

## Introduction

ProgrammingAdvices.com

مميزات الكورس



كل التوفيق للجميع  
محمد أبوهدهود

هذا الكورس هو كورس تطبيقي بحث لغرس جميع المفاهيم التي تعلمناها في الكورس السابق ، من واقع عملي لتحقيق اكبر استفاده والارتقاء بمستواكم البرمجي للافضل كما عودناكم في الكورسات السابقة عن طريق عمل مشاريع صغيرة واستخدام اساليب برمجة متطوره.

يجب ان تكون انهيت جميع الكورسات بالاخضر

- سلسلة حلول متقدمه للخوارزميات المستوى الاول
- سلسلة مقدمة للبرمجة بلغة C++ المستوى الاول
- سلسلة اساسيات مهمة لكل مبرمج المستوى الاول
- سلسلة حلول مشاكل الخوارزميات المستوى الثاني
- سلسلة مقدمة للبرمجة بلغة C++ المستوى الثاني
- سلسلة اساسيات مهمة لكل مبرمج المستوى الثاني
- سلسلة حلول مشاكل الخوارزميات والمشكل المستوى الرابع
- سلسلة حلول مشاكل الخوارزميات والمشكل المستوى الثالث
- سلسلة البرمجة الكائنية Object Oriented OOP تطبيقات مشروع صغير
- سلسلة البرمجة الكائنية Object Oriented OOP مفاهيم واساليب
- هيكل البيانات Data Structure DS. المستوى الاول
- البرمجة الكائنية Object Oriented OOP تطبيقات مشروع صغير
- اساسيات مهمة لكل مبرمج المستوى الثاني

ProgrammingAdvices.com

Mohammed Abu-Hadoud

Telegram Group for This Course

<https://t.me/+eoBRYv6zhLRmMGJk>

## Utility Library: Requirements.

قبل كده عملنا كلاس ال `string` وكلاس ال `date` هنعمل دلوقتي `utility library` هناخد كل ال `functions` اللي عملناها قبل كده زي ال `random number` ونحطهم في `static` ونخليةم كلهم كلام

```
#include <iostream>
#include "clsUtil.h"

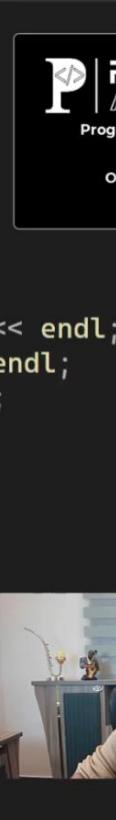
int main()
{
    clsUtil::Srand();

    cout << clsUtil::RandomNumber(1, 10) << endl;
    cout << clsUtil::GetRandomCharacter(clsUtil::MixChars) << endl;
    cout << clsUtil::GenerateWord(clsUtil::MixChars, 8) << endl;
    cout << clsUtil::GenerateKey(clsUtil::MixChars) << endl;
    clsUtil::GenerateKeys(10, clsUtil::MixChars);

    cout << "\n";

    //Swap Int
    int x = 10, y = 20;
    cout << x << " " << y << endl;
    clsUtil::Swap(x, y);
    cout << x << " " << y << endl << endl;

    //Swap double
    double a = 10.5, b = 20.5;
```



في ال `capital letter` عمل `enum` عشان يختار مابين انه يطلعه او `small letter` او `digits` او حاجه منهم اللي هيا ال `mix` في ال `generate word` انت بتختار عايز طول الكلمة قد ايه وبيبدأ بطلعلك حروف عشوائيه بعدد الحروف

```
//Swap double
double a = 10.5, b = 20.5;
cout << a << " " << b << endl;
clsUtil::Swap(a, b);
cout << a << " " << b << endl;

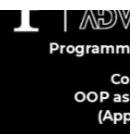
//Swap String
string s1="Ali", s2="Ahmed";
cout << s1 << " " << s2 << endl;
clsUtil::Swap(s1, s2);
cout << s1 << " " << s2 << endl;

//Swap Dates
clsDate d1(1, 10, 2022),
cout << d1.DateToString();
clsUtil::Swap(d1, d2);
cout << d1.DateToString();
```

```
//Shuffle Array
    I
//int array
int Arr1[5] = { 1,2,3,4,5
clsUtil::ShuffleArray(Arr1, 5);
cout << "\nArray after shuffle:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr1[i] << endl;
}

```

Copy



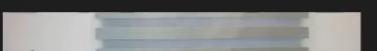
```
//string array
string Arr2[5] = { "Ali", "Fadi", "Ahmed", "Qaser", "Sara"
clsUtil::ShuffleArray(Arr2, 5);
cout << "\nArray after shuffle:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr2[i] << endl;
}

int Arr3[5];
clsUtil::FillArrayWithRandomNumbers(Arr3, 5, 20, 50);
cout << "\nArray after fill:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr3[i] << endl;
}
```



لكلام  
يأخذ الـ `array` ويعايز الأرقام تكون من كام `fillArrayWithRandomNumbers`

```
string Arr4[5];
clsUtil::FillArrayWithRandomWords(Arr4, 5, clsUtil::MixChars, 8)
cout << "\nArray after fill:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr4[i] << endl;
}
```



```

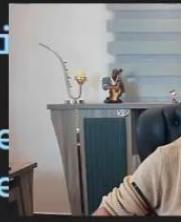
string Arr5[5];
clsUtil::FillArrayWithRandomKeys(Arr5, 5, clsUtil::enCharType::SamallLetter);
cout << "\nArray after filling keys:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr5[i] << endl;
}

cout << "\nText1 " << clsUtil::Tabs(5) << "Text2\n";

const short EncryptionKey = 2; //this is the key.

string TextAfterEncryption, TextAfterDecryption;
string Text = "Mohammed Abu-Hadhoud";
TextAfterEncryption = clsUtil::EncryptText(Text, EncryptionKey);
TextAfterDecryption = clsUtil::DecryptText(TextAfterEncryption, EncryptionKey);

```



ال tabs بتدبها عدد ال اللي عايز تطبعها (ال tabs اللي هيا ال "\t")  
وكمي بقى على باقي ال functions

```

#pragma once
#include <iostream>
#include <string>
#include "clsDate.h"

using namespace std;

class clsUtil
{
public:
    enum enCharType {
        SamallLetter = 1, CapitalLetter = 2,
        Digit = 3, MixChars = 4, SpecialCharacter = 5
    };

    static void Srand() { ... }
}

```

### Utility Library: Solution

//ProgrammingAdvices.com  
//Mohammed Abu-Hadhoud

```

#pragma once
#include <iostream>
#include <string>
#include "clsDate.h"

using namespace std;

class clsUtil

```

ده ال utility class

```
{  
  
public:  
    enum enCharType {  
        SamallLetter = 1, CapitalLetter = 2,  
        Digit = 3, MixChars = 4, SpecialCharacter = 5  
    };  
  
    static void Srand()  
    {  
        //Seeds the random number generator in C++, called only once  
        srand((unsigned)time(NULL));  
    }  
  
    static int RandomNumber(int From, int To)  
    {  
        //Function to generate a random number  
        int randNum = rand() % (To - From + 1) + From;  
        return randNum;  
    }  
  
    static char GetRandomCharacter(enCharType CharType)  
    {  
  
        //updated this method to accept mixchars  
        if (CharType == MixChars)  
        {  
            //Capital/Samll/Digits only  
            CharType = (enCharType)RandomNumber(1, 3);  
        }  
  
        switch (CharType)  
        {  
  
            case enCharType::SamallLetter:  
            {  
                return char(RandomNumber(97, 122));  
                break;  
            }  
            case enCharType::CapitalLetter:  
            {  
                return char(RandomNumber(65, 90));  
                break;  
            }  
            case enCharType::SpecialCharacter:  
            {  
                return char(RandomNumber(33, 47));  
                break;  
            }  
            case enCharType::Digit:  
            {  
                return char(RandomNumber(48, 57));  
                break;  
            }  
            default:  
            {  
                return char(RandomNumber(65, 90));  
                break;  
            }  
        }  
    }  
}
```

```

        }

    }

    static string GenerateWord(enCharType CharType, short Length)
    {
        string Word;
        for (int i = 1; i <= Length; i++)
        {
            Word = Word + GetRandomCharacter(CharType);
        }
        return Word;
    }

    static string GenerateKey(enCharType CharType = CapitalLetter)
    {
        string Key = "";
        Key = GenerateWord(CharType, 4) + "-";
        Key = Key + GenerateWord(CharType, 4) + "-";
        Key = Key + GenerateWord(CharType, 4) + "-";
        Key = Key + GenerateWord(CharType, 4);

        return Key;
    }

    static void GenerateKeys(short NumberOfKeys, enCharType CharType)
    {
        for (int i = 1; i <= NumberOfKeys; i++)
        {
            cout << "Key [" << i << "] : ";
            cout << GenerateKey(CharType) << endl;
        }
    }

    static void FillArrayWithRandomNumbers(int arr[100], int arrLength, int From, int To)
    {
        for (int i = 0; i < arrLength; i++)
            arr[i] = RandomNumber(From, To);
    }

    static void FillArrayWithRandomWords(string arr[100], int arrLength, enCharType CharType, short Wordlength)
    {
        for (int i = 0; i < arrLength; i++)
            arr[i] = GenerateWord(CharType, Wordlength);
    }

    static void FillArrayWithRandomKeys(string arr[100], int arrLength, enCharType CharType)

