1. Write a program to calculate bonus of the employees. The class master derives the information from both admin and account classes which derives information from class person. Create base and all derived classes having same member functions and different parameters called getdata, display data and bonus. Create a base class pointer that capable of accessing data of any class and calculates bonus of the specified employee.

Code:

# include <iostream>

using namespace std;

class person

{

int id;

char name[30];

public:

virtual void getdata();

virtual void display();

virtual void bonus();

};

void person :: getdata()

{ cout <<"\n Enter the name and number of the employee: "<< endl;

cin >> name>>id;

}

void person :: display()

{

cout <<"\n Employee Name = "<< name<<"\n Number = "<<id;

}

void person :: bonus()

{

cout <<"\n Bonus function of person base class is called ";

}

class admin : public virtual person

{

int sale\_admin;

public:

void getdata();

void display();

void bonus();

};

void admin :: getdata()

{

cout <<"\n Enter the Sale amount of the admin : "<< endl;

cin >> sale\_admin;

}

void admin :: display()

{

cout <<"\n Admin Employee sale amount = "<< sale\_admin;

}

void admin :: bonus()

{

if (sale\_admin >2000)

cout << "\n The bonus of the admin is : "<< (sale\_admin\*0.25);

}

class account : public virtual person

{

long int amt;

public :

void getdata();

void display();

void bonus();

};

void account :: getdata()

{

cout <<"\n Enter the amount accumulated in the account : "<< endl;

cin >> amt;

}

void account:: display()

{

cout <<"\n Employee Account amount = "<< amt;

}

void account :: bonus()

{

if (amt >10000)

cout << "\n The bonus of the admin is : "<< (amt\*0.25);

}

class master : public admin, public account

{

public:

void getdata();

void display();

void bonus();

};

void master :: getdata()

{

cout <<"\n Executing getdata of master class :";

person::getdata();

admin:: getdata();

account::getdata();

}

void master :: display()

{

cout <<"\n executing display of master class :";

person::display();

admin:: display();

account::display();

}

void master :: bonus()

{

cout <<"\n executing bonus of master class :";

person::bonus();

admin:: bonus();

account::bonus();

}

int main()

{

person p, \*per\_ptr;

admin a, \*admin\_ptr;

account ac, \*acc\_ptr;

master ms, \*ms\_ptr;

cout <<"\nassigning the addr of base class person to base class pointer ";

per\_ptr=&p;

per\_ptr->getdata();

per\_ptr->display();

per\_ptr->bonus();

cout <<"\n assigning the addr of admin class to base class pointer ";

per\_ptr=&a;

per\_ptr->getdata();

per\_ptr->display();

per\_ptr->bonus();

cout <<"\n assigning the addr of account class to base class pointer ";

per\_ptr=&ac;

per\_ptr->getdata();

per\_ptr->display();

per\_ptr->bonus();

cout <<"\n assigning the addr of class master to base class pointer ";

per\_ptr=&ms;

per\_ptr->getdata();

per\_ptr->display();

per\_ptr->bonus();

return 0;

}

Output:

ssigning the addr of base class person to base class pointer

Enter the name and number of the employee:

arjun

23

Employee Name = arjun

Number = 23

Bonus function of person base class is called

assigning the addr of admin class to base class pointer

Enter the Sale amount of the admin :

78000

Admin Employee sale amount = 78000

The bonus of the admin is : 19500

assigning the addr of account class to base class pointer

Enter the amount accumulated in the account :

76000

Employee Account amount = 76000

The bonus of the admin is : 19000

2.Write a C++ program to calculate the gross and net pay of employee from basic salary. Create employee which consists of employee name,emp\_id, and basic salary as its data members. Use parameterized constructions in the derived class to initialize data members of the base class and calculate gross and net pay of the employee in the derived class.

Code

:# include<iostream>

using namespace std;

class salary

{

public:

int bs,net,hra,oa,alla,gp;

salary()

{

cout<<"Enter basic salary : ";

cin>>bs;

cout<<"Enter HRA : ";

cin>>hra;

cout<<"Enter other allowence : ";

cin>>oa;

cout<<"Enter all allowance for net salary : ";

cin>>alla;

net=bs+hra+oa+alla;

cout<<"Your net salary is "<<net<<"\n";

gp=bs+hra+oa;

cout<<"Your gross pay is "<<gp<<"\n";

}

};

class deduction

{

public:

void deduction1()

{

cout<<"\n"<<"Income tax calculation Enter your details below "<<"\n";

}

};

class incometax : public salary

{

public:

incometax()

