CCC Training-DAY 3

C++ DMA[Dynamic memory allocation]

Example 1:

#include<iostream>

using namespace std;

int main(){

int x=50;

int \*y=&x;

cout << \*y;

}

Output: 50

Example 2:

#include<iostream>

using namespace std;

int main(){

int x=50;

int \*y=&x;

cout<<x<<endl;

cout<<&x<<endl;

cout<<\*y<<endl;

cout<<&y<<endl;

}

output: 50

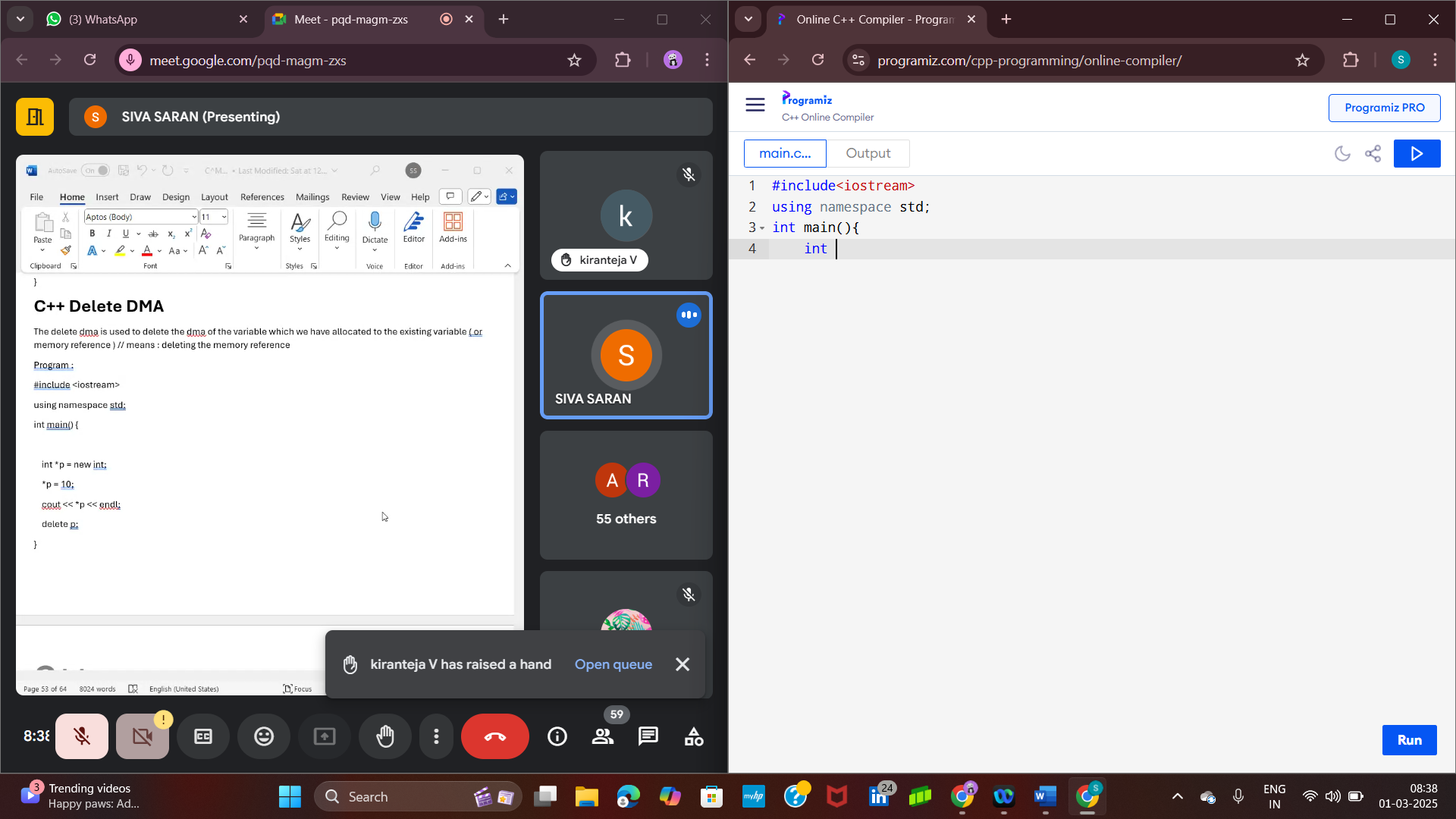
0x7ffe47871e7c

50

0x7ffe47871e70C++

C++ Delete DMA

Example:



