Assignment oops 2

Ans1. #include <iostream>

using namespace std;

#define rows 3

#define cols 3

class demo{

private:

int A[rows][cols];

public:

void get\_data(){

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

{

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

{

cin>>A[i][j];

}

}

}

void show\_data(){

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

{

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

{

cout<<A[i][j]<<"\t";

}

cout<<endl;

}

}

demo operator \*(demo obj){

demo temp;

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

{

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

{

temp.A[i][j]=A[i][j]\*obj.A[i][j];

}

}

return temp;

}

};

int main(){

demo d1,d2,d3;

cout<<"Enter data for first matrix.."<<endl;

d1.get\_data();

cout<<"Enter data for second matrix.."<<endl;

d2.get\_data();

cout<<"First matrix.."<<endl;

d1.show\_data();

cout<<"second matrix.."<<endl;

d2.show\_data();

d3=d1\*d2;

cout<<"third matrix.."<<endl;

d3.show\_data();

}

Ans2. #include <iostream>

#include <string>

class Student {

public:

Student(const std::string& name, int age, const std::string& city) {

this->name = name;

this->age = age;

this->city = city;

}

void displayStudentInfo() {

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

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

std::cout << "City: " << city << std::endl;

}

private:

std::string name;

int age;

std::string city;

};

class Transport {

public:

Transport(const Student& student, const std::string& route, int busFee) : student(student) {

this->route = route;

this->busFee = busFee;

}

void displayDetails() {

student.displayStudentInfo(); // Call base class function to display student info

std::cout << "Route: " << route << std::endl;

std::cout << "Bus Fee: " << busFee << std::endl;

}

private:

Student student; // Composition: Transport HAS-A Student

std::string route;

int busFee;

};

int main() {

// Create a Student object

Student student1("John Doe", 18, "New York");

// Create a Transport object using the Student object

Transport transport(student1, "School Bus - Route 10", 30);

// Display complete student details including transport info

transport.displayDetails();

return 0;

}

Ans 3.

#include <iostream>

using namespace std;

class Test {

public:

// Default constructor initializes all members to 0

Test() {

num1 = num2 = num3 = 0;

}

// get\_data() function takes variable number of arguments

void get\_data(int arg1, int arg2 = 0, int arg3 = 0) {

num1 = arg1;

num2 = arg2;

num3 = arg3;

}

// show\_data() function displays the values of member variables

void show\_data() {

std::cout << "num1: " << num1 << ", num2: " << num2 << ", num3: " << num3 << std::endl;

}

private:

int num1;

int num2;

int num3;

};

int main() {

// Create three objects of the Test class

Test obj1, obj2, obj3;

// Invoke get\_data() member function with different arguments

obj1.get\_data(10);

obj2.get\_data(20, 30);

obj3.get\_data(40, 50, 60);

// Display the values of all the objects

std::cout << "Object 1:" << std::endl;

obj1.show\_data();

std::cout << "Object 2:" << std::endl;

obj2.show\_data();

std::cout << "Object 3:" << std::endl;

obj3.show\_data();

return 0;

}

Ans4.

#include <iostream>

using namespace std;

class BCA\_OOPS {

private:

int var1;

int var2;

public:

// Default constructor

BCA\_OOPS() {

var1 = 0;

var2 = 0;

}

// Parameterized constructor

BCA\_OOPS(int v1, int v2) {

this->var1 = v1;

this->var2 = v2;

}

// Function to display member variables

void displayValues() {

cout << "Var1: " << var1 << endl;

cout << "Var2: " << var2 << endl;

}

// Function to copy data of one object to another

void copyData(const BCA\_OOPS& obj) {

this->var1 = obj.var1;

this->var2 = obj.var2;

}

// Function to decrement values

void decrementValues() {

var1 -= (var1 >= 0 && var1 <= 10)? 2 : 5;

var2 -= (var2 >= 0 && var2 <= 10)? 2 : 5;

}

// Function to display member variables (not a member of the class)

friend void displayValuesExternally(const BCA\_OOPS& obj);

};

// Function to display member variables (not a member of the class)

void displayValuesExternally(const BCA\_OOPS& obj) {

cout << "Var1: " << obj.var1 << endl;

cout << "Var2: " << obj.var2 << endl;

}

int main() {

// Initializing using default constructor

BCA\_OOPS obj1;

cout << "Object 1 (Default Constructor):" << endl;

obj1.displayValues();

cout << endl;

// Initializing using parameterized constructor

BCA\_OOPS obj2(15, 7);

cout << "Object 2 (Parameterized Constructor):" << endl;

obj2.displayValues();

cout << endl;

// Using this pointer inside member function

obj2.decrementValues();

cout << "Object 2 after decrementing values:" << endl;

obj2.displayValues();

cout << endl;

// Copying data of object 2 into object 1

obj1.copyData(obj2);

cout << "Object 1 after copying data from object 2:" << endl;

obj1.displayValues();

cout << endl;

// Displaying member variables externally

cout << "Displaying member variables of object 1 externally:" << endl;

displayValuesExternally(obj1);

cout << endl;

return 0;

}