1. Aim: To perform insertion sort.

Code:

#include <iostream>

using namespace std;

int main(){

int a[10];

for(int i=0;i<10;i++)

cin >>a[i];

for(int i=1;i<10;i++){

int j=i;

while(a[j]>a[j-1]&&j>0){

int t=a[j];

a[j]=a[j-1];

a[j-1]=t;

j--;

}

for(j=0;j<10;j++)

cout<<a[j]<<"\t";

cout<<"\n";

}

}

1. Aim: To perform bubble sort.

Code:

#include <iostream>

using namespace std;

int main(){

int a[10];

for(int i=0;i<10;i++)

cin >>a[i];

for(int i=9;i>=0;i--){

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

if(a[j]>a[j+1]){

int t=a[j];

a[j]=a[j+1];

a[j+1]=t;

}

}

for(int j=0;j<10;j++)

cout<<a[j]<<"\t";

cout<<"\n";

}

}

1. Aim: To perform function overriding and overloading

Code:

#include <iostream>

using namespace std;

// Base class

class Base {

public:

// Virtual function to allow overriding

virtual int sum(int a, int b) {

return a + b;

}

// Overloaded functions for sum

int sum(int a, int b, int c) {

return a + b + c;

}

int sum(int a, int b, int c, int d) {

return a + b + c + d;

}

};

// Derived class

class Derived : public Base {

public:

// Overriding the sum function

int sum(int a, int b) override {

cout << "Derived class sum called." << endl;

return a + b;

}

};

int main() {

int choice;

cout << "Choose an operation:" << endl;

cout << "1. Sum of two numbers (overridden method)" << endl;

cout << "2. Sum of three numbers (overloaded method)" << endl;

cout << "3. Sum of four numbers (overloaded method)" << endl;

cin >> choice;

Derived derivedObj;

Base\* basePtr = &derivedObj;

if (choice == 1) {

int a, b;

cout << "Enter two integers: ";

cin >> a >> b;

// Call overridden method

cout << "Result: " << basePtr->sum(a, b) << endl;

} else if (choice == 2) {

int a, b, c;

cout << "Enter three integers: ";

cin >> a >> b >> c;

// Call overloaded method

cout << "Result: " << derivedObj.sum(a, b, c) << endl;

} else if (choice == 3) {

int a, b, c, d;

cout << "Enter four integers: ";

cin >> a >> b >> c >> d;

// Call overloaded method

cout << "Result: " << derivedObj.sum(a, b, c, d) << endl;

} else {

cout << "Invalid choice!" << endl;

}

return 0;

}

1. Aim: To print array index reference.

Code:

#include <iostream>

using namespace std;

int main(){

int n;

cin>>n;

int\* a=new int[n];

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

cin>>a[i];

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

cout<<&a[i]<<"\t";

}

1. Quick sort

#include<bits/stdc++.h>

using namespace std;

int partion(int arr[],int low,int high){

int pivot = arr[low];

int i = low;

int j = high;

while(i<j){

while(arr[i]<pivot&&i<=high-1){

i++;

}

while(arr[j]>=pivot&&j>low+1){

j--;

}

if(i<j){

swap(arr[i],arr[j]);

}

}

swap(arr[low],arr[j]);

return j;

}

void quicksort(int arr[],int low,int high){

if(low<high){

int pIndex = partion(arr,low,high);

quicksort(arr,low,pIndex-1);

quicksort(arr,pIndex+1,high);

}

}

int main(){

int n;

cout<<"Enter size of array: ";

cin>>n;

int arr[n];

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

cin>>arr[i];

}

quicksort(arr,0,n-1);

cout<<"sorted array: ";

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

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

}

}