{

if(net>200000 and net<500000)

{

cout<<"Your income tax is "<<net\*0.5;

}

else if(net>500000 and net<700000)

{

cout<<"Your income tax is "<<(net\*0.10)+12500;

}

else if(net>700000 and net<1000000)

{

cout<<"Your income tax is "<<(net\*0.15)+37500;

}

else if(net>1000000)

{

cout<<"Your income tax is "<<(net\*0.20)+75000;

}

else

{

cout<<"No income tax";

}

}

};

int main()

{

salary p1;

deduction p2;

p2.deduction1();

incometax p3;

}

Output: Enter basic salary : 56000

Enter HRA : 4500

Enter other allowence : 3000

Enter all allowance for net salary : 3900

Your net salary is 67400

Your gross pay is 63500

3. Write a C++ program to demonstrate the multiple inheritance by creating a class cuboid which extends class rectangle, class shape. It calculates area and volume.

Code:

#include <iostream>

using namespace std;

class Volume {

public:

float volume(float l, float b, float h) {

return (l \* b \* h);

}

};

class Area {

public:

float area(float l, float b, float h) {

return (2 \* (l \* b + l \* h + b \* h));

}

};

class Cuboid: private Volume, private Area {

private: float length,

breadth,

height;

public: Cuboid(): length(0.0),

breadth(0.0),

height(0.0) {}

void getDimensions() {

cout << "\nEnter the length of the Cuboid: ";

cin >> length;

cout << "\nEnter the breadth of the Cuboid: ";

cin >> breadth;

cout << "\nEnter the height of the Cuboid: ";

cin >> height;

}

float volume() {

return Volume::volume(length, breadth, height);

}

float area() {

return Area::area(length, breadth, height);

}

};

int main() {

cout << "\n\nWelcome to Studytonight :-)\n\n\n";

cout << " ===== Program to demonstrate the concept of Multiple Level Inheritence in CPP ===== \n\n";

Cuboid cuboid;

cout << "\nCalling the getDimensions() method from the main() method:\n\n";

cuboid.getDimensions();

cout << "\n\n";

cout << "\nArea of the Cuboid computed using Area Class is : " << cuboid.area() << "\n\n\n";

cout << "Volume of the Cuboid computed using Volume Class is: " << cuboid.volume();

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

return 0;

}

Output:

Enter the length of the Cuboid: 4

Enter the breadth of the Cuboid: 4

Enter the height of the Cuboid: 3

Area of the Cuboid computed using Area Class is : 80

Volume of the Cuboid computed using Volume Class is: 48

4. Write a program for matrix addition?

Code:

#include<iostream>

using namespace std;

int main()

{

int a[10][10],b[10][10],c[10][10],row,col,i,j;

cout<<"enter the number of rows:";

cin>>row;

cout<<"enter the number of coloumn:";

cin>>col;

cout<<"enter A matrix element:";

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

for(j=0;j<col;j++){

cin>>a[i][j];

}

}

cout<<"enter B matrix element:";

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

for(j=0;j<col;j++){

cin>>a[i][j];

}

}

cout<<"ADDITION OF MATRIX A&B : \n";

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

for(j=0;j<col;j++){

c[i][j]=a[i][j]+b[i][j];

cout<<c[i][j];

cout<<" ";

}

cout<<"\n";

}

return 0;

}

Output:

enter the number of rows:2

enter the number of coloumn:1

enter A matrix element:3

3

enter B matrix element:5

4

ADDITION OF MATRIX A&B :

5

4342193

5. Write a program to print rectangle symbol pattern. Get the symbol as input from user

Code:

#include <iostream>

using namespace std;

int main() {

int l, w, i, j;

cout << "Enter the length: ";

cin >> l;

cout << "Enter the width: ";

cin >> w;

cout << endl;

for (i = 1; i <= w; i++) {

for (j = 1; j <= l; j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

Output:

Enter the length: 3

Enter the width: 4

\*\*\*

\*\*\*

\*\*\*

6. Write a program to calculate tax given the following conditions:

If income is less than or equal to 1,50,000 then no tax

If taxable income is 1,50,001 – 3,00,000 the charge 10% tax

If taxable income is 3,00,001 – 5,00,000 the charge 20% tax

If taxable income is above 5,00,001 then charge 30% tax

Code:

#include<iostream>

using namespace std;

int main()

{

int income,tax;

cout<<" enter the income of the person :";

cin>>income;

if (income<=150000)

tax = 0;

else if(income>=150001 && income<=300000)

tax = income\*0.1;

else if(income>=300001 && income<=500000)

tax = income\*0.2;

else

tax=income\*0.3;

cout<<" the income tax is :"<<tax<<endl;

return 0;

}

Output:

enter the income of the person :6000

the income tax is :0

7.Write a program that would sort a list of names in alphabetical order Ascending or Descending, choice get from the user?

