**EASY**

1. **Program**

#include <iostream>

using namespace std;

class Person {

protected:

string name;

int age;

public:

void getData() {

cout << "Enter name: ";

cin >> name;

cout << "Enter age: ";

cin >> age;

}

void displayData() {

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

cout << "Age: " << age << endl;

}

};

class Student : public Person {

private:

int rollNumber;

public:

void getStudentData() {

getData();

cout << "Enter Roll Number: ";

cin >> rollNumber;

}

void displayStudentData() {

displayData();

cout << "Roll Number: " << rollNumber << endl;

}

};

int main() {

Student s;

cout << "Enter student details: " << endl;

s.getStudentData();

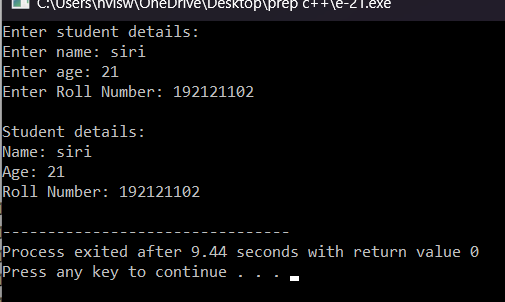
cout << "\nStudent details: " << endl;

s.displayStudentData();

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Base {

private:

int x;

public:

void setData(int val) {

x = val;

}

int getData() {

return x;

}

};

class Derived : private Base {

public:

void setDataDerived(int val) {

setData(val);

}

int getDataDerived() {

return getData();

}

};

int main() {

Derived d;

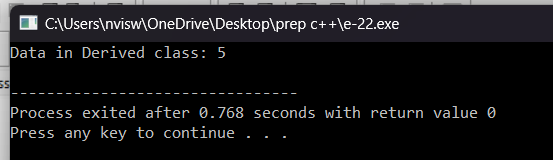
d.setDataDerived(5);

cout << "Data in Derived class: " << d.getDataDerived() << endl;

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

int main() {

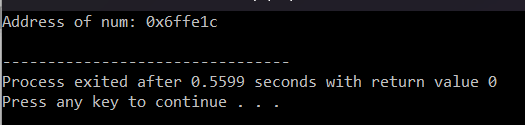
int num = 10;

cout << "Address of num: " << &num << endl;

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Animal {

public:

void speak() {

cout << "Animal sound" << endl;

}

};

class Dog : public Animal {

public:

void speak() {

cout << "Woof!" << endl;

}

};

class Cat : public Animal {

public:

void speak() {

cout << "Meow!" << endl;

}

};

int main() {

Animal animal;

Dog dog;

Cat cat;

animal.speak();

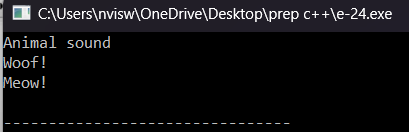
dog.speak();

cat.speak();

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Shape {

protected:

float dimension;

public:

void getDimension() {

cin >> dimension;

}

virtual float calculateArea() = 0;

};

class Circle : public Shape {

public:

float calculateArea() {

return 3.14 \* dimension \* dimension;

}

};

class Square : public Shape {

public:

float calculateArea() {

return dimension \* dimension;

}

};

int main() {

Square square;

Circle circle;

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

square.getDimension();

cout << "Area of square: " << square.calculateArea() << endl;

cout << "\nEnter radius of the circle: ";

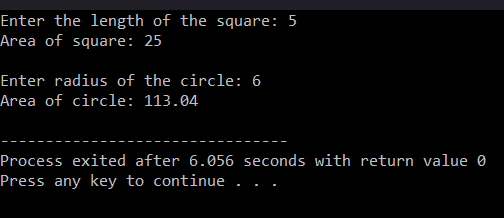
circle.getDimension();

cout << "Area of circle: " << circle.calculateArea() << endl;

return 0;

}

**Output**

****

**6. Program**

#include <iostream>

using namespace std;

class Employee {

public:

int id;

string name;

float salary;

Employee(int id, string name, float salary) {

this->id = id;

this->name = name;

this->salary = salary;

}

void display() {

cout << id << " " << name << " " << salary << endl;

}

};

int main() {

Employee e1 = Employee(101, "Sonoo", 890000);

Employee e2 = Employee(102, "Nakul", 59000);

