**Lab – 2**

Array of objects, Passing and returning objects, Static and const

1. **Write a program to pass an object as an argument and return the object from a function.**
   1. **Use pass-by-value.**

#include<iostream>

using namespace std;

class test

{

public:

int num1;

void getnum();

void display();

test sum(test,test);

}C1,C2,C3;

void test::getnum(){

cout<<"Enter numbers: ";

cin>>num1;

}

test test::sum(test C1,test C2){

C3.num1=C1.num1+C2.num1;

return C3;

}

void test::display(){

cout<<"The sum is: "<<num1;

}

int main(){

C1.getnum();

C2.getnum();

C3=C3.sum(C1,C2);

C3.display();

return 0;

}

* 1. **Use pass-by-address.**

#include<iostream>

using namespace std;

class test

{

public:

int num1;

void getnum();

void display();

test sum(test&,test&);

}C1,C2,C3;

void test::getnum(){

cout<<"Enter numbers: ";

cin>>num1;

}

test test::sum(test &C1,test &C2){

C3.num1=C1.num1+C2.num1;

return C3;

}

void test::display(){

cout<<"The sum is: "<<num1;

}

int main(){

C1.getnum();

C2.getnum();

C3=C3.sum(C1,C2);

C3.display();

return 0;

}

1. **Write a program using Array of Objects to display area of multiple rectangles.**

#include<iostream>

using namespace std;

class rect{

public:

int l;

int b;

int area;

void display();

void getdim();

};

void rect::getdim(){

cout<<"Enter length and breadth: ";

cin>>l>>b;

}

void rect::display(void){

cout<<"Area is: "<<l\*b<<endl;

}

int main(){

rect obj[5];

int i=0;

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

obj[i].getdim();

}

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

cout<<i+1<<". ";

obj[i].display();

}

Return 0;

}

1. **Create code snippets that illustrates the use of scope resolution operator (: :) for the following uses:**
   1. **Member functions defined outside class**
   2. **To access a global variable when there is a local variable with same name**

#include<iostream>

int num1=5;

using namespace std;

class test{

private:

int num1;

public:

void getnum();

void display();

};

void test::getnum(){

cout<<"Enter number: "<<endl;

cin>>num1;

}

void test::display(){

cout<<"Number is: "<<::num1;

}

int main(){

test obj;

obj.getnum();

obj.display();

return 0;

}

* 1. **To access a class’s static variables**

#include<iostream>

using namespace std;

class test{

private:

int static num;

public:

void display();

void getnum();

};

int test::num=9;

void test::display(){

cout<<"Number is: "<<test::num;

}

int main(){

test obj;

obj.display();

return 0;

}

* 1. **Including in-built libraries**

#include<iostream>

int main(){

std::cout<<"Hello world";

return 0;

}

1. **Write a program to define function *cube()* as inline for calculating cube of a number.**

#include<iostream>

using namespace std;

class test{

public:

int side;

void getside();

void cube();

};

void test::getside(){

cout<<"Enter side: ";

cin>>side;

}

inline void test::cube(){

cout<<"Cube is: "<<side\*side\*side;

}

int main(){

test obj;

obj.getside();

obj.cube();

return 0;

}

1. **Write a program to declare *static* public member variable, global and local variable with the same name. Initialize and display their contents.**

#include<iostream>

using namespace std;

int num1=9;

class test{

private:

int num1=5;

public:

void display(){

cout<<"local variable is: "<<num1<<endl;

cout<<"global variable is: "<<::num1;

};

};

int main(){

test obj;

obj.display();

return 0;

}

1. **Implement *static* member function using a C++ program.**

#include<iostream>

using namespace std;

class test{

int static num;

public:

void static display(){

cout<<"Number is: "<<num;

}

};

int test::num=10;

int main(){

test::display();

return 0;

}

1. **Write a C++ program to declare *const* member function and attempt any operation within it.**

#include<iostream>

using namespace std;

class test{

int num1;

static int num2,num3;

public:

void getnum();

void display(test) const;

};

int test::num2=2;

int test::num3=3;

void test::display(test obj) const{

cout<<obj.num2+obj.num3;

}

int main(){

test obj;

obj.display(obj);

return 0;

}