1. **Write a C++ program to Calculate the factorial of N. N is declared at different scope levels. Use scope resolution operator to display the various factorial values of the same variables declared at different scope levels.**

#include<iostream>

using namespace std;

int N=5;

int factorial(int n){

int fact=1;

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

fact\*=i;

}

return fact;

}

int main(){

int N=4;

cout<<"Factorial of local scope N(4):"<< factorial(N) <<endl;

cout<<"Factorial of Global scope N(5) :"<< factorial(::N) <<endl;

return 0;

}

2. **Write a C++ program to create a Student class with attributes (name, rollNumber, and marks) which are private. Implement appropriate methods to read the data, calculate the grade based on the marks and display it for 3 students.**

#include <iostream>

using namespace std;

class student{

private:

string name;

int rollnumber;

double marks;

public:

void read(){

cout<<"Enter the name:";

cin>>name;

cout<<"Enter the roll number:";

cin>>rollnumber;

cout<<"Enter the marks:";

cin>>marks;

}

char grade(){

if(marks >=90)

return 'A';

else if(marks >=75 && marks < 90)

return 'B';

else if(marks >=45 && marks<75)

return 'C';

else

return 'F';

}

void display(){

cout<<"Name:"<<name << ", Roll Number :"<<rollnumber <<", Marks :" << marks <<",Grade:"<<

grade() << endl;

}

};

int main(){

int n=3,i;

student s[n];

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

cout<<"Enter the details of "<< i+1 <<" student \n";

s[i].read();

}

cout<<"Details of all the students \n";

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

s[i].display();

}

return 0;

}

3.

#include <iostream>

#include <cmath>

class Interest {

private:

float principal;

float time;

float rate;

public:

Interest() {

principal = 0.0;

time = 0.0;

rate = 12.0;

}

Interest(int p, int t, int r) {

principal = p;

time = t;

rate = r;

}

void get() {

std::cout << "Enter principal: ";

std::cin >> principal;

std::cout << "Enter time (in years): ";

std::cin >> time;

}

float CompInt() {

return principal \* (pow((1 + rate / 100), time) - 1);

}

void display() {

float compoundInterest = CompInt();

float totalAmount = principal + compoundInterest;

std::cout << "Principal: " << principal << std::endl;

std::cout << "Time: " << time << " years" << std::endl;

std::cout << "Rate of Interest: " << rate << "%" << std::endl;

std::cout << "Compound Interest: " << compoundInterest << std::endl;

std::cout << "Total Amount: " << totalAmount << std::endl;

}

~Interest() {

}

};

int main() {

Interest defaultInterest;

defaultInterest.get();

defaultInterest.display();

Interest parameterizedInterest(1000, 5, 10);

parameterizedInterest.display();

return 0;

}

4.

#include <iostream>

using namespace std;

int findLargest(const int matrix[4][4]) {

int largest = matrix[0][0];

for (int i = 0; i < 4; ++i) {

for (int j = 0; j < 4; ++j) {

if (matrix[i][j] > largest) {

largest = matrix[i][j];

}

}

}

return largest;

}

int sumArray(const int arr[], int N) {

int sum = 0;

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

sum += arr[i];

}

return sum;

}

int main() {

int matrix[4][4];

cout << "Enter values for a 4x4 matrix:\n";

for (int i = 0; i < 4; ++i) {

for (int j = 0; j < 4; ++j) {

cout << "Enter value for matrix[" << i << "][" << j << "]: ";

cin >> matrix[i][j];

}

}

int N;

cout << "Enter the number of elements in the array: ";

cin >> N;

int\* arr = new int[N];

cout << "Enter " << N << " values for the array:\n";

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

cout << "Enter value for arr[" << i << "]: ";

cin >> arr[i];

}

int largestInMatrix = findLargest(matrix);

cout << "Largest number in the 4x4 matrix: " << largestInMatrix << endl;

int sumOfArray = sumArray(arr, N);

cout << "Sum of the array: " << sumOfArray << endl;

delete[] arr;

return 0;

}