Code:

#include<iostream>

#include<string.h>

#include<stdio.h>

using namespace std;

int main()

{

char str[5][20], t[20];

int i, j;

cout<<"\n Enter Any Five Names : \n\n";

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

{

cout<<" ";

cin>>str[i];

}

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

{

for(j=1; j<5; j++)

{

if(strcmp(str[j-1], str[j])>0)

{

strcpy(t, str[j-1]);

strcpy(str[j-1], str[j]);

strcpy(str[j], t);

}

}

}

cout<<"\n Names Sorted in Alphabetical Order : \n\n";

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

{

cout<<" ";

cout<<str[i]<<"\n";

}

return 0;

}

Output:

Enter any five names:

Ball

Apple

Cat

Dad

Red

Alphabetical order:

Apple

Ball

Cat

Dad

Red

8. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Code:

#include <iostream>

using namespace std;

int main(){

float a,b,c,d,tot;

float agg;

cout<<"Enter the marks in python ";

cin>>a;

cout<<"Enter the marks in c programming:";

cin>>b;

cout<<"Enter the marks in Mathematics:";

cin>>c;

cout<<"Enter the marks in Physics:";

cin>>d;

tot=a+b+c+d;

cout<<"TOTAL : " << tot;

cout<<"\n";

agg=tot/4;

cout<<"AGGREGATE : " <<agg;

cout<<"\n";

if (agg>=75){

cout<<"DISTINCTION";

}

else if (agg>=60 &&agg<75){

cout<<"FIRST DIVISION";

}

else if (agg>= 50 &&agg<60){

cout<<"SECOND DIVISION";

}

else if (agg>= 40 &&agg<50){

cout<<"THIRD DIVISION";

}

else if (agg< 40){

cout<<"FAIL";

}

else {

cout<<"invalid input";

}

return 0;

}

Output:

Enter the marks in python 56

Enter the marks in c programming:67

Enter the marks in Mathematics:78

Enter the marks in Physics:89

TOTAL : 290

AGGREGATE : 72.5

FIRST DIVISION

9. Find the factorial of n?

Code:

#include <iostream>

using namespace std;

int main() {

int n;

long factorial = 1.0;

cout << "Enter a positive integer: ";

cin >> n;

if (n < 0)

cout << "Error! Factorial of a negative number doesn't exist.";

else {

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

factorial \*= i;

}

cout << "Factorial of " << n << " = " << factorial;

}

return 0;

}

Output:

Enter a positive integer: 5

Factorial of 5 = 120

10. Write a Program to create a list of all numbers in a range which are perfect squares and the sum of the digits of the number is less than 10.

Code:# include<iostream>

#include<math.h>

using namespace std;

int main()

{

int range1,range2;

cout<<"Enter a range:";

cin>>range1>>range2;

int count=0;

cout<<"Perfect numbers between "<<range1<<" and "<<range2<<" are: ";

for(int i=range1;i<=range2;i++)

{

double sqr=sqrt(i);

if(sqr-floor(sqr)==0)

cout<<i<<" ";

}

}

Output:

Enter a range:3

19

Perfect numbers between 3 and 19 are: 4 9 16

11. Write a Program to Find the Nth Largest Number in a array

Code:

#include <iostream>

using namespace std;

int main() {

int i, n;

float arr[100];

cout << "Enter total number of elements(1 to 100): ";

cin >> n;

cout << endl;

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

cout << "Enter Number " << i + 1 << " : ";

cin >> arr[i];

}

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

if(arr[0] < arr[i])

arr[0] = arr[i];

}

cout << endl << "Largest element = " << arr[0];

return 0;

}

Output: Enter total number of elements(1 to 100): 4

Enter Number 1 : 34

Enter Number 2 : 67

Enter Number 3 : 78

Enter Number 4 : 56

12. Write a program to find the sum of digits of N digit number (sum should be single digit)

Code:

#include <iostream>

using namespace std;

int main()

{

int n,sum=0,m,i;

cout<<"Enter how many numbers:";

cin>>i;

cout<<"Enter n number: ";

cin>>n;

while(n>0)

{

m=n%10;

sum=sum+m;

n=n/10;

}

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

return 0;

}

Output:

Enter how many numbers:

2

1st num:2

2nd num:4

Sum:6