<<endl;

}

};

int main()

{

copy\_constructor obj1(10,20);

cout<<"Contents of Object-1 :\n"<<endl;

obj1.display();

copy\_constructor obj2=obj1;

cout<<"\nContents of Object-2 copied from object-1 :\n";

obj2.display();

return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*

------------------------------------------------------------------------------------------------------------------------------------------