OOP Lab – 04

In lab done: Q1,Q2  
After lab done: Q3,Q4,Q5

Name: M. Muzammil Siddiqui

Roll no: 23K-2001

**Q1.**

//23K2001 Muzammil

#include<iostream>

using namespace std;

class bankAccount{

    private:

    int accNum;

    string accHolder;

    float bal;

    public:

    void setNum(int num)

    {

        accNum = num;

    }

    void setHol(string name)

    {

        accHolder = name;

    }

    void setBal(int balance)

    {

        bal = balance;

    }

    int getNum()

    {

        return accNum;

    }

    string getHol()

    {

        return accHolder;

    }

    int getBal()

    {

        return bal;

    }

    bankAccount(int num, string name, float balance)

    {

        accNum = num;

        accHolder = name;

        bal = balance;

    }

    void deposit(int amt)

    {

        bal+=amt;

        cout<<"Amount was deposited succesfully."<<endl;

        cout<<"Balance: "<<bal;

    }

    void withdraw(int amt)

    {

        if(bal>amt)

        {

            bal-=amt;

            cout<<"Amount was withdrawed succesfully."<<endl;

            cout<<"Balance: "<<bal<<endl;

        }

        else

        cout<<"Not enough balance is present."<<endl;

    }

    ~bankAccount()

    {

        cout<<"\n\nDestructor called, memory released!"<<endl;

    }

};

int main()

{

    bankAccount fastian(230000,"student",5000);

    cout<<"Values passed through parameterized constructor: \n"<<endl;

    cout<<"Account number: "<<fastian.getNum()<<endl;

    cout<<"Holder name: "<<fastian.getHol()<<endl;

    cout<<"Balance: "<<fastian.getBal()<<endl;

    int num;

    string name;

    float balance;

    cout<<"\nInput account number: ";

    cin>>num;

    cout<<"Input account holder name: ";

    cin>>name;

    cout<<"Input balance: ";

    cin>>balance;

    fastian.setNum(num);

    fastian.setHol(name);

    fastian.setBal(balance);

    cout<<"\nValues set by user: \n"<<endl;

    cout<<"Account number: "<<fastian.getNum()<<endl;

    cout<<"Holder name: "<<fastian.getHol()<<endl;

    cout<<"Balance: "<<fastian.getBal()<<endl;

    float amount;

    int ch;

    do{

        cout<<"\n1. Deposit\n2. Withdraw\n0. Exit"<<endl;

        cin>>ch;

        switch(ch)

        {

            case 0:

            return 0;

            case 1:

                {

                cout<<"\nInput amount to deposit: ";

                cin>>amount;

                fastian.deposit(amount);

                break;

                }

            case 2:

            {

            cout<<"\nInput amount to withdraw: ";

            cin>>amount;

            fastian.withdraw(amount);

            break;

            }

            default:

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

        }

    }while(ch!=0);

    return 0;

}

**Q2.**

//232001 Muzammil

#include<iostream>

#include<math.h>

using namespace std;

class point{

    private:

    int x,y,z;

    public:

    void setX(int m)

    {

        x = m;

    }

    void setY(int n)

    {

        y = n;

    }

    void setZ(int p)

    {

        z = p;

    }

    int getX()

    {

        return x;

    }

    int getY()

    {

        return y;

    }

    int getZ()

    {

        return z;

    }

    point(int m,int n,int p)

    {

        x=m;

        y=n;

        z=p;

    }

    float org\_distance()

    {

        return sqrt(x\*x+y\*y+z\*z);

    }

    point()

    {

        x=0;

        y=0;

        z=0;

    }

    ~point()

    {

        cout<<"\nDestructor called, memory released."<<endl;

    }

};

int main()

{

    point img(5,8,11);

    point img2;

    cout<<"Values passed by default constructor: \n"<<endl;

    cout<<"X: "<<img2.getX()<<endl;

    cout<<"Y: "<<img2.getY()<<endl;

    cout<<"Z: "<<img2.getZ()<<endl;

    cout<<"Values passed through parameterized constructor: \n"<<endl;

    cout<<"X: "<<img.getX()<<endl;

    cout<<"Y: "<<img.getY()<<endl;

    cout<<"Z: "<<img.getZ()<<endl;

    cout<<"Distance from origin of passed point: "<<endl;

    cout<<img.org\_distance()<<" units."<<endl;

    cout<<"\nInput your own points: "<<endl;

    int a,b,c;

    cin>>a>>b>>c;

    img2.setX(a);

    img2.setY(b);

    img2.setZ(c);

    cout<<"\nValues passed by user using setters: \n"<<endl;

    cout<<"X: "<<img2.getX()<<endl;

    cout<<"Y: "<<img2.getY()<<endl;

    cout<<"Z: "<<img2.getZ()<<endl;

    cout<<"Distance from origin of your point: "<<endl;

    cout<<img2.org\_distance()<<" units."<<endl;

    return 0;