```

```
{  
    for (int i = 0; i < arrLength; i++)  
        arr[i] = GenerateKey(CharType);  
}  
  
static void Swap(int& A, int& B)  
{  
    int Temp;  
  
    Temp = A;  
    A = B;  
    B = Temp;  
}  
  
static void Swap(double& A, double& B)  
{  
    double Temp;  
  
    Temp = A;  
    A = B;  
    B = Temp;  
}  
  
static void Swap(bool& A, bool& B)  
{  
    bool Temp;  
  
    Temp = A;  
    A = B;  
    B = Temp;  
}  
  
static void Swap(char& A, char& B)  
{  
    char Temp;  
  
    Temp = A;  
    A = B;  
    B = Temp;  
}  
  
static void Swap(string& A, string& B)  
{  
    string Temp;  
  
    Temp = A;  
    A = B;  
    B = Temp;  
}  
  
static void Swap(clsDate& A, clsDate& B)  
{  
    clsDate::SwapDates(A, B);  
}  
  
static void ShuffleArray(int arr[100], int arrLength)  
{  
    for (int i = 0; i < arrLength; i++)
```

```
{  
    Swap(arr[RandomNumber(1, arrLength) - 1], arr[RandomNumber(1, arrLength) - 1]);  
}  
  
}  
  
static void ShuffleArray(string arr[100], int arrLength)  
{  
  
    for (int i = 0; i < arrLength; i++)  
    {  
        Swap(arr[RandomNumber(1, arrLength) - 1], arr[RandomNumber(1, arrLength) - 1]);  
    }  
  
}  
  
static string Tabs(short NumberOfTabs)  
{  
    string t = "";  
  
    for (int i = 1; i < NumberOfTabs; i++)  
    {  
        t = t + "\t";  
        cout << t;  
    }  
    return t;  
}  
  
}  
  
static string EncryptText(string Text, short EncryptionKey)  
{  
  
    for (int i = 0; i <= Text.length(); i++)  
    {  
  
        Text[i] = char((int)Text[i] + EncryptionKey);  
    }  
  
    return Text;  
}  
  
}  
  
static string DecryptText(string Text, short EncryptionKey)  
{  
  
    for (int i = 0; i <= Text.length(); i++)  
    {  
  
        Text[i] = char((int)Text[i] - EncryptionKey);  
    }  
    return Text;  
}  
};
```

```

//ProgrammingAdvices.com
//Mohammed Abu-Hadhoud
#pragma warning(disable : 4996)
#pragma once

#include<iostream>
#include<string>
#include "clsString.h"

using namespace std;

class clsDate
{
private:
    short _Day = 1;
    short _Month = 1;
    short _Year = 1900;

public:
    clsDate()
    {
        time_t t = time(0);
        tm* now = localtime(&t);
        _Day = now->tm_mday;
        _Month = now->tm_mon + 1;
        _Year = now->tm_year + 1900;
    }

    clsDate(string sDate)
    {
        vector <string> vDate;
        vDate = clsString::Split(sDate, "/");

        _Day = stoi(vDate[0]);
        _Month = stoi(vDate[1]);
        _Year = stoi(vDate[2]);
    }

    clsDate(short Day, short Month, short Year)
    {
        _Day = Day;
        _Month = Month;
        _Year = Year;
    }

    clsDate(short DateOrderInYear, short Year)
    {
        //This will construct a date by date order in year
        clsDate Date1 = GetDateFromDayOrderInYear(DateOrderInYear, Year);
        _Day = Date1.Day;
        _Month = Date1.Month;
        _Year = Date1.Year;
    }

    void SetDay(short Day) {
        _Day = Day;
    }

    short GetDay() {
        return _Day;
    }
    __declspec(property(get = GetDay, put = SetDay)) short Day;

    void SetMonth(short Month) {
        _Month = Month;
    }
}

```

```

short GetMonth() {
    return _Month;
}
__declspec(property(get = GetMonth, put = SetMonth)) short Month;

void SetYear(short Year) {
    _Year = Year;
}

short GetYear() {
    return _Year;
}
__declspec(property(get = GetYear, put = SetYear)) short Year;

void Print()
{
    cout << DateToString() << endl;
}

static clsDate GetSystemDate()
{
    //system date
    time_t t = time(0);
    tm* now = localtime(&t);

    short Day, Month, Year;

    Year = now->tm_year + 1900;
    Month = now->tm_mon + 1;
    Day = now->tm_mday;

    return clsDate(Day, Month, Year);
}

static bool IsValidDate(clsDate Date)
{
    if (Date.Day < 1 || Date.Day>31)
        return false;

    if (Date.Month < 1 || Date.Month>12)
        return false;

    if (Date.Month == 2)
    {
        if (isLeapYear(Date.Year))
        {
            if (Date.Day > 29)
                return false;
        }
        else
        {
            if (Date.Day > 28)
                return false;
        }
    }

    short DaysInMonth = NumberOfDaysInAMonth(Date.Month, Date.Year);

    if (Date.Day > DaysInMonth)
        return false;

    return true;
}

bool IsValid()
{
    return IsValidDate(*this);
}

static string DateToString(clsDate Date)

```

```

    {
        return to_string(Date.Day) + "/" + to_string(Date.Month) + "/" + to_string(Date.Year);
    }

    string DateToString()
    {
        return DateToString(*this);
    }

    static bool isLeapYear(short Year)
    {

        // if year is divisible by 4 AND not divisible by 100
        // OR if year is divisible by 400
        // then it is a leap year
        return (Year % 4 == 0 && Year % 100 != 0) || (Year % 400 == 0);
    }

    bool isLeapYear()
    {
        return isLeapYear(_Year);
    }

    static short NumberOfDaysInAYear(short Year)
    {
        return isLeapYear(Year) ? 365 : 364;
    }

    short NumberOfDaysInAYear()
    {
        return NumberOfDaysInAYear(_Year);
    }

    static short NumberOfHoursInAYear(short Year)
    {
        return NumberOfDaysInAYear(Year) * 24;
    }

    short NumberOfHoursInAYear()
    {
        return NumberOfHoursInAYear(_Year);
    }

    static int NumberOfMinutesInAYear(short Year)
    {
        return NumberOfHoursInAYear(Year) * 60;
    }

    int NumberOfMinutesInAYear()
    {
        return NumberOfMinutesInAYear(_Year);
    }

    static int NumberOfSecondsInAYear(short Year)
    {
        return NumberOfMinutesInAYear(Year) * 60;
    }

    int NumberOfSecondsInAYear()
    {
        return NumberOfSecondsInAYear();
    }

    static short NumberOfDaysInAMonth(short Month, short Year)
    {

        if (Month < 1 || Month > 12)
            return 0;

        int days[12] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
        return (Month == 2) ? (isLeapYear(Year) ? 29 : 28) : days[Month - 1];
    }
}

```

```

short NumberOfDaysInAMonth()
{
    return NumberOfDaysInAMonth(_Month, _Year);
}

static short NumberOfHoursInAMonth(short Month, short Year)
{
    return NumberOfDaysInAMonth(Month, Year) * 24;
}

short NumberOfHoursInAMonth()
{
    return NumberOfDaysInAMonth(_Month, _Year) * 24;
}

static int NumberOfMinutesInAMonth(short Month, short Year)
{
    return NumberOfHoursInAMonth(Month, Year) * 60;
}

int NumberOfMinutesInAMonth()
{
    return NumberOfHoursInAMonth(_Month, _Year) * 60;
}

static int NumberOfSecondsInAMonth(short Month, short Year)
{
    return NumberOfMinutesInAMonth(Month, Year) * 60;
}

int NumberOfSecondsInAMonth()
{
    return NumberOfMinutesInAMonth(_Month, _Year) * 60;
}

static short DayOfWeekOrder(short Day, short Month, short Year)
{
    short a, y, m;
    a = (14 - Month) / 12;
    y = Year - a;
    m = Month + (12 * a) - 2;
    // Gregorian:
    // 0:sun, 1:Mon, 2:Tue...etc
    return (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;
}

short DayOfWeekOrder()
{
    return DayOfWeekOrder(_Day, _Month, _Year);
}

static string DayShortName(short DayOfWeekOrder)
{
    string arrDayNames[] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

    return arrDayNames[DayOfWeekOrder];
}

static string DayShortName(short Day, short Month, short Year)
{
    string arrDayNames[] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

    return arrDayNames[DayOfWeekOrder(Day, Month, Year)];
}

string DayShortName()
{
    string arrDayNames[] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

    return arrDayNames[DayOfWeekOrder(_Day, _Month, _Year)];
}

```

```

}

static string MonthShortName(short MonthNumber)
{
    string Months[12] = { "Jan", "Feb", "Mar",
                         "Apr", "May", "Jun",
                         "Jul", "Aug", "Sep",
                         "Oct", "Nov", "Dec"
    };

    return (Months[MonthNumber - 1]);
}

string MonthShortName()
{
    return MonthShortName(_Month);
}

static void PrintMonthCalendar(short Month, short Year)
{
    int NumberOfDays;

    // Index of the day from 0 to 6
    int current = DayOfWeekOrder(1, Month, Year);

    NumberOfDays = NumberOfDaysInAMonth(Month, Year);

    // Print the current month name
    printf("\n _____%s_____\\n\\n",
           MonthShortName(Month).c_str());

    // Print the columns
    printf(" Sun Mon Tue Wed Thu Fri Sat\\n");

    // Print appropriate spaces
    int i;
    for (i = 0; i < current; i++)
        printf("   ");

    for (int j = 1; j <= NumberOfDays; j++)
    {
        printf("%5d", j);

        if (++i == 7)
        {
            i = 0;
            printf("\\n");
        }
    }

    printf("\\n _____\\n");
}

void PrintMonthCalendar()
{
    PrintMonthCalendar(_Month, _Year);
}

static void PrintYearCalendar(int Year)
{
    printf("\n _____\\n\\n");
    printf("      Calendar - %d\\n", Year);
    printf(" _____\\n");

    for (int i = 1; i <= 12; i++)
    {
        PrintMonthCalendar(i, Year);
    }
}

```

```

        return;
    }

    void PrintYearCalendar()
    {
        printf("\n _____\n\n");
        printf("     Calendar - %d\n", _Year);
        printf(" _____\n");

        for (int i = 1; i <= 12; i++)
        {
            PrintMonthCalendar(i, _Year);
        }

        return;
    }

    static short DaysFromTheBeginingOfTheYear(short Day, short Month, short Year)
    {

        short TotalDays = 0;

        for (int i = 1; i <= Month - 1; i++)
        {
            TotalDays += NumberOfDaysInAMonth(i, Year);
        }

        TotalDays += Day;

        return TotalDays;
    }

    short DaysFromTheBeginingOfTheYear()
    {

        short TotalDays = 0;

        for (int i = 1; i <= _Month - 1; i++)
        {
            TotalDays += NumberOfDaysInAMonth(i, _Year);
        }

        TotalDays += _Day;

        return TotalDays;
    }

    static clsDate GetDateFromDayOrderInYear(short DateOrderInYear, short Year)
    {

        clsDate Date;
        short RemainingDays = DateOrderInYear;
        short MonthDays = 0;

        Date.Year = Year;
        Date.Month = 1;

        while (true)
        {
            MonthDays = NumberOfDaysInAMonth(Date.Month, Year);

            if (RemainingDays > MonthDays)
            {
                RemainingDays -= MonthDays;
                Date.Month++;
            }
            else
            {
                Date.Day = RemainingDays;
                break;
            }
        }
    }
}