C++ Segmentation fault

Example:

#include<iostream>

using namespace std;

int main(){

int x=10;

int \*y=&x;

int \*\*z=&y;

cout<<x<<endl;

delete y;

cout<<\*\*z<<endl;

}

Output: 10

Segmentation fault

Classes:

* The object oriented programming is nothing but of creating a class and inside the class we need to pass the object.
* OPP=data + functions

Advantages:

* OPP is faster and easier.
* Provides a clear data structure

Syntax:

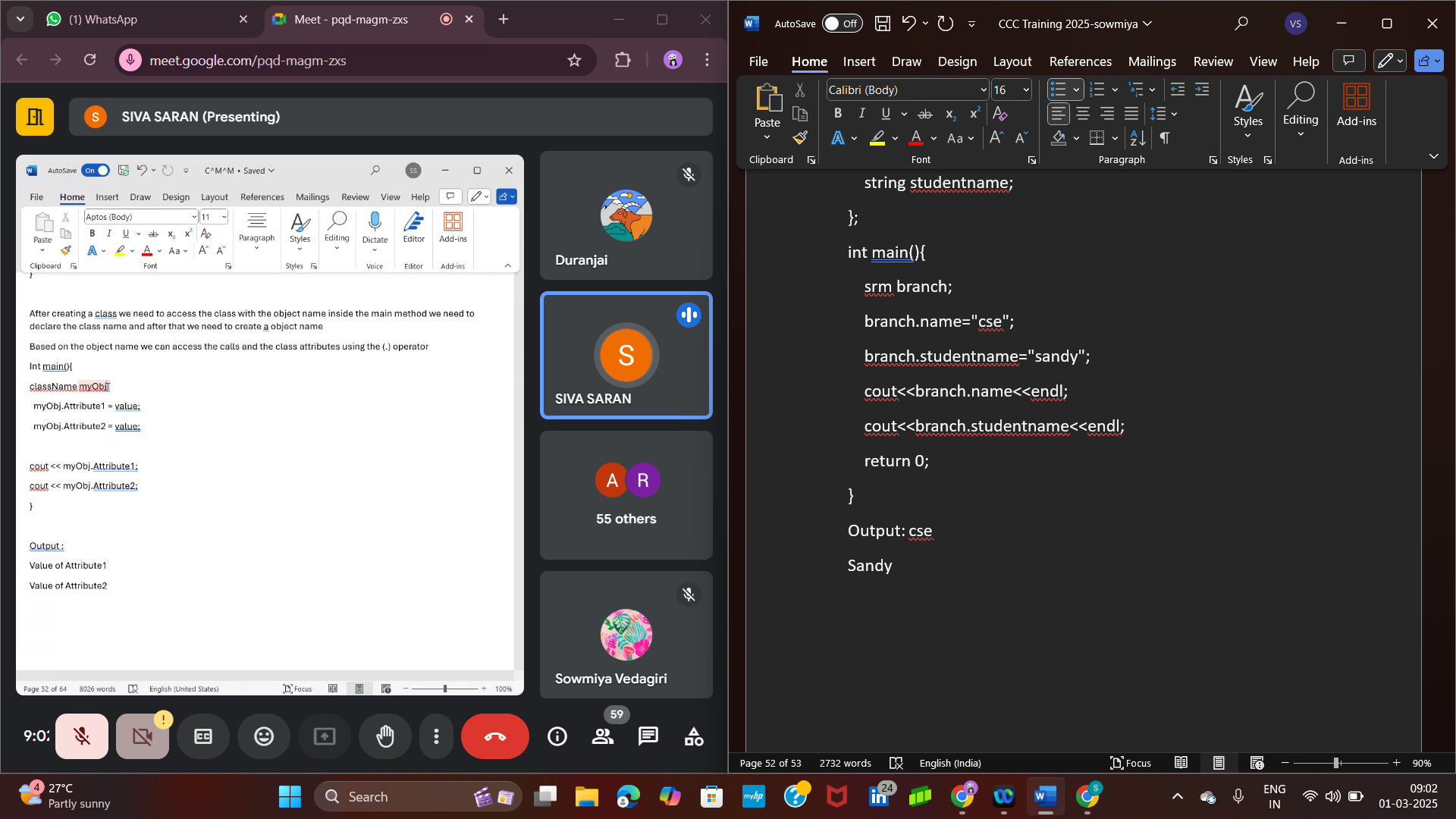
Class classname{

Public; // access specifier

Attribute 1;

