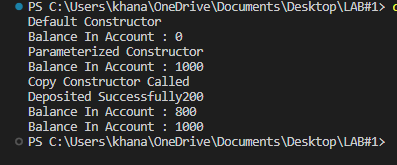
**DS LAB # 1 K230607 BSCS-3F**

**Date : 22 August, 2024**

**QUESTION # 01**

****

#include<iostream>

using namespace std;

class BankAccount{

    private:

    double balance;

    public:

    BankAccount(){

        balance = 0.0;

        cout << "Default Constructor" << endl;

    }

    BankAccount( double val ){

        balance = val;

        cout << "Parameterized Constructor" << endl;

    }

    BankAccount( const BankAccount& other ){

        if( this != &other ){

            balance = other.balance;

        }

        cout << "Copy Constructor Called" << endl;

    }

    void Deduct( double val ){

        if( balance - val < 0  || val <0 ){

            cout << "Not Enough Balance" << endl;

        }

        else{

            balance -= val;

            cout << "Deposited Successfully" << val << endl;

        }

    }

    void display(){

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

    }

};

int main(){

    BankAccount account1;

    account1.display();

    BankAccount account2(1000);

    account2.display();

    BankAccount account3 = account2;

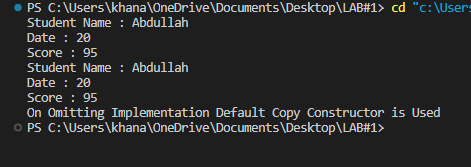
    account3.Deduct(200);

    account3.display();

    account2.display();

}

**QUESTION # 2**

****

#include<iostream>

using namespace std;

class Exam{

    private:

    string std\_name;

    int date;

    int score;

    public:

    void SetName( string name){

        std\_name = name;

    }

    void SetDate( int date ){

        this->date = date;

    }

    void SetScore( int sc ){

        score = sc;

    }

    void display(){

        cout << "Student Name : " << std\_name << endl;

        cout << "Date : " << date << endl;

        cout << "Score : " << score << endl;

    }

};

int main(){

    Exam obj1;

    obj1.SetDate(20);

    obj1.SetName("Abdullah");

    obj1.SetScore(95);

    obj1.display();

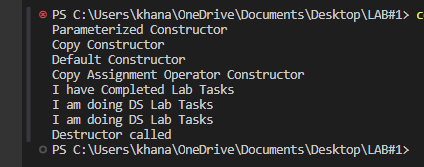
    Exam obj2 = obj1;

    obj2.display();

    cout << "On Omitting Implementation Default Copy Constructor is Used" << endl;

}

**QUESTION # 3**

****

#include<bits/stdc++.h>

using namespace std;

class Document{

    private:

    char\* text;

    public:

    Document(){

        cout << "Default Constructor" << endl;

    }

    Document( const char\* data ){

        text = new char [strlen(data)+1];

        strcpy( text,data );

        cout << "Parameterized Constructor" << endl;

    }

    void Change( const char\* data ){

        delete[] text;

        strcpy( text,data );

    }

    ~Document(){

        cout << "Destructor called" << endl;

        delete[] text;

    }

    Document( const Document& other ){

        if( this != &other){

              text = new char [strlen( other.text )+1];

              strcpy( text,other.text);

        }

        cout << "Copy Constructor" << endl;

    }

    Document& operator = ( const Document& other ){

        if( this != &other){

            delete[] text;

            text = new char [strlen( other.text )+1];

            strcpy( text,other.text);

        }

        cout << "Copy Assignment Operator " << endl;

        return \*this;

    }

    void display(){

        cout << text << endl;

    }

};

int main(){

    Document obj1( "I am doing DS Lab Tasks" );

    Document obj2 = obj1;

    Document obj3;

    obj3 = obj1;

    obj1.Change("I have Completed Lab Tasks");

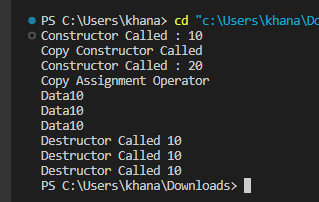
    obj1.display();

    obj2.display();

    obj3.display();

}

**IN CLASS TASKS**

****

#include<bits/stdc++.h>

using namespace std;

class myClass{

    private:

        int\* data;

    public:

    myClass(int value){

        data = new int;

        \*data = value;

        cout << "Constructor Called : " << \*data << endl;

    }

    ~myClass(){

        cout << "Destructor Called " << \*data << endl;

        delete data;

    }

    myClass( const myClass &other ){

        data = new int( \*other.data );

        cout << "Copy Constructor Called " << endl;

    }

    myClass &operator = ( const myClass &other ){

        if( this == &other ){

            return \*this;

        }

        delete data;

        data = new int ( \*other.data );

        cout << "Copy Assignment Operator " << endl;

        return \*this;

    }

    void display(){

        cout << "Data" << \*data << endl;

    }

};

int main(){

    myClass obj1(10);

    myClass obj2 = obj1;

    myClass obj3(20);

    obj3 = obj1;

    obj1.display();

    obj2.display();

    obj3.display();

}

#include<iostream>

#include"myclass.cpp"

using namespace std;

using namespace N;

int main(){

    my\_class mc;

    mc.do\_something();

}

#include "myclass.h"

#include<iostream>

using namespace std;

using namespace N;

void my\_class::do\_something(){

    cout << "Hello world";

}

**#ifndef MY\_CLASS\_H //include guards**

**#define MY\_CLASS\_H**

**namespace N{**

**class my\_class{**

**public:**

**void do\_something();**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**THE END**