```

```

        }

        return Date;
    }

    void AddDays(short Days)
    {

        short RemainingDays = Days + DaysFromTheBeginingOfTheYear(_Day, _Month, _Year);
        short MonthDays = 0;

        _Month = 1;

        while (true)
        {
            MonthDays = NumberOfDaysInAMonth(_Month, _Year);

            if (RemainingDays > MonthDays)
            {
                RemainingDays -= MonthDays;
                _Month++;

                if (_Month > 12)
                {
                    _Month = 1;
                    _Year++;
                }
            }
            else
            {
                _Day = RemainingDays;
                break;
            }
        }
    }

    static bool IsDate1BeforeDate2(clsDate Date1, clsDate Date2)
    {
        return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ? (Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day < Date2.Day : false)) : false);
    }

    bool IsDateBeforeDate2(clsDate Date2)
    {
        //note: *this sends the current object :-)
        return IsDate1BeforeDate2(*this, Date2);
    }

    static bool IsDate1EqualDate2(clsDate Date1, clsDate Date2)
    {
        return (Date1.Year == Date2.Year) ? ((Date1.Month == Date2.Month) ? ((Date1.Day == Date2.Day) ? true : false) : false) : false;
    }

    bool IsDateEqualDate2(clsDate Date2)
    {
        return IsDate1EqualDate2(*this, Date2);
    }

    static bool IsLastDayInMonth(clsDate Date)
    {

        return (Date.Day == NumberOfDaysInAMonth(Date.Month, Date.Year));
    }

    bool IsLastDayInMonth()

```

```

{
    return IsLastDayInMonth(*this);
}

static bool IsLastMonthInYear(short Month)
{
    return (Month == 12);
}

static clsDate AddOneDay(clsDate Date)
{
    if (IsLastDayInMonth(Date))
    {
        if (IsLastMonthInYear(Date.Month))
        {
            Date.Month = 1;
            Date.Day = 1;
            Date.Year++;
        }
        else
        {
            Date.Day = 1;
            Date.Month++;
        }
    }
    else
    {
        Date.Day++;
    }

    return Date;
}

void AddOneDay()
{
    *this = AddOneDay(*this);
}

static void SwapDates(clsDate& Date1, clsDate& Date2)
{
    clsDate TempDate;
    TempDate = Date1;
    Date1 = Date2;
    Date2 = TempDate;
}

static int GetDifferenceInDays(clsDate Date1, clsDate Date2, bool IncludeEndDay = false)
{
    //this will take care of negative diff
    int Days = 0;
    short SawpFlagValue = 1;

    if (!IsDate1BeforeDate2(Date1, Date2))
    {
        //Swap Dates
        SwapDates(Date1, Date2);
        SawpFlagValue = -1;
    }

    while (IsDate1BeforeDate2(Date1, Date2))
    {
        Days++;
        Date1 = AddOneDay(Date1);
    }

    return IncludeEndDay ? ++Days * SawpFlagValue : Days * SawpFlagValue;
}

```

```

int GetDifferenceInDays(clsDate Date2, bool IncludeEndDay = false)
{
    return GetDifferenceInDays(*this, Date2, IncludeEndDay);
}

static short CalculateMyAgeInDays(clsDate DateOfBirth)
{
    return GetDifferenceInDays(DateOfBirth, clsDate::GetSystemDate(), true);
}
//above no need to have nonstatic function for the object because it does not depend on any data
from it.

static clsDate IncreaseDateByOneWeek(clsDate& Date)
{
    for (int i = 1; i <= 7; i++)
    {
        Date = AddOneDay(Date);
    }

    return Date;
}

void IncreaseDateByOneWeek()
{
    IncreaseDateByOneWeek(*this);
}

clsDate IncreaseDateByXWeeks(short Weeks, clsDate& Date)
{
    for (short i = 1; i <= Weeks; i++)
    {
        Date = IncreaseDateByOneWeek(Date);
    }
    return Date;
}

void IncreaseDateByXWeeks(short Weeks)
{
    IncreaseDateByXWeeks(Weeks, *this);
}

clsDate IncreaseDateByOneMonth(clsDate& Date)
{
    if (Date.Month == 12)
    {
        Date.Month = 1;
        Date.Year++;
    }
    else
    {
        Date.Month++;
    }

    //last check day in date should not exceed max days in the current month
    // example if date is 31/1/2022 increasing one month should not be 31/2/2022, it should
    // be 28/2/2022
    short NumberOfDaysInCurrentMonth = NumberOfDaysInAMonth(Date.Month, Date.Year);
    if (Date.Day > NumberOfDaysInCurrentMonth)
    {
        Date.Day = NumberOfDaysInCurrentMonth;
    }

    return Date;
}

void IncreaseDateByOneMonth()
{
    IncreaseDateByOneMonth(*this);
}

```

```

clsDate IncreaseDateByXDays(short Days, clsDate& Date)
{
    for (short i = 1; i <= Days; i++)
    {
        Date = AddOneDay(Date);
    }
    return Date;
}

void IncreaseDateByXDays(short Days)
{
    IncreaseDateByXDays(Days, *this);
}

clsDate IncreaseDateByXMonths(short Months, clsDate& Date)
{
    for (short i = 1; i <= Months; i++)
    {
        Date = IncreaseDateByOneMonth(Date);
    }
    return Date;
}

void IncreaseDateByXMonths(short Months)
{
    IncreaseDateByXMonths(Months, *this);
}

static clsDate IncreaseDateByOneYear(clsDate& Date)
{
    Date.Year++;
    return Date;
}

void IncreaseDateByOneYear()
{
    IncreaseDateByOneYear(*this);
}

clsDate IncreaseDateByXYears(short Years, clsDate& Date)
{
    Date.Year += Years;
    return Date;
}

void IncreaseDateByXYears(short Years)
{
    IncreaseDateByXYears(Years);
}

clsDate IncreaseDateByOneDecade(clsDate& Date)
{
    //Period of 10 years
    Date.Year += 10;
    return Date;
}

void IncreaseDateByOneDecade()
{
    IncreaseDateByOneDecade(*this);
}

clsDate IncreaseDateByXDecades(short Decade, clsDate& Date)
{
    Date.Year += Decade * 10;
    return Date;
}

void IncreaseDateByXDecades(short Decade)

```

```

    {
        IncreaseDateByXDecades(Decade, *this);
    }

    clsDate IncreaseDateByOneCentury(clsDate& Date)
    {
        //Period of 100 years
        Date.Year += 100;
        return Date;
    }

    void IncreaseDateByOneCentury()
    {
        IncreaseDateByOneCentury(*this);
    }

    clsDate IncreaseDateByOneMillennium(clsDate& Date)
    {
        //Period of 1000 years
        Date.Year += 1000;
        return Date;
    }

    clsDate IncreaseDateByOneMillennium()
    {
        IncreaseDateByOneMillennium(*this);
    }

    static clsDate DecreaseDateByOneDay(clsDate Date)
    {
        if (Date.Day == 1)
        {
            if (Date.Month == 1)
            {
                Date.Month = 12;
                Date.Day = 31;
                Date.Year--;
            }
            else
            {
                Date.Month--;
                Date.Day = NumberOfDaysInAMonth(Date.Month, Date.Year);
            }
        }
        else
        {
            Date.Day--;
        }
    }

    return Date;
}

void DecreaseDateByOneDay()
{
    DecreaseDateByOneDay(*this);
}

static clsDate DecreaseDateByOneWeek(clsDate& Date)
{
    for (int i = 1; i <= 7; i++)
    {
        Date = DecreaseDateByOneDay(Date);
    }

    return Date;
}

void DecreaseDateByOneWeek()
{
    DecreaseDateByOneWeek(*this);
}

```

```

static clsDate DecreaseDateByXWeeks(short Weeks, clsDate& Date)
{
    for (short i = 1; i <= Weeks; i++)
    {
        Date = DecreaseDateByOneWeek(Date);
    }
    return Date;
}

void DecreaseDateByXWeeks(short Weeks)
{
    DecreaseDateByXWeeks(Weeks, *this);
}

static clsDate DecreaseDateByOneMonth(clsDate& Date)
{
    if (Date.Month == 1)
    {
        Date.Month = 12;
        Date.Year--;
    }
    else
        Date.Month--;
}

//last check day in date should not exceed max days in the current month
// example if date is 31/3/2022 decreasing one month should not be 31/2/2022, it should
// be 28/2/2022
short NumberOfDaysInCurrentMonth = NumberOfDaysInAMonth(Date.Month, Date.Year);
if (Date.Day > NumberOfDaysInCurrentMonth)
{
    Date.Day = NumberOfDaysInCurrentMonth;
}

return Date;
}

void DecreaseDateByOneMonth()
{
    DecreaseDateByOneMonth(*this);
}

static clsDate DecreaseDateByXDays(short Days, clsDate& Date)
{
    for (short i = 1; i <= Days; i++)
    {
        Date = DecreaseDateByOneDay(Date);
    }
    return Date;
}

void DecreaseDateByXDays(short Days)
{
    DecreaseDateByXDays(Days, *this);
}

static clsDate DecreaseDateByXMonths(short Months, clsDate& Date)
{
    for (short i = 1; i <= Months; i++)
    {
        Date = DecreaseDateByOneMonth(Date);
    }
    return Date;
}

void DecreaseDateByXMonths(short Months)
{
    DecreaseDateByXMonths(Months, *this);
}

```

```

static clsDate DecreaseDateByOneYear(clsDate& Date)
{
    Date.Year--;
    return Date;
}

void DecreaseDateByOneYear()
{
    DecreaseDateByOneYear(*this);
}

static clsDate DecreaseDateByXYears(short Years, clsDate& Date)
{
    Date.Year -= Years;
    return Date;
}

void DecreaseDateByXYears(short Years)
{
    DecreaseDateByXYears(Years, *this);
}

static clsDate DecreaseDateByOneDecade(clsDate& Date)
{
    //Period of 10 years
    Date.Year -= 10;
    return Date;
}

void DecreaseDateByOneDecade()
{
    DecreaseDateByOneDecade(*this);
}

static clsDate DecreaseDateByXDecades(short Decades, clsDate& Date)
{
    Date.Year -= Decades * 10;
    return Date;
}

void DecreaseDateByXDecades(short Decades)
{
    DecreaseDateByXDecades(Decades, *this);
}

static clsDate DecreaseDateByOneCentury(clsDate& Date)
{
    //Period of 100 years
    Date.Year -= 100;
    return Date;
}

void DecreaseDateByOneCentury()
{
    DecreaseDateByOneCentury(*this);
}

static clsDate DecreaseDateByOneMillennium(clsDate& Date)
{
    //Period of 1000 years
    Date.Year -= 1000;
    return Date;
}

void DecreaseDateByOneMillennium()
{
    DecreaseDateByOneMillennium(*this);
}