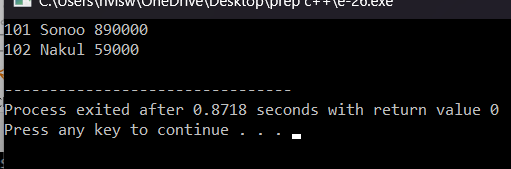
e1.display();

e2.display();

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

int main() {

int arr[5];

cout << "Enter 5 integers: ";

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

cin >> arr[i];

}

cout << "Addresses of elements:" << endl;

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

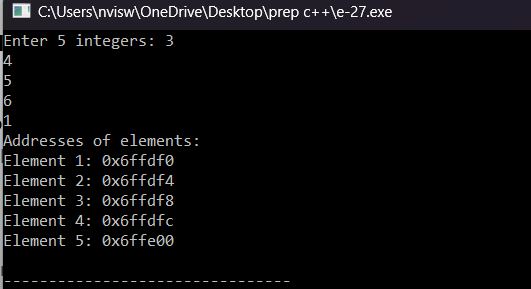
cout << "Element " << i + 1 << ": " << &arr[i] << endl;

}

return 0;

}

**Output**

****

1. **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()

{

std::cout<<"The value of a is : "<<a<<std::endl;

std::cout<<"The value of b is : "<<b<<std::endl;

cout<<"Addition of a and b is : "<<a+b;

}

};

int main()

{

C c;

c.get\_a(10);

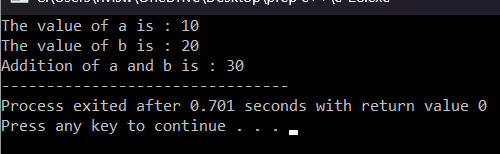
c.get\_b(20);

c.display();

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Rectangle {

private:

int length;

int width;

public:

void setDimensions(int l, int w) {

length = l;

width = w;

}

int area() {

return length \* width;

}

};

int main() {

Rectangle rectArr[3];

for (int i = 0; i < 3; ++i) {

int l, w;

cout << "Enter length and width of rectangle " << i + 1 << ": ";

cin >> l >> w;

rectArr[i].setDimensions(l, w);

}

cout << "Areas of rectangles:" << endl;

for (int i = 0; i < 3; ++i) {

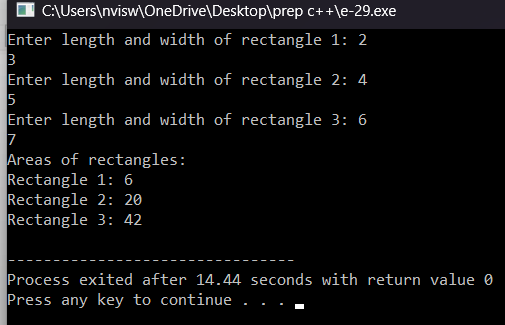
cout << "Rectangle " << i + 1 << ": " << rectArr[i].area() << endl;

}

return 0;

}

**Output**

****

**10.Program**

#include <iostream>

using namespace std;

class Point {

public:

int x;

int y;

Point(int x, int y) {

this->x = x;

this->y = y;

}

void print() {

cout << "(" << x << ", " << y << ")" << endl;

}

};

int main() {

Point point(1, 2);

Point\* pointer = &point;

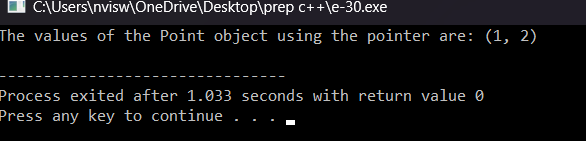
cout << "The values of the Point object using the pointer are: ";

pointer->print();

return 0;

}

**Output**

****

**MEDIUM**

1. **Program**

#include <iostream>

using namespace std;

class Shape {

protected:

float dimension;

public:

void getDimension() {

cin >> dimension;

}

};

class Circle : public Shape {

public:

float calculateArea() {

return 3.14 \* dimension \* dimension;

}

};

class Rectangle : public Shape {

protected:

float breadth;

public:

void getBreadth() {

cin >> breadth;

}

float calculateArea() {

return dimension \* breadth;

}

};

class Cylinder : public Rectangle {

public:

float calculateVolume() {

return calculateArea() \* dimension;

}

};

int main() {

Circle circle;

Rectangle rectangle;