Attribute2;

}



Example:

#include<iostream>

using namespace std;

class srm{

public:

string name;

string studentname;

};

int main(){

srm branch;

branch.name="cse";

branch.studentname="sandy";

cout<<branch.name<<endl;

cout<<branch.studentname<<endl;

return 0;

}

Output: cse

Sandy

Different datatypes inside the class

Example:

#include<iostream>

using namespace std;

class srm{

public:

int roll;

string name;

string branch;

float grade;

};

int main(){

srm college;

college.roll=1234;

college.name="RMV";

college.branch="cse";

college.grade=8.97;

cout<<"roll no"<<college.roll<<endl;

cout<<"college name"<<college.name<<endl;

cout<<"branch"<<college.branch<<endl;

cout<<"grade"<<college.grade<<endl;

return 0;

}

Output:

roll no1234

college nameRMV

branchcse

grade8.97

Classes and Objects programs

program 1:

#include<iostream>

using namespace std;

class Dog{

public:

string name;

string breed;

int age;

};

int main(){

Dog myDog;

myDog.name="buddy";

myDog.breed="labrador";

myDog.age=3;

cout<< "Dog name:"<<myDog.name<<endl;

cout<< "Dog breed:"<<myDog.breed<<endl;

cout<< "Dog age:"<<myDog.age<<endl;

return 0;

}

Output: Dog name:buddy

Dog breed:labrador

Dog age:3

woof!

Program 2:

#include<iostream>

using namespace std;

class car{

public:

string make;

string model;

int year;

};

int main(){

car mycar;

mycar.make="toyota";

mycar.model="carolla";

mycar.year=2024;

cout<< "car make:"<<mycar.make<<endl;

cout<< "car model:"<<mycar.model<<endl;

cout<< "car year:"<<mycar.year<<endl;

return 0;

}

Output: car make:toyota

car model:carolla

car year:2024

the engine is starting

program 3:

#include<iostream>

using namespace std;

class student{

public:

string name;

string grade;

int age;

};

int main(){

student mystudent;

mystudent.name="sandy";

mystudent.grade="A";

mystudent.age=18;

cout<< "student name:"<<mystudent.name<<endl;

cout<< "student grade:"<<mystudent.grade<<endl;

cout<< "student age:"<<mystudent.age<<endl;

return 0;

}

Output: student name:sandy

student grade:A

student age:18

student details

program 4:

#include<iostream>

using namespace std;

class book{

public:

string title;

string author;

int pages;

};

int main(){

book demobook;

demobook.title="The god of small things";

demobook.author="arundhati roy";

demobook.pages=356;

cout<< "book title:"<<demobook.title<<endl;

cout<< "book author:"<<demobook.author<<endl;

cout<< "book pages:"<<demobook.pages<<endl;

return 0;

}

Output: book title:The god of small things

book author:arundhati roy

book pages:356

program 5:

#include<iostream>

using namespace std;

class rec {

public:

double height;

double width;

double area() {

return width \* height;

}

double perimeter() {

return 2 \* (width + height);

}

};

int main() {

rec arearec;

arearec.height = 15;

arearec.width = 22;

cout << "rec height: " << arearec.height << endl;

cout << "rec width: " << arearec.width << endl;

cout << "rec area: " << arearec.area() << endl;

cout << "rec perimeter: " << arearec.perimeter() << endl;

return 0;

}

Output: rec height: 15

rec width: 22

rec area: 330

rec perimeter: 74

program 6:

#include<iostream>

using namespace std;

class Cir {

public:

double r;

double area() {

return 3.14 \*r\*r;

}

double circumference() {

return 2 \*3.14\*r;

}

};

int main() {

Cir myCir;

myCir.r = 7.0;

cout << "Area: " << myCir.area() << endl;

cout << "Circumference: " << myCir.circumference() << endl;

return 0;

}

Output: Area: 153.86

Circumference: 43.96

C++ Class methods:

Create method inside the class

Syntax:

#include <iostream>

using namespace std;

class srm{

public:

void mymethod(){

cout<<"hello world";

}

};

Calling method inside the main method

Syntax:

int main(){

Class methodObj;

methodObj.myMethod();

return 0;

}

Example:

#include<iostream>

using namespace std;

class srm{

public:

void cse(){

cout<<"hello cse"<<endl;

}

void ece(){

cout<<"hello ece"<<endl;

}

int add(int x,int y){

return x+y;

}

};

int main(){

srm branch;

branch.cse();

cout<<endl;

branch.ece();

cout<<endl;

int res = branch.add(2,5);

cout<<"the result is;"<<res<<endl;

return 0;

}

Output: hello cse

hello ece

the result is;7

program 1:

#include <iostream>

using namespace std;

class person{

public:

string firstname;

string lastname;

string fullname(string first,string last){

return first + " " + last;

}

};

int main() {

person p;

cout<<"Enter first name: ";

cin>>p.firstname;

cout<<"Enter last name: ";

cin>>p.lastname;

cout<<"Full name is: "<<p.fullname(p.firstname,p.lastname);

return 0;

}

Output: Enter first name: riya

Enter last name: doss

Full name is: riya doss

program 2:

#include <iostream>

using namespace std;

class BankAccount {

private:

double balance;

public:

string accountno;

string accountdetails() {

cout << "Enter account number: ";

cin >> accountno;

cout << "Enter balance: ";

cin >> balance;

return "Account details entered ";

}

double depositmoney(double amount) {

balance += amount;

return balance;

}

double withdrawmoney(double amount) {

if (amount <= balance) {

balance -= amount;

return balance;

} else {

cout << "Insufficient funds!" << endl;

return balance;

}

}

double currentbalance() {

return balance;

}

};

int main() {

BankAccount account;

cout << account.accountdetails() << endl;

double depositAmount, withdrawAmount;

cout << "Enter deposit amount: ";

cin >> depositAmount;

double newBalance = account.depositmoney(depositAmount);

cout << "Deposited: " << depositAmount << ". New Balance: " << newBalance << endl;

cout << "Enter withdrawal amount: ";

cin >> withdrawAmount;

newBalance = account.withdrawmoney(withdrawAmount);

cout << "Withdrawn: " << withdrawAmount << ". New Balance: " << newBalance << endl;

cout << "Current Balance: " << account.currentbalance() << endl;

return 0;

}

Output: Enter account number: 644

Enter balance: 965

Account details entered

Enter deposit amount: 8756

Deposited: 8756. New Balance: 9721

Enter withdrawal amount: 087

Withdrawn: 87. New Balance: 9634

Current Balance: 9634

program 3:

#include <iostream>

using namespace std;

class Movie {

public:

string title, director;

int year;

Movie() {

cout << "Enter movie title: ";

cin.ignore();

getline(cin, title);

cout << "Enter director name: ";

getline(cin, director);

cout << "Enter release year: ";

cin >> year;

}

string describe() {

return title + " directed by " + director + " in " + to\_string(year) + ".";}

};

int main() {

Movie m;

cout << m.describe() << endl;

return 0;

}

Output: Enter movie title: remo

Enter director name: bakiyaraj

Enter release year: 2016

emo directed by bakiyaraj in 2016.

program 4:

#include <iostream>

using namespace std;

class Employee {

public:

string name, position;

double salary;

Employee() {

cout << "Enter name: ";

cin.ignore();

getline(cin, name);

cout << "Enter position: ";

getline(cin, position);

cout << "Enter salary: ";

cin >> salary;

}

double give\_raise(double amount) {

salary += amount;

return salary;

}

string display\_info() {

return "Name: " + name + ", Position: " + position + ", Salary: " + to\_string(salary);

}

};

int main() {

Employee emp;

double raiseAmount;

cout << "Enter raise amount: ";

cin >> raiseAmount;

double newSalary = emp.give\_raise(raiseAmount);

cout << emp.name << " received a raise of " << raiseAmount << ". New Salary: " << newSalary << endl;

cout << emp.display\_info() << endl;

return 0;

} #include <iostream>

using namespace std;

class Employee {

public:

string name, position;

double salary;

Employee() {

cout << "Enter name: ";

cin.ignore();

getline(cin, name);

cout << "Enter position: ";

getline(cin, position);

cout << "Enter salary: ";

cin >> salary;

}

double give\_raise(double amount) {

salary += amount;

return salary;

}

string display\_info() {

return "Name: " + name + ", Position: " + position + ", Salary: " + to\_string(salary);