```

```

static short IsEndOfWeek(clsDate Date)
{
    return DayOfWeekOrder(Date.Day, Date.Month, Date.Year) == 6;
}

short IsEndOfWeek()
{
    return IsEndOfWeek(*this);
}

static bool IsWeekEnd(clsDate Date)
{
    //Weekends are Fri and Sat
    short DayIndex = DayOfWeekOrder(Date.Day, Date.Month, Date.Year);
    return (DayIndex == 5 || DayIndex == 6);
}

bool IsWeekEnd()
{
    return IsWeekEnd(*this);
}

static bool IsBusinessDay(clsDate Date)
{
    //Weekends are Sun,Mon,Tue,Wed and Thur

/*
    short DayIndex = DayOfWeekOrder(Date.Day, Date.Month, Date.Year);
    return (DayIndex >= 5 && DayIndex <= 4);
*/

//shorter method is to invert the IsWeekEnd: this will save updating code.
return !IsWeekEnd(Date);
}

bool IsBusinessDay()
{
    return IsBusinessDay(*this);
}

static short DaysUntilTheEndOfWeek(clsDate Date)
{
    return 6 - DayOfWeekOrder(Date.Day, Date.Month, Date.Year);
}

short DaysUntilTheEndOfWeek()
{
    return DaysUntilTheEndOfWeek(*this);
}

static short DaysUntilTheEndOfMonth(clsDate Date1)
{
    clsDate EndOfMontDate;
    EndOfMontDate.Day = NumberOfDaysInAMonth(Date1.Month, Date1.Year);
    EndOfMontDate.Month = Date1.Month;
    EndOfMontDate.Year = Date1.Year;

    return GetDifferenceInDays(Date1, EndOfMontDate, true);
}

short DaysUntilTheEndOfMonth()
{
    return DaysUntilTheEndOfMonth(*this);
}

static short DaysUntilTheEndOfYear(clsDate Date1)
{
    clsDate EndOfYearDate;
    EndOfYearDate.Day = 31;
    EndOfYearDate.Month = 12;
}

```

```

        EndOfDayDate.Year = Date1.Year;

        return GetDifferenceInDays(Date1, EndOfDayDate, true);

    }

    short DaysUntilTheEndOfDay()
    {
        return DaysUntilTheEndOfDay(*this);
    }

    //i added this method to calculate business days between 2 days
    static short CalculateBusinessDays(clsDate DateFrom, clsDate DateTo)
    {

        short Days = 0;
        while (IsDate1BeforeDate2(DateFrom, DateTo))
        {
            if (IsBusinessDay(DateFrom))
                Days++;

            DateFrom = AddOneDay(DateFrom);
        }

        return Days;
    }

    static short CalculateVacationDays(clsDate DateFrom, clsDate DateTo)
    {
        /*short Days = 0;
        while (IsDate1BeforeDate2(DateFrom, DateTo))
        {
            if (IsBusinessDay(DateFrom))
                Days++;

            DateFrom = AddOneDay(DateFrom);
        }*/

        return CalculateBusinessDays(DateFrom, DateTo);
    }

    //above method is enough , no need to have method for the object

    static clsDate CalculateVacationReturnDate(clsDate DateFrom, short VacationDays)
    {

        short WeekEndCounter = 0;

        for (short i = 1; i <= VacationDays; i++)
        {

            if (IsWeekEnd(DateFrom))
                WeekEndCounter++;

            DateFrom = AddOneDay(DateFrom);
        }
        //to add weekends
        for (short i = 1; i <= WeekEndCounter; i++)
            DateFrom = AddOneDay(DateFrom);

        return DateFrom;
    }

    static bool IsDate1AfterDate2(clsDate Date1, clsDate Date2)
    {
        return (!IsDate1BeforeDate2(Date1, Date2) && !IsDate1EqualDate2(Date1, Date2));
    }

    bool IsDateAfterDate2(clsDate Date2)
    {
        return IsDate1AfterDate2(*this, Date2);
    }
}

```

```

enum enDateCompare { Before = -1, Equal = 0, After = 1 };

static enDateCompare CompareDates(clsDate Date1, clsDate Date2)
{
    if (IsDate1BeforeDate2(Date1, Date2))
        return enDateCompare::Before;

    if (IsDate1EqualDate2(Date1, Date2))
        return enDateCompare::Equal;

    /* if (IsDate1AfterDate2(Date1, Date2))
       return enDateCompare::After; */

    //this is faster
    return enDateCompare::After;
}

enDateCompare CompareDates(clsDate Date2)
{
    return CompareDates(*this, Date2);
}

};


```

//ProgrammingAdvices.com  
//Mohammed Abu-Hadhoud

```

#include <iostream>
#include "clsUtil.h"

int main()

{
    clsUtil::Srand();
    cout << clsUtil::RandomNumber(1, 10) << endl;
    cout << clsUtil::GetRandomCharacter(clsUtil::CapitalLetter) << endl;
    cout << clsUtil::GenerateWord(clsUtil::MixChars, 8) << endl;
    cout << clsUtil::GenerateKey(clsUtil::MixChars) << endl;
    clsUtil::GenerateKeys(10, clsUtil::MixChars);

    cout << "\n";

    //Swap Int
    int x = 10, y = 20;
    cout << x << " " << y << endl;
    clsUtil::Swap(x, y);
    cout << x << " " << y << endl << endl;

    //Swap double
    double a = 10.5, b = 20.5;
    cout << a << " " << b << endl;
    clsUtil::Swap(a, b);
    cout << a << " " << b << endl << endl;

    //Swap String
    string s1 = "Ali", s2 = "Ahmed";
    cout << s1 << " " << s2 << endl;
    clsUtil::Swap(s1, s2);
    cout << s1 << " " << s2 << endl << endl;

    //Swap Dates

```

دہ جل  
main  
class

```

clsDate d1(1, 10, 2022), d2(1, 1, 2022);
cout << d1.DateToString() << " " << d2.DateToString() << endl;
clsUtil::Swap(d1, d2);
cout << d1.DateToString() << " " << d2.DateToString() << endl;

//Shuffle Array

//int array
int Arr1[5] = { 1,2,3,4,5 };
clsUtil::ShuffleArray(Arr1, 5);
cout << "\nArray after shuffle:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr1[i] << endl;
}

//string array
string Arr2[5] = { "Ali","Fadi","Ahmed","Qasem","Khalid" };
clsUtil::ShuffleArray(Arr2, 5);
cout << "\nArray after shuffle:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr2[i] << endl;
}

int Arr3[5];
clsUtil::FillArrayWithRandomNumbers(Arr3, 5, 20, 50);
cout << "\nArray after fill:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr3[i] << endl;
}

string Arr4[5];
clsUtil::FillArrayWithRandomWords(Arr4, 5, clsUtil::MixChars, 8);
cout << "\nArray after fill:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr4[i] << endl;
}

string Arr5[5];
clsUtil::FillArrayWithRandomKeys(Arr5, 5, clsUtil::MixChars);
cout << "\nArray after filling keys:\n";
for (int i = 0; i < 5; i++)
{
    cout << Arr5[i] << endl;
}

cout << "\nText1 " << clsUtil::Tabs(5) << "Text2\n";

const short EncryptionKey = 2; //this is the key.

string TextAfterEncryption, TextAfterDecryption;
string Text = "Mohammed Abu-Hadoud";
TextAfterEncryption = clsUtil::EncryptText(Text, EncryptionKey);
TextAfterDecryption = clsUtil::DecryptText(TextAfterEncryption, EncryptionKey);

```

```

cout << "\nText Before Encryption : ";
cout << Text << endl;
cout << "Text After Encryption : ";
cout << TextAfterEncryption << endl;
cout << "Text After Decryption : ";
cout << TextAfterDecryption << endl;

system("pause>0");

return 0;
}

```

## Input & Validation Library (Requirements)

```

#ifndef <iostream>
#include "clsInputValidate.h"

int main()
{
    cout << clsInputValidate::IsNumberBetween(5, 1, 10) << endl;
    cout << clsInputValidate::IsNumberBetween(5.5, 1.3, 10.8) << endl;

    cout << clsInputValidate::IsDateBetween(clsDate(),
                                            clsDate(8,12,2022),
                                            clsDate(31, 12, 2022)) << endl;

    cout << clsInputValidate::IsDateBetween(clsDate(),
                                            clsDate(31, 12, 2022),
                                            clsDate(8, 12, 2022)) << endl;

    cout << "\nPlease Enter a Number:\n";
    int x= clsInputValidate::ReadIntNumber("Invalid Number",
    cout << "x=" << x;

```



فيها static functions

والنزم باسمي ال functions عشان لما تستخدمها بعد كده

```

cout << "\nPlease Enter a Number:\n";
int x= clsInputValidate::ReadIntNumber("Invalid Number, Enter again:\n");
cout << "x=" << x;

cout << "\nPlease Enter a Number between 1 and 5:\n";
int y = clsInputValidate::ReadIntNumberBetween(1,5,"Number is not within range");
cout << "y=" << y;

```

باتاخد منك رسالة الخطأ readIntNumber

لو عايز اليوزر يدخل قيمة من رقم كذا لرقم كذا ReadIntNumberBetween

```

"\nPlease Enter a Double Number between 1 and 5:\n";
b = clsInputValidate::ReadDblNumberBetween(1, 5, "Number is not within range, enter again:");
"b=" << b;

endl << clsInputValidate::IsValidDate([clsDate(35,12,2022))] << endl;

"pause>0");

```



## Input & Validation Library (Solution)

```
#include <iostream>
#include "clsInputValidate.h"

int main()

{
    cout << clsInputValidate::IsNumberBetween(5, 1, 10) << endl;
    cout << clsInputValidate::IsNumberBetween(5.5, 1.3, 10.8) << endl;

    cout << clsInputValidate::IsDateBetween(clsDate(),
        clsDate(8, 12, 2022),
        clsDate(31, 12, 2022)) << endl;

    cout << clsInputValidate::IsDateBetween(clsDate(),
        clsDate(31, 12, 2022),
        clsDate(8, 12, 2022)) << endl;

    cout << "\nPlease Enter a Number:\n";
    int x = clsInputValidate::ReadIntNumber("Invalid Number, Enter again:\n");
    cout << "x=" << x;

    cout << "\nPlease Enter a Number between 1 and 5:\n";
    int y = clsInputValidate::ReadIntNumberBetween(1, 5, "Number is not within range, enter again:\n");
    cout << "y=" << y;

    cout << "\nPlease Enter a Double Number:\n";
    double a = clsInputValidate::ReadDblNumber("Invalid Number, Enter again:\n");
    cout << "a=" << a;

    cout << "\nPlease Enter a Double Number between 1 and 5:\n";
    double b = clsInputValidate::ReadDblNumberBetween(1, 5, "Number is not within range, enter again:\n");
    cout << "b=" << b;

    cout << endl << clsInputValidate::IsValidDate(clsDate(35, 12, 2022)) << endl;

    system("pause>0");

    return 0;
}
```

```
//ProgrammingAdvces.com
//Mohammed Abu-Hadhoud
```

```
#pragma once
#include <iostream>
#include <string>
#include "clsString.h"
#include "clsDate.h"

class clsInputValidate
{

public:

    static bool IsNumberBetween(short Number, short From, short To)
    {
        if (Number >= From && Number <= To)
            return true;
        else