}

**Q3.**

//23K2001 Muzammil

#include<iostream>

using namespace std;

class sale{

    private:

    int saleID,qty;

    string item\_name;

    public:

    sale()

    {

        saleID = 123456;

        qty = 5;

        item\_name = "sample";

    }

    sale(int id, int q, string item)

    {

        cout<<"\*\*\*Constructor implementing user input..\*\*\*"<<endl;

        saleID = id;

        qty = q;

        item\_name = item;

    }

    sale(string item,int id, int q)

    {

        cout<<"\n\*\*\*Parameterized constructor initializing variables..\*\*\*"<<endl;

        saleID = id;

        qty = q;

        item\_name = item;

    }

    sale(const sale &sample)

    {

        saleID = sample.saleID;

        qty = sample.qty;

        item\_name = sample.item\_name;

    }

    ~sale()

    {

        cout<<"\nDestructor called, memory released."<<endl;

    }

    int getID()

    {

        return saleID;

    }

    int getQuantity()

    {

        return qty;

    }

    string getItem()

    {

        return item\_name;

    }

};

int main()

{

    sale shop1;

    cout<<"Shop1 sales values (Set by default constructor): "<<endl;

    cout<<"ID: "<<shop1.getID()<<endl;

    cout<<"Quantity: "<<shop1.getQuantity()<<endl;

    cout<<"Item name: "<<shop1.getItem()<<endl;

    int id, q;

    string item;

    cout<<"\nInput values for your shop sales: "<<endl;

    cout<<"Input id: ";

    cin>>id;

    cout<<"Input quantity: ";

    cin>>q;

    cout<<"Input name: ";

    cin>>item;

    sale shop2(id,q,item);

    cout<<"\nShop2 sales values (Set by user input): "<<endl;

    cout<<"ID: "<<shop2.getID()<<endl;

    cout<<"Quantity: "<<shop2.getQuantity()<<endl;

    cout<<"Item name: "<<shop2.getItem()<<endl;

    sale shop3("T-shirts",55493,10);

    cout<<"\nShop3 sales values (Passed by parameterized constructor): "<<endl;

    cout<<"ID: "<<shop3.getID()<<endl;

    cout<<"Quantity: "<<shop3.getQuantity()<<endl;

    cout<<"Item name: "<<shop3.getItem()<<endl;

    sale shop4 = shop2;

    cout<<"\nShop4 sales values (Copied from shop2): "<<endl;

    cout<<"ID: "<<shop4.getID()<<endl;

    cout<<"Quantity: "<<shop4.getQuantity()<<endl;

    cout<<"Item name: "<<shop4.getItem()<<endl;

    return 0;

}

**Q4.**

//23K2001 Muzammil

#include<iostream>

using namespace std;

class Distance{

    private:

    float feet, inch;

    public:

    Distance()

    {

        feet = 100;

        inch = 100;

    }

    void userinput()

    {

        float d;

        cout<<"Input distance in meters: ";

        cin>>d;

        feet = d\*3.281;

        inch = d\*39.37;

    }

    float displayMeter(float x)

    {

        return feet\*0.3048;

    }

    float displayFeet(float x)

    {

        return feet;

    }

    float getFeet()

    {

        return feet;

    }

    float getInch()

    {

        return inch;

    }

    ~Distance()

    {

        cout<<"\nObject was killed, memory released.."<<endl;

    }

};

int main()

{

    Distance route;

    cout<<"Default values set by constructor: "<<endl;

    cout<<"Feet: "<<route.getFeet()<<endl;

    cout<<"Inch: "<<route.getInch()<<endl;

    cout<<"\nSet your values: "<<endl;

    route.userinput();

    cout<<"\nValues now are: "<<endl;

    cout<<"Feet: "<<route.getFeet()<<endl;

    cout<<"Inch: "<<route.getInch()<<endl;

    return 0;

}

**Q5.**

//23K2001 Muzammil

#include<iostream>

using namespace std;

class phone{

    private:

    string area,exc,num;

    public:

    phone(string telephone)

    {

        area = telephone.substr(0,3);

        exc = telephone.substr(3,4);

        num = telephone.substr(7,4);

    }

    void displayDetails()

    {

        cout<<"Your area code is:- "<<area<<endl;

        cout<<"Your exchange code is:- "<<exc<<endl;

        cout<<"Your consumer no# is:- "<<num<<endl;

    }

};

int main()

{

    cout<<"Input your contact number (11 digit): ";

    string p;

    cin>>p;

    phone me(p);

    me.displayDetails();

    return 0;

}