}

};

int main() {

Employee emp;

double raiseAmount;

cout << "Enter raise amount: ";

cin >> raiseAmount;

double newSalary = emp.give\_raise(raiseAmount);

cout << emp.name << " received a raise of " << raiseAmount << ". New Salary: " << newSalary << endl;

cout << emp.display\_info() << endl;

return 0;

}

Output: Enter name: riya

Enter position: manager

Enter salary: 2767654

Enter raise amount: 3654653

iya received a raise of 3.65465e+06. New Salary: 6.42231e+06

Name: iya, Position: manager, Salary: 6422307.000000

program 5:

#include <iostream>

#include <cmath>

using namespace std;

class Point {

public:

double x, y;

void get\_details() {

cout << "Enter x-coordinate: ";

cin >> x;

cout << "Enter y-coordinate: ";

cin >> y;

}

void move(double dx, double dy) {

x += dx;

y += dy;

}

double distance\_to(Point other) {

return sqrt(pow(other.x - x, 2) + pow(other.y - y, 2));

}

};

int main() {

Point p1, p2;

cout << "Enter details for Point 1:" << endl;

p1.get\_details();

cout << "Enter details for Point 2:" << endl;

p2.get\_details();

cout << "Initial Distance: " << p1.distance\_to(p2) << endl;

double dx, dy;

cout << "Enter movement dx and dy: ";

cin >> dx >> dy;

p1.move(dx, dy);

cout << "Distance after moving: " << p1.distance\_to(p2) << endl;

return 0;

}

Output: Enter details for Point 1:

Enter x-coordinate: 4

Enter y-coordinate: 6

Enter details for Point 2:

Enter x-coordinate:

IMPORTANT QUESTIONS

Program 1:

#include <iostream>

using namespace std;

class Calculator {

private:

int a, b;

public:

void set\_values(int x, int y) {

a = x;

b = y;

}

int add() { return a + b; }

int subtract() { return a - b; }

int multiply() { return a \* b; }

int divide() { return (b != 0) ? a / b : 0; } // Handling divide by zero

};

int main() {

Calculator calc;

int x, y;

cout << "Enter two numbers: ";

cin >> x >> y;

calc.set\_values(x, y);

cout << "Addition: " << calc.add() << endl;

cout << "Subtraction: " << calc.subtract() << endl;

cout << "Multiplication: " << calc.multiply() << endl;

cout << "Division: " << calc.divide() << endl;

return 0;

}

Output: Enter two numbers: 10

2

Addition: 12

Subtraction: 8

Multiplication: 20

Division: 5

Program 2:

#include <iostream>

using namespace std;

class ArrayHandler {

private:

int arr[5];

public:

void set\_array() {

cout << "Enter 5 elements: ";

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

cin >> arr[i];

}

void print\_array() {

cout << "Array elements: ";

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

cout << arr[i] << " ";

cout << endl;

}

};

int main() {

ArrayHandler obj;

obj.set\_array();

obj.print\_array();

return 0;

}

Output: Enter 5 elements: 5

4

5

4

3

Array elements: 5 4 5 4 3

Program 3:

#include <iostream>

using namespace std;

class Matrix {

private:

int mat[2][2];

public:

void set\_matrix() {

cout << "Enter 2x2 matrix elements:\n";

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

cin >> mat[i][j];

}

void print\_matrix() {

cout << "Matrix elements:\n";

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++)

cout << mat[i][j] << " ";

cout << endl;

}

}

};

int main() {

Matrix m;

m.set\_matrix();

m.print\_matrix();

return 0;

}

Output: Enter 2x2 matrix elements:

2 4 5 6

Matrix elements:

2 4

5 6

Program 4:

#include <iostream>

using namespace std;

class RecursionExample {

public:

int factorial(int n) {

if (n <= 1) return 1;

return n \* factorial(n - 1);

}

};

int main() {

RecursionExample obj;

int num;

cout << "Enter a number: ";

cin >> num;

cout << "Factorial: " << obj.factorial(num) << endl;

return 0;

}

Output: Enter a number: 3

Factorial: 6

Program 5:

#include <iostream>

using namespace std;

class OverloadExample {

public:

int add(int a, int b) { return a + b; }

double add(double a, double b) { return a + b; }

};

int main() {

OverloadExample obj;

cout << "Sum (int): " << obj.add(5, 3) << endl;

cout << "Sum (double): " << obj.add(5.5, 3.3) << endl;

return 0;

}

Output: Sum (int): 8

Sum (double): 8.8