Cylinder cylinder;

cout << "Enter the radius of the circle: ";

circle.getDimension();

cout << "Area of circle: " << circle.calculateArea() << endl;

cout << "\nEnter length and breadth of the rectangle: ";

rectangle.getDimension();

rectangle.getBreadth();

cout << "Area of rectangle: " << rectangle.calculateArea() << endl;

cout << "\nEnter the radius and height of the cylinder: ";

cylinder.getDimension();

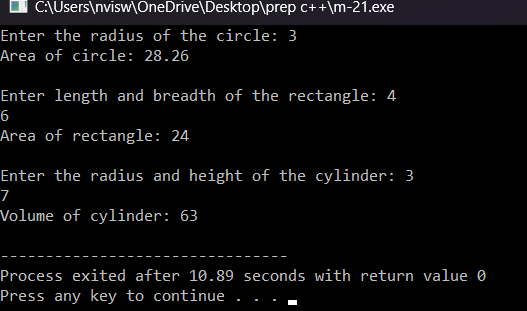
cylinder.getBreadth();

cout << "Volume of cylinder: " << cylinder.calculateVolume() << endl;

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Employee {

private:

string name;

int id;

public:

void getData() {

cout << "Enter name: ";

cin >> name;

cout << "Enter ID: ";

cin >> id;

}

void displayData() {

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

cout << "ID: " << id << endl;

}

};

int main() {

const int numEmployees = 3;

Employee employees[numEmployees];

cout << "Enter details of " << numEmployees << " employees:" << endl;

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

cout << "Employee " << i + 1 << ":" << endl;

employees[i].getData();

}

cout << "\nDetails of employees:" << endl;

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

cout << "Employee " << i + 1 << ":" << endl;

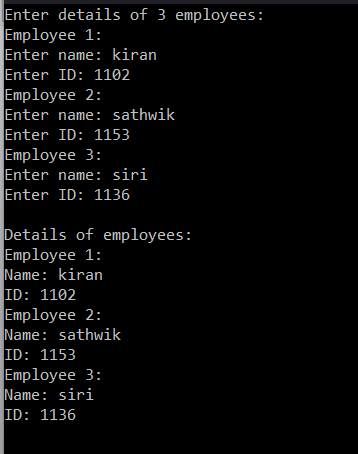
employees[i].displayData();

}

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Number {

protected:

int num;

public:

void getNumber() {

cout << "Enter a number: ";

cin >> num;

}

};

class Square : public Number {

public:

int calculateSquare() {

return num \* num;

}

};

class Cube : public Number {

public:

int calculateCube() {

return num \* num \* num;

}

};

int main() {

Square square;

Cube cube;

square.getNumber();

cout << "Square of the number: " << square.calculateSquare() << endl;

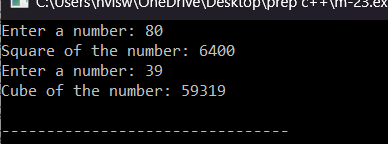
cube.getNumber();

cout << "Cube of the number: " << cube.calculateCube() << endl;

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

int main() {

int num1, num2, num3;

int \*ptr1 = &num1, \*ptr2 = &num2, \*ptr3 = &num3;

cout << "Enter three numbers: ";

cin >> num1 >> num2 >> num3;

int greatest = \*ptr1;

if (\*ptr2 > greatest) {

greatest = \*ptr2;

}

if (\*ptr3 > greatest) {

greatest = \*ptr3;

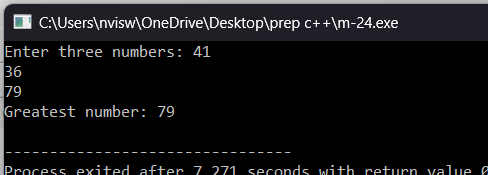
}

cout << "Greatest number: " << greatest << endl;

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

int main() {

int size;

cout << "Enter the size of the array: ";

cin >> size;

int \*arr = new int[size];

cout << "Enter " << size << " elements:" << endl;

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

cin >> \*(arr + i);

}

cout << "\nDisplaying elements using pointer notation:" << endl;

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

cout << \*(arr + i) << " ";

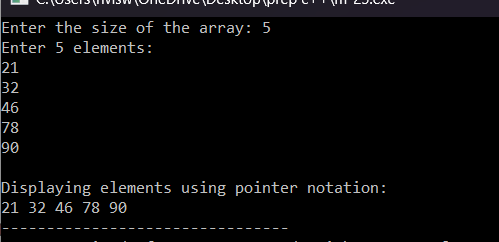
}

delete[] arr;

return 0;