```

دہ ال  
main

دہ کلاس  
ال  
validate

```

        return false;
    }

    static bool IsNumberBetween(int Number, int From, int To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsNumberBetween(float Number, float From, float To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsNumberBetween(double Number, double From, double To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsDateBetween(clsDate Date, clsDate From, clsDate To)
    {
        //Date>=From && Date<=To
        if ((clsDate::IsDate1AfterDate2(Date, From) || clsDate::IsDate1EqualDate2(Date, From))
            &&
            (clsDate::IsDate1BeforeDate2(Date, To) || clsDate::IsDate1EqualDate2(Date,
To)))
        )
        {
            return true;
        }

        //Date>=To && Date<=From
        if ((clsDate::IsDate1AfterDate2(Date, To) || clsDate::IsDate1EqualDate2(Date, To))
            &&
            (clsDate::IsDate1BeforeDate2(Date, From) || clsDate::IsDate1EqualDate2(Date,
From)))
        )
        {
            return true;
        }

        return false;
    }

    static int ReadIntNumber(string ErrorMessage = "Invalid Number, Enter again\n")
    {
        int Number;
        while (!(cin >> Number)) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << ErrorMessage;
    }
}

```

```

        }

        return Number;
    }

    static int ReadIntNumberBetween(int From, int To, string ErrorMessage = "Number is not
within range, Enter again:\n")
    {
        int Number = ReadIntNumber();

        while (!IsNumberBetween(Number, From, To))
        {
            cout << ErrorMessage;
            Number = ReadIntNumber();
        }
        return Number;
    }

    static double ReadDblNumber(string ErrorMessage = "Invalid Number, Enter again\n")
    {
        double Number;
        while (!(cin >> Number)) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << ErrorMessage;
        }
        return Number;
    }

    static double ReadDblNumberBetween(double From, double To, string ErrorMessage =
"Number is not within range, Enter again:\n")
    {
        double Number = ReadDblNumber();

        while (!IsNumberBetween(Number, From, To)) {
            cout << ErrorMessage;
            Number = ReadDblNumber();
        }
        return Number;
    }

    static bool IsValideDate(clsDate Date)
    {
        return clsDate::IsValidDate(Date);
    }
}

```

انا حطيت الاكواود الجديد عشان مااكررش في الاكواود وازود مساحه عالفاضي

### Lesson #01 - Find Client

ده شكل الملف اللي فيه بيانات العملاء

العميل بيكون من ( first name, last name , email, phone number, ( account number, pin code, balance

Clients - Notepad

File Edit Format View Help

```
Mohammed##Abu-Hadhoud##Mo@gmail.com###0799997886##A101##1234##5000.00000
Khaili##Ahmed##Khalil##8928982##A102##1234##8748.00000
Adli##Haddad##Adli@gmail.com##8983883##A103##1234##555.00000
Jamil##Adli##Jamil@gmail.com##23123123##A104##1234##8912.00000
Lina##Loay##Lina@gmail.com##1234##A07##125##423.00000
Mazem##Kareem##Mazin@gmail.com##898234##A106##1234##8002.00000
Lama##Karmi##Lama@gmail.com##8983883##A107##1234##1234.00000
Gandi##Omran##Ghandi@gmail.com##1234##A108##1234##7000.00000
Hamed##Abbas##H@Gmail.com##8838##A109##1234##5000.00000
Hilal##Mazin##A@A.com##123443##A110##1234##9000.00000
Majida##Yousef##Majida##1234##A111##1234##717.00000
```

واحنا هنحوّل ال object ده ل line الكلاس person فيه الاتي (first name , last name , email , phone) وعامله fullname و ال decspec property get و property set constructor

```
#pragma once
#include <iostream>
#include <string>
using namespace std;

class clsPerson
{
private:
    string _FirstName;
    string _LastName;
    string _Email;
    string _Phone;

public:
    clsPerson( string FirstName, string LastName, string Email, string Phone)
    {
        _FirstName = FirstName;
        _LastName = LastName;
        _Email = Email;
        _Phone = Phone;
    }
}
```



بعد كده عندي كلاس ال person ببورث من كلاس ال clients

```

#pragma once
#include <iostream>
#include <string>
#include "clsPerson.h"
#include "clsString.h"
#include <vector>
#include <fstream>

using namespace std;

class clsBankClient : public clsPerson
{
private:

    enum enMode { EmptyMode = 0, UpdateMode = 1 };
    enMode _Mode;

    string _AccountNumber;
    string _PinCode;
    float _AccountBalance;

```

ال mode ده الحاله اللي بيكون فيها ال object يالما بيكون فاضي يالما update وفي الدروس اللي جايه هنضيف عليها في الدرس ده كل اللي هنعمله هو find client وبنعملهم (account number , pincode , account balance) وبعرف فيه person وبيوديهاله والمتغيرات بتاعت ال mode وبياخد ال constructor get set وعاملين client

```

static clsBankClient _ConvertLinetoClientObject(string Line, string Client);

static string _ConverClientObjectToLine(clsBankClient Client, string Line);

static clsBankClient _GetEmptyClientObject() { ... }

public:
    Copyright 2022

public:

    clsBankClient(enMode Mode, string FirstName, string LastName,
        string Email, string Phone, string AccountNumber, string PinCode,
        float AccountBalance) :
        clsPerson(FirstName, LastName, Email, Phone)

    {
        _Mode = Mode;
        _AccountNumber = AccountNumber;
        _PinCode = PinCode;
        _AccountBalance = AccountBalance;
    }
}

```

```
bool IsEmpty()
{
    return (_Mode == enMode::EmptyMode);
}

string AccountNumber()
{
    return _AccountNumber;
}

void SetPinCode(string PinCode)
{
    _PinCode = PinCode;
}

string GetPinCode()
{
    return _PinCode;
}

__declspec(property(get = GetPinCode, put = SetPinCode)) string PinCode;

void SetAccountBalance(float AccountBalance)
{
    _AccountBalance = AccountBalance;
}
```

Copyright 2022

```

void Print()
{
    cout << "\nClient Card:";  

    cout << "\n-----";  

    cout << "\nFirstName : " << FirstName;  

    cout << "\nLastName : " << LastName;  

    cout << "\nFull Name : " << FullName();  

    cout << "\nEmail : " << Email;  

    cout << "\nPhone : " << Phone;  

    cout << "\nAcc. Number : " << _AccountNumber;  

    cout << "\nPassword : " << _PinCode;  

    cout << "\nBalance : " << _AccountBalance;  

    cout << "\n-----\n";
}

```

في ال person و عندي هنا كمان print

```

static clsBankClient Find(string AccountNumber) { ... }

static clsBankClient Find(string AccountNumber, string PinCode) { ... }

static bool IsClientExist(string AccountNumber) { ... }

```

```

#include <iostream>
#include "clsPerson.h"
#include "clsBankClient.h"

int main()
{
    clsBankClient Client1 = clsBankClient::Find("A101");
    Client1.Print(); [highlighted]

    clsBankClient Client2 = clsBankClient::Find("A101", "1234");
    Client2.Print();

    system("pause>0");
    return 0;
}

```

في ال main ما عنديش اكواود عندي سطرين بس

الفكرة هنا اني عاوز ادور على `client` معين ولو لقيته بمل Yi ببياناته في ال `object` ولو مالقيتهوش  
هرجعه فاضي

عشان كده هعمل `method` من النوع `static` اسمها `find` وبتاخد رقم الحساب وواحده تانيه بتبحث  
عن العميل عن طريق رقم الحساب وال `pincode` بتوعه

```
static clsBankClient Find(string AccountNumber)
{
    vector<clsBankClient> vClients;

    fstream MyFile;
    MyFile.open("Clients.txt", ios::in); //read Mode

    if (MyFile.is_open())
    {
        string Line;
        while (getline(MyFile, Line))
        {
            clsBankClient Client = _ConvertLinetoClientObject(Line);
            if (Client.AccountNumber() == AccountNumber)
            {
                MyFile.close();
                return Client;
            }
            vClients.push_back(Client);
        }
    }
}
```

Copyright 2022

```
    MyFile.close();
```

```
}
```

```
    return _GetEmptyClientObject();
}
```

Copyright 2022

```
static clsBankClient _ConvertLinetoClientObject(string Line, string Seperator = "#/#")
{
    vector<string> vClientData;
    vClientData = clsString::split(Line, Seperator);
    static inline std::vector<std::string> clsString::split(std::string S1, std::string Delim)
    +1 overload
    Search Online
    vClientData[1], vClientData[2],
    vClientData[3], vClientData[4], vClientData[5], stod(vCli
}
```



```
}

static clsBankClient _GetEmptyClientObject()
{
    return clsBankClient(enMode::EmptyMode, "", "", "", "", "", "", "", "", 0);
}
```

enum clsBankClient::enMode::EmptyMode = 0  
Search Online

```

static bool IsClientExist(string AccountNumber)
{
    clsBankClient Client1 = clsBankClient::Find(AccountNumber);
    return (!Client1.IsEmpty());
}

```

نزل الكود عندك وافهمه وعيده بآيدك تاني

<pre> #pragma once #include&lt;iostream&gt; using namespace std;  class clsPerson { private:     string _FirstName;     string _LastName;     string _Email;     string _Phone;  public:     //constructors     clsPerson(string FirstName,string LastName,string Email,string Phone) {         this-&gt;_FirstName = FirstName;         this-&gt;_LastName = LastName;         this-&gt;_Email = Email;         this-&gt;_Phone = Phone;     }      //setters     void set_FirstName (string FirstName) { this-&gt;_FirstName = FirstName; }     void set_LastName (string LastName) { this-&gt;_LastName = LastName; }     void set_Email(string Email) { this-&gt;_Email = Email; }     void set_Phone(string Phone) { this-&gt;_Phone = Phone; }      //getters      string get_FirstName() { return this-&gt;_FirstName; }     string get_LastName() { return this-&gt;_LastName; }     string get_Email() { return this-&gt;_Email; }     string get_Phone() { return this-&gt;_Phone; }     string get_FullName() { return this-&gt;_FirstName + " " + this-&gt;_LastName; }      //declspec     _declspec(property(get = get_FirstName, put = set_FirstName))string FirstName;     _declspec(property(get = get_LastName, put = set_LastName))string LastName;     _declspec(property(get = get_Email, put = set_Email))string Email;     _declspec(property(get = get_Phone, put = set_Phone))string Phone;      //person functions     void Print()     {         cout &lt;&lt; "\nInfo:" ;         cout &lt;&lt; "\n_____";         cout &lt;&lt; "\nFirstName: " &lt;&lt; this-&gt;_FirstName;         cout &lt;&lt; "\nLastName : " &lt;&lt; this-&gt;_LastName;         cout &lt;&lt; "\nFull Name: " &lt;&lt; this-&gt;get_FullName();     } } </pre>	<p>ده كود ال person</p>
--	-----------------------------

```

        cout << "\nEmail : " << this->_Email;
        cout << "\nPhone : " << this->_Phone;
        cout << "\n_____ \n";

    }

};

#pragma once
#include<iostream>
#include"clsPerson.h"
#include"clsString.h"
#include <vector>
#include <string>
#include <fstream>
using namespace std;

class clsBankClient: public clsPerson
{
private:
    enum enMode {EmptyMode=0,UpdateMode=1};

    //variables
    enMode _Mode;
    string _AccountNumber;
    string _PinCode;
    float _AccountBalance;

    //other functions
    static clsBankClient _ConvertLineToClientObject(string Line,string Seperator="#/#") {
        vector<string> vClientData;

        vClientData=clsString::Split(Line,Seperator);

        return clsBankClient(enMode::UpdateMode, vClientData[0], vClientData[1]
                           , vClientData[2], vClientData[3], vClientData[4], vClientData[5],
                           stod(vClientData[6]));
    }

    static string _ConvertClientObjecttoLine(clsBankClient Client,string Seperator="#/#") {
        return Client.get_FirstName()
            +Seperator+Client.get_LastName()
            +Seperator+Client.get_Email()
            +Seperator+Client.get_Phone()
            +Seperator+Client._AccountNumber
            +Seperator+Client._PinCode
            +Seperator+to_string(Client._AccountBalance);
    }

    static clsBankClient _GetEmptyClientObject() {
        return clsBankClient(enMode::EmptyMode,"","","","", "",0);
    }

public:

    //constructors
    clsBankClient(enMode Mode,string FirstName, string LastName, string Email, string Phone
    , string AccountNumber,string PinCode,float AccountBalance)
    :clsPerson(FirstName,LastName,Email,Phone){

        this->_Mode = Mode;
        this->_AccountNumber = AccountNumber;
    }
}

```

ده کود ال  
client

```

        this->_PinCode = PinCode;
        this->_AccountBalance = AccountBalance;
    }

    //setters
    void set_Mode(enMode Mode) { this->_Mode = Mode; }
    void set_AccountNumber(string AccountNumber) { this->_AccountNumber = AccountNumber; }
    void set_PinCode(string PinCode) { this->_PinCode = PinCode; }
    void set_AccountBalance(float AccountBalance) { this->_AccountBalance = AccountBalance; }

    //getters
    enMode get_Mode() { return this->_Mode; }
    string get_AccountNumber() { return this->_AccountNumber; }
    string get_PinCode() { return this->_PinCode; }
    float get_AccountBalance() { return this->_AccountBalance; }

    //declspec
    _declspec(property(get = get_Mode, put = set_Mode))enMode Mode;
    _declspec(property(get = get_AccountNumber, put = set_AccountNumber))string
AccountNumber;
    _declspec(property(get = get_PinCode, put = set_PinCode))string PinCode;
    _declspec(property(get = get_AccountBalance, put = set_AccountBalance))float
AccountBalance;

    //client functions
    bool IsEmpty() { return (this->_Mode == enMode::EmptyMode); }

    void Print() {
        cout << "\nClient Card:";
        cout << "\n-----";
        cout << "\nFirstName :" << this->get_FirstName();
        cout << "\nLastName :" << this->get_LastName();
        cout << "\nFullName :" << this->get_FullName();
        cout << "\nEmail :" << this->get_Email();
        cout << "\nPhone :" << this->get_Phone();
        cout << "\nAcc. Number :" << this->_AccountNumber;
        cout << "\nPassword :" << this->_PinCode;
        cout << "\nBalance :" << this->_AccountBalance;
        cout << "\n-----";
    }

    static clsBankClient Find(string AccountNumber) {

        fstream MyFile;
        MyFile.open("Clients.txt",ios::in);

        if (MyFile.is_open()) {

            string Line;

            while (getline(MyFile,Line))
            {
                clsBankClient Client = _ConvertLineToClientObject(Line);
                if (Client._AccountNumber == AccountNumber) {
                    MyFile.close();
                    return Client;
                }
            }
        }
        return _GetEmptyClientObject();
    }
}

```

```

        }

    }

    static clsBankClient Find(string AccountNumber, string PinCode) {

        fstream MyFile;
        MyFile.open("Clients.txt", ios::in);

        if (MyFile.is_open()) {

            string Line;

            while (getline(MyFile, Line))
            {
                clsBankClient Client = _ConvertLineToClientObject(Line);
                if (Client._AccountNumber ==
                    AccountNumber && Client._PinCode == PinCode) {
                    MyFile.close();
                    return Client;
                }
            }
            return _GetEmptyClientObject();
        }
    }

    static bool IsClientExist(string AccountNumber) {

        clsBankClient Client = clsBankClient::Find(AccountNumber);
        return (!Client.IsEmpty());
    }
}

```

```

#include <iostream>
#include "clsBankClient.h"
using namespace std;
int main()

{
    clsBankClient Client1 = clsBankClient::Find("A101");
    Client1.Print();

    clsBankClient Client2 = clsBankClient::Find("A101","1234");
    Client2.Print();

    system("pause>0");

    return 0;
}

```

ده کود ال  
main

## Lesson #2 - Update Client

هنعمل update لبيانات العميل

```

#include <iostream>
#include "clsBankClient.h"
#include "clsInputValidate.h"

void ReadClientInfo(clsBankClient& Client) { ... }

void UpdateClient() { ... }

int main()
{
    UpdateClient();
    system("pause>0");
    return 0;
}

```

هنسخدم المكتبه بتاعت `clsInputValidate` عشان نشيك عالقيم اللي اليوزر هيدخلها

```

C:\Users\Acer\source\repo\ConsoleApplication1\x64\Debug\ConsoleApplicati

Account number is not found, choose another one: A38838
Account number is not found, choose another one: A101
Client Card:
FirstName : Mohammed
LastName : Abu-Hadoud
Full Name : Mohammed Abu-Hadoud
Email : Mo@gmail.com
Phone : 07999123
Acc. Number : A101
Password : 16000
Balance : 16000

Update Client Info:
Enter FirstName: Ahmad
Enter LastName: ALi
Enter Email: Ahmad@Gmail.com
Enter Phone: 093883
Enter PinCode: 123

```

```

enum enSaveResults { svFaildEmptyObject = 0, svSucceeded = 1 };

enSaveResults Save() { ... }

static bool IsClientExist(string AccountNumber) { ... }

```

عملنا enum عشان نعرف منها ان كانت عملية الحفظ تمت ولا لا  
ال function دي بتحفظ الداتا في ال file وبرجعلك ان كانت العملية تمت ولا لا  
عشان بعد كده في ال update يطلعك رساله ان كانت عملية الحفظ تمت ولا لا

```
clsBankClient::enSaveResults SaveResult;

SaveResult = Client1.Save();

switch (SaveResult)
{
case clsBankClient::enSaveResults::svSucceeded:
{
    cout << "\nAccount Updated Successfully :-)\n";
    Client1.Print();
    break;
}
case clsBankClient::enSaveResults::svFailedEmptyObject:
{
    cout << "\nError account was not saved because it's Empty";
    break;
}
```

ال save كل اللي بتعمله هو انها تشوف ال mode بتاع الكلاس لو كان empty هيرجعلك  
ولو كان update هيستدعي ال function اللي اسمها update ويرجعلك ان العملية تمت  
ال constructor ده كنا بنجيه من ال mode

```
enSaveResults Save()
{
    switch (_Mode)
    {
        case enMode::EmptyMode:
        {
            return enSaveResults::svFailedEmptyObject;
        }

        case enMode::UpdateMode:
        {
            _Update();

            return enSaveResults::svSucceeded;
        }
    }
}
```

انا دلوقتي المفروض اني دورت علي العميل بال find وبياناته كلها ظهرت عندي وبالتالي الكلاس  
اصبح فيه داتا

ال update function بتقرأ بيانات العملاء من الملف وبتدور على رقم الحساب و بتبدل البيانات القديمه بالجديده وتعيد كتابه الداتا في الملف

```

static vector <clsBankClient> _LoadClientsDataFromFile() { ... }

static void _SaveClientsDataToFile(vector <clsBankClient> vClients)

void _Update()
{
    vector <clsBankClient> _vClients;
    _vClients = _LoadClientsDataFromFile();

    for (clsBankClient& C : _vClients)
    {
        if (C.AccountNumber() == AccountNumber())
        {
            C = *this;
            break;
        }
    }

    _SaveClientsDataToFile(_vClients);
}

```

وبقولك افضل دايما الحاجات الخاصه بال ui عن الداتا نفسها

```

#pragma once
#include<iostream>
#include"clsPerson.h"
#include"clsString.h"
#include <vector>
#include <string>
#include <fstream>
using namespace std;

class clsBankClient:public clsPerson
{
private:
    enum enMode {EmptyMode=0,UpdateMode=1};

    //variables
    enMode _Mode;
    string _AccountNumber;
    string _PinCode;
    float _AccountBalance;

    //private client functions
    static clsBankClient _ConvertLineToClientObject(string Line,string Seperator="#//#") {
        vector<string> vClientData;

        vClientData=clsString::Split(Line,Seperator);

        return clsBankClient(enMode::UpdateMode, vClientData[0], vClientData[1]
                           , vClientData[2], vClientData[3], vClientData[4], vClientData[5],
                           stod(vClientData[6]));
    }
}