}

**Output**

****

**HARD**

1. **Program**

#include <iostream>

#include <string>

using namespace std;

class Customer {

protected:

string name;

int accountNumber;

public:

Customer(string name, int accountNumber) : name(name), accountNumber(accountNumber) {}

virtual void displayDetails() {

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

cout << "Account Number: " << accountNumber << endl;

}

};

class Account : public Customer {

protected:

double balance;

public:

Account(string name, int accountNumber, double balance) : Customer(name, accountNumber), balance(balance) {}

virtual void deposit(double amount) {

balance += amount;

cout << "Amount deposited successfully." << endl;

}

virtual void withdraw(double amount) {

if (amount > balance) {

cout << "Insufficient balance." << endl;

} else {

balance -= amount;

cout << "Amount withdrawn successfully." << endl;

}

}

virtual void displayBalance() {

cout << "Balance: " << balance << endl;

}

};

int main() {

string name = "John Doe";

int accountNumber = 123456;

double balance = 1000.00;

Account account(name, accountNumber, balance);

int choice;

do {

cout << "\nMenu:" << endl;

cout << "1. Display Details" << endl;

cout << "2. Deposit" << endl;

cout << "3. Withdraw" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

account.displayDetails();

break;

case 2:

double depositAmount;

cout << "Enter amount to deposit: ";

cin >> depositAmount;

account.deposit(depositAmount);

break;

case 3:

double withdrawAmount;

cout << "Enter amount to withdraw: ";

cin >> withdrawAmount;

account.withdraw(withdrawAmount);

break;

case 4:

cout << "Exiting..." << endl;

break;

default:

cout << "Invalid choice." << endl;

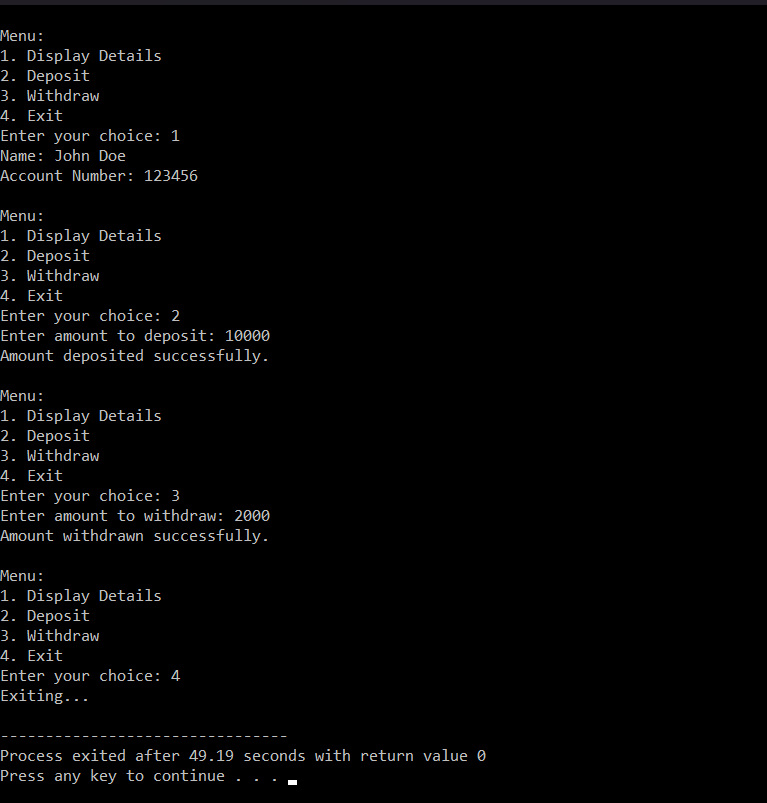
}

} while (choice != 4);

return 0;

}

**Output**

****

1. **Program**

#include<iostream>

using namespace std;

class Student {

public:

int roll;

char sname[100];

char cname[100];

void input() {

cout << "Enter Your Roll Number:";

cin>>roll;

cin.get();

cout << "Enter Your Name:";

cin.getline(sname, 100);

cout << "Enter College Name:";

cin.getline(cname, 100);

}

};

class College : public Student {

public:

void display() {

cout << "Your Roll is :" << roll;

cout << "\nYour Name is :" << sname;

cout << "\nCollege Name is :" << cname;