```

ده كود ال  
client

```

}

static string _ConvertClientObjecttoLine(clsBankClient Client,string Seperator="#/#") {
    return Client.get_FirstName()
        +Seperator+Client.get_LastName()
        +Seperator+Client.get_Email()
        +Seperator+Client.get_Phone()
        +Seperator+Client._AccountNumber
        +Seperator+Client._PinCode
        +Seperator+to_string(Client._AccountBalance);
}

static clsBankClient _GetEmptyClientObject() {
    return clsBankClient(enMode::EmptyMode,"","","","","","","",0);
}

static vector<clsBankClient> _LoadClientDataFromFile() {
    vector<clsBankClient> vClients;
    fstream MyFile;
    MyFile.open("Clients.txt",ios::in);

    if (MyFile.is_open()) {
        string Line;
        while (getline(MyFile,Line))
        {
            clsBankClient Client=_ConvertLineToClientObject(Line);
            vClients.push_back(Client);
        }
        MyFile.close();
    }
    return vClients;
}

static void _SaveClientsDataToFile(vector<clsBankClient> &vClients) {
    fstream MyFile;
    MyFile.open("Clients.txt",ios::out);

    string DataLine;

    if (MyFile.is_open()) {
        for (clsBankClient &C:vClients)
            DataLine = _ConvertClientObjecttoLine(C);
            MyFile << DataLine << endl;
    }
    MyFile.close();
}

void _Update() {
    vector<clsBankClient> _vClients;
    _vClients = _LoadClientDataFromFile();

    for (clsBankClient &C:_vClients) {
        if (C.get_AccountNumber()==get_AccountNumber()) {
            C = *this;
            break;
        }
    }
    _SaveClientsDataToFile(_vClients);
}

```

```

public :

    //constructors
    clsBankClient(enMode Mode,string FirstName, string LastName, string Email, string Phone
    , string AccountNumber,string PinCode,float AccountBalance)
    :clsPerson(FirstName,LastName,Email,Phone){

        this->_Mode = Mode;
        this->_AccountNumber = AccountNumber;
        this->_PinCode = PinCode;
        this->_AccountBalance = AccountBalance;
    }

    //setters
    void set_Mode(enMode Mode) { this->_Mode = Mode; }
    void set_AccountNumber(string AccountNumber) { this->_AccountNumber = AccountNumber;
}

    void set_PinCode(string PinCode) { this->_PinCode = PinCode; }
    void set_AccountBalance(float AccountBalance) { this->_AccountBalance = AccountBalance; }

    //getters
    enMode get_Mode() { return this->_Mode; }
    string get_AccountNumber() { return this->_AccountNumber; }
    string get_PinCode() { return this->_PinCode; }
    float get_AccountBalance() { return this->_AccountBalance; }

    //declspec
    _declspec(property(get = get_Mode, put = set_Mode))enMode Mode;
    _declspec(property(get = get_AccountNumber, put = set_AccountNumber))string
AccountNumber;
    _declspec(property(get = get_PinCode, put = set_PinCode))string PinCode;
    _declspec(property(get = get_AccountBalance, put = set_AccountBalance))float
AccountBalance;

    //public client functions
    bool IsEmpty() { return (this->_Mode == enMode::EmptyMode); }

    void Print() {
        cout << "\nClient Card:";
        cout << "\n-----";
        cout << "\nFirstName :"<<this->get_FirstName();
        cout << "\nLastName :"<<this->get_LastName();
        cout << "\nFullName :" << this->get_FullName();
        cout << "\nEmail :"<< this->get_Email();
        cout << "\nPhone :"<< this->get_Phone();
        cout << "\nAcc. Number :"<<this->_AccountNumber;
        cout << "\nPassword :"<<this->_PinCode;
        cout << "\nBalance :"<<this->_AccountBalance;
        cout << "\n-----";
    }

    static clsBankClient Find(string AccountNumber) {

        fstream MyFile;
        MyFile.open("Clients.txt",ios::in);

        if (MyFile.is_open()) {

            string Line;

```

```

        while (getline(MyFile,Line))
        {
            clsBankClient Client = _ConvertLineToClientObject(Line);
            if (Client._AccountNumber == AccountNumber) {
                MyFile.close();
                return Client;
            }
        }
        return _GetEmptyClientObject();
    }

static clsBankClient Find(string AccountNumber, string PinCode) {

    fstream MyFile;
    MyFile.open("Clients.txt", ios::in);

    if (MyFile.is_open()) {

        string Line;

        while (getline(MyFile, Line))
        {
            clsBankClient Client = _ConvertLineToClientObject(Line);
            if (Client._AccountNumber ==
AccountNumber&&Client._PinCode==PinCode) {
                MyFile.close();
                return Client;
            }
        }
        return _GetEmptyClientObject();
    }
}

enum enSaveResults { svFailedEmptyObject = 0, svSucceeded = 1 };

enSaveResults save() {
    switch (_Mode)
    {
    case clsBankClient::EmptyMode:
        return enSaveResults::svFailedEmptyObject;
        break;
    case clsBankClient::UpdateMode:
        _Update();
        return enSaveResults::svSucceeded;
        break;
    default:
        break;
    }
}

static bool IsClientExist(string AccountNumber) {

    clsBankClient Client = clsBankClient::Find(AccountNumber);
    return (!Client.IsEmpty());
}

```

د کود ال  
main

```
};

#include <iostream>
#include "clsBankClient.h"
#include "clsInputValidate.h"
using namespace std;

void ReadClientInfo(clsBankClient &Client) {
    cout << "\nEnter FirstName: ";
    Client.FirstName = clsInputValidate::ReadString();

    cout << "\nEnter LastName: ";
    Client.LastName = clsInputValidate::ReadString();

    cout << "\nEnter Email: ";
    Client.Email = clsInputValidate::ReadString();

    cout << "\nEnter Phone: ";
    Client.Phone = clsInputValidate::ReadString();

    cout << "\nEnter PinCode: ";
    Client.PinCode = clsInputValidate::ReadString();

    cout << "\nEnter AccountBalance: ";
    Client.AccountBalance = clsInputValidate::ReadFloatNumber();

}

void UpdateClient() {
    string AccountNumber = "";
    cout << "\nPlease Enter client Account Number: ";
    AccountNumber = clsInputValidate::ReadString();

    while (!clsBankClient::IsClientExist(AccountNumber)) {
        cout << "\nAccount number is not found, choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }

    clsBankClient Client1 = clsBankClient::Find(AccountNumber);
    Client1.Print();

    cout << "\n\nUpdate Client Info:";
    cout << "\n_____ \n";
    ReadClientInfo(Client1);

    clsBankClient::enSaveResults SaveResult;
    SaveResult = Client1.save();

    switch (SaveResult)
    {
        case clsBankClient::svFailedEmptyObject:
            cout << "\nError account was not saved because it's Empty";
            break;
        case clsBankClient::svSucceeded:
            cout << "\nAccount Updated Successfully :-) \n";
            Client1.Print();
            break;
        default:
            break;
    }
}
```

```

int main()
{
    UpdateClient();
    system("pause>0");
    return 0;
}

#pragma once
#include<iostream>
using namespace std;

class clsPerson
{
private:
    string _FirstName;
    string _LastName;
    string _Email;
    string _Phone;

public:
    //constructors
    clsPerson(string FirstName, string LastName, string Email, string Phone) {
        this->_FirstName = FirstName;
        this->_LastName = LastName;
        this->_Email = Email;
        this->_Phone = Phone;
    }

    //setters
    void set_FirstName (string FirstName) { this->_FirstName = FirstName; }
    void set_LastName (string LastName) { this->_LastName = LastName; }
    void set_Email(string Email) { this->_Email = Email; }
    void set_Phone(string Phone) { this->_Phone = Phone; }

    //getters

    string get_FirstName() { return this->_FirstName; }
    string get_LastName() { return this->_LastName; }
    string get_Email() { return this->_Email; }
    string get_Phone() { return this->_Phone; }
    string get_FullName() { return this->_FirstName + " " + this->_LastName; }

    //declspec
    __declspec(property(get = get_FirstName, put = set_FirstName)) string FirstName;
    __declspec(property(get = get_LastName, put = set_LastName)) string LastName;
    __declspec(property(get = get_Email, put = set_Email)) string Email;
    __declspec(property(get = get_Phone, put = set_Phone)) string Phone;

    //person functions
    void Print()
    {
        cout << "\nInfo:" ;
        cout << "\n_____";
        cout << "\nFirstName: " << this->_FirstName;
        cout << "\nLastName : " << this->_LastName;
        cout << "\nFull Name: " << this->get_FullName();
        cout << "\nEmail : " << this->_Email;
    }
}

```

ده کود آن  
person

```

        cout << "\nPhone : " << this->_Phone;
        cout << "\n_____ \n";
    }

};

//ProgrammingAdvces.com
//Mohammed Abu-Hadhoud

#pragma once
#include <iostream>
#include <string>
#include "clsString.h"
#include "clsDate.h"

class clsInputValidate
{

public:

    static bool IsNumberBetween(short Number, short From, short To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsNumberBetween(int Number, int From, int To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsNumberBetween(double Number, double From, double To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsDateBetween(clsDate Date, clsDate From, clsDate To)
    {
        //Date>=From && Date<=To
        if ((clsDate::IsDate1AfterDate2(Date, From) || clsDate::IsDate1EqualDate2(Date, From))
            &&
            (clsDate::IsDate1BeforeDate2(Date, To) || clsDate::IsDate1EqualDate2(Date,
To)))
        )
        {
            return true;
        }

        //Date>=To && Date<=From
        if ((clsDate::IsDate1AfterDate2(Date, To) || clsDate::IsDate1EqualDate2(Date, To))

```

د کود ال  
validate

```

        &&
        (clsDate::IsDate1BeforeDate2(Date, From) || clsDate::IsDate1EqualDate2(Date,
From))
    )
{
    return true;
}

return false;
}

static int ReadIntNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{
    int Number;
    while (!(cin >> Number)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << ErrorMessage;
    }
    return Number;
}

static int ReadIntNumberBetween(int From, int To, string ErrorMessage = "Number is not
within range, Enter again:\n")
{
    int Number = ReadIntNumber();

    while (!IsNumberBetween(Number, From, To))
    {
        cout << ErrorMessage;
        Number = ReadIntNumber();
    }
    return Number;
}

static double ReadFloatNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{
    float Number;
    while (!(cin >> Number)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << ErrorMessage;
    }
    return Number;
}

static double ReadFloatNumberBetween(double From, double To, string ErrorMessage =
"Number is not within range, Enter again:\n")
{
    float Number = ReadFloatNumber();

    while (!IsNumberBetween(Number, From, To)) {
        cout << ErrorMessage;
        Number = ReadDblNumber();
    }
    return Number;
}

static double ReadDblNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{

```

```

        double Number;
        while (!(cin >> Number)) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << ErrorMessage;
        }
        return Number;
    }

    static double ReadDblNumberBetween(double From, double To, string ErrorMessage =
"Number is not within range, Enter again:\n")
    {
        double Number = ReadDblNumber();

        while (!IsNumberBetween(Number, From, To)) {
            cout << ErrorMessage;
            Number = ReadDblNumber();
        }
        return Number;
    }

    static bool IsValidDate(clsDate Date)
    {
        return clsDate::IsValidDate(Date);
    }

    static string ReadString()
    {
        string S1 = "";
        // Usage of std::ws will extract allthe whitespace character
        getline(cin >> ws, S1);
        return S1;
    }
}