}

};

int main() {

College obj;

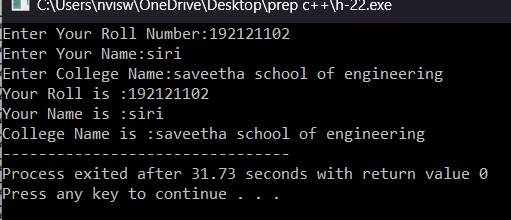
obj.input();

obj.display();

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Employee {

protected:

string name;

int employeeId;

public:

void setEmployeeDetails(string n, int id) {

name = n;

employeeId = id;

}

void displayEmployee() {

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

cout << "Employee ID: " << employeeId << endl;

}

};

class Salary {

protected:

float basicSalary;

public:

void setSalaryDetails(float salary) {

basicSalary = salary;

}

void displaySalary() {

cout << "Basic Salary: " << basicSalary << endl;

}

};

class EmployeeSalary : public Employee, public Salary {

public:

float calculateTotalSalary() {

return basicSalary + 5000;

}

void displayEmployeeSalary() {

displayEmployee();

displaySalary();

cout << "Total Salary: " << calculateTotalSalary() << endl;

}

};

int main() {

EmployeeSalary emp;

emp.setEmployeeDetails("Jane Doe", 1001);

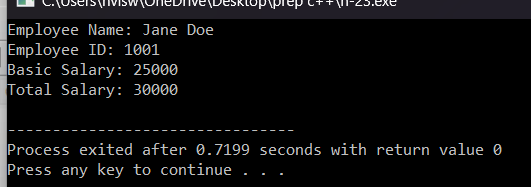
emp.setSalaryDetails(25000.0);

emp.displayEmployeeSalary();

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

void sortArray(int \*arr, int size) {

for (int i = 0; i < size - 1; ++i) {

for (int j = 0; j < size - i - 1; ++j) {

if (\*(arr + j) > \*(arr + j + 1)) {

int temp = \*(arr + j);

\*(arr + j) = \*(arr + j + 1);

\*(arr + j + 1) = temp;

}

}

}

}

int main() {

int size;

cout << "Enter the size of the array: ";

cin >> size;

int \*arr = new int[size];

cout << "Enter " << size << " elements:" << endl;

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

cin >> \*(arr + i);

}

sortArray(arr, size);

cout << "\nSorted array in ascending order:" << endl;

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

cout << \*(arr + i) << " ";

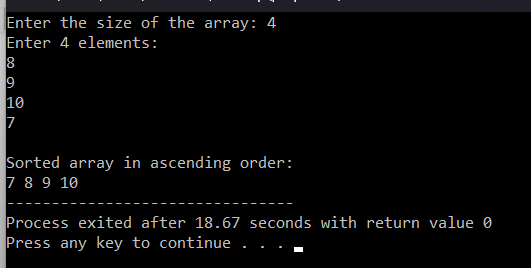
}

delete[] arr;

return 0;

}

**Output**

****

1. **Program**

#include <iostream>

using namespace std;

class Patient {

protected:

string name;

int age;

public:

void setPatientDetails(string n, int a) {

name = n;

age = a;

}

void displayPatientDetails() {

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

cout << "Patient Age: " << age << endl;

}

};

class MedicalRecord : public Patient {

protected:

string diagnosis;

string treatment;

public:

void setMedicalRecord(string d, string t) {

diagnosis = d;

treatment = t;

}

void displayMedicalRecord() {

cout << "Diagnosis: " << diagnosis << endl;

cout << "Treatment: " << treatment << endl;

}

};

int main() {

MedicalRecord patient;

patient.setPatientDetails("John Doe", 35);

patient.setMedicalRecord("Common Cold", "Prescribed rest and fluids");

cout << "Patient Details:" << endl;

patient.displayPatientDetails();

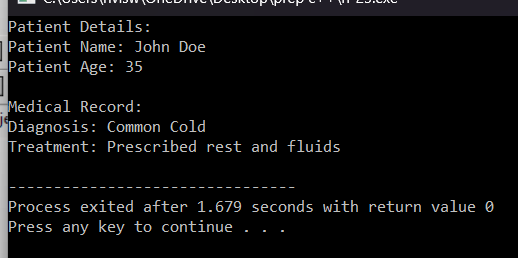
cout << "\nMedical Record:" << endl;

patient.displayMedicalRecord();

return 0;

}

**Output**

****