```

### Lesson #3 - Add New Client

```

void AddNewClient()
{
    string AccountNumber = "";

    cout << "\nPlease Enter Account Number: ";
    AccountNumber = clsInputValidate::ReadString();
    while (clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nAccount Number Is Already Used, Choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }

    clsBankClient NewClient = clsBankClient::GetAddNewClientObject(AccountNumber);

    ReadClientInfo(NewClient);

```

Copyright 2022

بিখليك تكتب رقم الحساب الأول لو موجود بيقولك انه موجود ولو مش موجود بيعمل object جديد عن طريق ال add new client object وببيديله رقم الحساب  
بعدين بيقرأ الداتا ويحفظها ويقولك لو عملية الحفظ تمت ولا لا

```

clsBankClient::enSaveResults SaveResult;

SaveResult = NewClient.Save();

switch (SaveResult)
{
case clsBankClient::enSaveResults::svSucceeded:
{
    cout << "\nAccount Added Successfully :-)\n";
    NewClient.Print();
    break;
}
case clsBankClient::enSaveResults::svFailedEmptyObject:
{
    cout << "\nError account was not saved because it's empty\n";
    break;
}
case clsBankClient::enSaveResults::svFailedAccountNumberExists:
{
}
}

```

## clsBankClient class

```

#pragma once
#include<iostream>
#include"clsPerson.h"
#include"clsString.h"
#include <vector>
#include <string>
#include <fstream>
using namespace std;

class clsBankClient:public clsPerson
{
private:
    enum enMode {EmptyMode=0,UpdateMode=1,AddNewMode=2};

    //variables
    enMode _Mode;
    string _AccountNumber;
    string _PinCode;
    float _AccountBalance;

    //private client functions
    static clsBankClient _ConvertLineToClientObject(string Line,string Seperator="#//#") {
        vector<string> vClientData;

        vClientData=clsString::Split(Line,Seperator);

        return clsBankClient(enMode::UpdateMode, vClientData[0], vClientData[1]
                           , vClientData[2], vClientData[3], vClientData[4], vClientData[5],
                           stod(vClientData[6]));
    }

    static string _ConvertClientObjecttoLine(clsBankClient Client,string Seperator="#//#")
    {
        return Client.get_FirstName()
            +Seperator+Client.get_LastName()
            +Seperator+Client.get_Email()
            +Seperator+Client.get_Phone()
            +Seperator+Client._AccountNumber
            +Seperator+Client._PinCode
            +Seperator+to_string(Client._AccountBalance);
    }
}

```

```

static clsBankClient _GetEmptyClientObject() {
    return clsBankClient(enMode::EmptyMode,"","","","","","","",0);
}

static vector<clsBankClient> _LoadClientDataFromFile() {
    vector<clsBankClient> vClients;
    fstream MyFile;
    MyFile.open("Clients.txt",ios::in);

    if (MyFile.is_open()) {
        string Line;
        while (getline(MyFile,Line))
        {
            clsBankClient Client=_ConvertLineToClientObject(Line);
            vClients.push_back(Client);
        }
        MyFile.close();
    }
    return vClients;
}

void _AddDataLineToFile(string stDataLine) {
    fstream MyFile;
    MyFile.open("Clients.txt", ios::out | ios::app);

    if (MyFile.is_open())
        MyFile << stDataLine << endl;
    MyFile.close();
}
}

static void _SaveClientsDataToFile(vector<clsBankClient> &vClients) {
    fstream MyFile;
    MyFile.open("Clients.txt",ios::out);

    string DataLine;

    if (MyFile.is_open())
        for (clsBankClient &C:vClients) {
            DataLine = _ConvertClientObjecttoLine(C);
            MyFile << DataLine << endl;
        }
    MyFile.close();
}

void _Update() {
    vector<clsBankClient> _vClients;
    _vClients = _LoadClientDataFromFile();

    for (clsBankClient &C:_vClients) {
        if (C.get_AccountNumber()==get_AccountNumber())
            C = *this;
        break;
    }
    _SaveClientsDataToFile(_vClients);
}

```

```

void _AddNew() {
    _AddDataLineToFile(_ConvertClientObjecttoLine(*this));
}

public :

//constructors
clsBankClient(enMode Mode,string FirstName, string LastName, string Email, string
Phone
, string AccountNumber,string PinCode,float AccountBalance)
:clsPerson(FirstName,LastName,Email,Phone){

    this->_Mode = Mode;
    this->_AccountNumber = AccountNumber;
    this->_PinCode = PinCode;
    this->_AccountBalance = AccountBalance;
}

//setters
void set_Mode(enMode Mode) { this->_Mode = Mode; }
void set_AccountNumber(string AccountNumber) { this->_AccountNumber =
AccountNumber; }
void set_PinCode(string PinCode) { this->_PinCode = PinCode; }
void set_AccountBalance(float AccountBalance) { this->_AccountBalance =
AccountBalance; }

//getters
enMode get_Mode() { return this->_Mode; }
string get_AccountNumber() { return this->_AccountNumber; }
string get_PinCode() { return this->_PinCode; }
float get_AccountBalance() { return this->_AccountBalance; }

//declspec
_declspec(property(get = get_Mode, put = set_Mode))enMode Mode;
_declspec(property(get = get_AccountNumber, put = set_AccountNumber))string
AccountNumber;
_declspec(property(get = get_PinCode, put = set_PinCode))string PinCode;
_declspec(property(get = get_AccountBalance, put = set_AccountBalance))float
AccountBalance;

//public client functions
bool IsEmpty() { return (this->_Mode == enMode::EmptyMode); }

void Print() {
    cout << "\nClient Card:";
    cout << "\n-----";
    cout << "\nFirstName :" << this->get_FirstName();
    cout << "\nLastName :" << this->get_LastName();
    cout << "\nFullName :" << this->get_FullName();
    cout << "\nEmail :" << this->get_Email();
    cout << "\nPhone :" << this->get_Phone();
    cout << "\nAcc. Number :" << this->_AccountNumber;
    cout << "\nPassword :" << this->_PinCode;
    cout << "\nBalance :" << this->_AccountBalance;
    cout << "\n-----";
}

static clsBankClient Find(string AccountNumber) {

```

```

fstream MyFile;
MyFile.open("Clients.txt",ios::in);

if (MyFile.is_open()) {

    string Line;

    while (getline(MyFile,Line))
    {
        clsBankClient Client = _ConvertLineToClientObject(Line);
        if (Client._AccountNumber == AccountNumber) {
            MyFile.close();
            return Client;
        }
    }
    return _GetEmptyClientObject();
}

static clsBankClient Find(string AccountNumber, string PinCode) {

    fstream MyFile;
    MyFile.open("Clients.txt", ios::in);

    if (MyFile.is_open()) {

        string Line;

        while (getline(MyFile, Line))
        {
            clsBankClient Client = _ConvertLineToClientObject(Line);
            if (Client._AccountNumber ==
AccountNumber&&Client._PinCode==PinCode) {
                MyFile.close();
                return Client;
            }
        }
        return _GetEmptyClientObject();
    }
}

enum enSaveResults { svFailedEmptyObject = 0, svSucceeded = 1,
svFaildAccountNumberExists=2};

enSaveResults save() {
    switch (_Mode)
    {
    case clsBankClient::EmptyMode:
        return enSaveResults::svFailedEmptyObject;
        break;
    case clsBankClient::UpdateMode:
        _Update();
        return enSaveResults::svSucceeded;
        break;
    case clsBankClient::AddNewMode:
        if (clsBankClient::IsClientExist(_AccountNumber)) {
            return enSaveResults::svFaildAccountNumberExists;
        }
    }
}

```

```

        else {
            _AddNew();
            _Mode = enMode::UpdateMode;
            return enSaveResults::svSucceeded;
        }
        break;

    default:
        break;
    }
}

static bool IsClientExist(string AccountNumber) {

    clsBankClient Client = clsBankClient::Find(AccountNumber);
    return (!Client.IsEmpty());
}

static clsBankClient GetAddNewClientObject(string AccountNumber)
{
    return clsBankClient(enMode::AddNewMode, "", "", "", "", AccountNumber,
"", 0);
}
};


```

Main

```

#include <iostream>
#include "clsBankClient.h"
#include "clsInputValidate.h"
using namespace std;

void ReadClientInfo(clsBankClient &Client) {
    cout << "\nEnter FirstName: ";
    Client.FirstName = clsInputValidate::ReadString();

    cout << "\nEnter LastName: ";
    Client.LastName = clsInputValidate::ReadString();

    cout << "\nEnter Email: ";
    Client.Email = clsInputValidate::ReadString();

    cout << "\nEnter Phone: ";
    Client.Phone = clsInputValidate::ReadString();

    cout << "\nEnter PinCode: ";
    Client.PinCode = clsInputValidate::ReadString();

    cout << "\nEnter AccountBalance: ";
    Client.AccountBalance = clsInputValidate::ReadFloatNumber();
}

void UpdateClient() {
    string AccountNumber = "";
    cout << "\nPlease Enter client Account Number: ";
    AccountNumber = clsInputValidate::ReadString();

    while (!clsBankClient::IsClientExist(AccountNumber)) {
        cout << "\nAccount number is not found, choose another one: ";

```

```

    AccountNumber = clsInputValidate::ReadString();
}

clsBankClient Client1 = clsBankClient::Find(AccountNumber);
Client1.Print();

cout << "\n\nUpdate Client Info:";
cout << "\n_____ \n";
ReadClientInfo(Client1);

clsBankClient::enSaveResults SaveResult;
SaveResult = Client1.save();

switch (SaveResult)
{
case clsBankClient::svFailedEmptyObject:
    cout << "\nError account was not saved because it's Empty";
    break;
case clsBankClient::svSucceeded:
    cout << "\nAccount Updated Successfully :-) \n";
    Client1.Print();
    break;
default:
    break;
}

void AddNewClient() {
    string AccountNumber = "";

    cout << "\nPlease Enter Account Number: ";
    AccountNumber = clsInputValidate::ReadString();

    while (clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nAccount Number Is Already used, Choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }
    clsBankClient NewClient = clsBankClient::GetAddNewClientObject(AccountNumber);

    ReadClientInfo(NewClient);

    clsBankClient::enSaveResults SaveResult;

    SaveResult = NewClient.save();

    switch (SaveResult)
    {
case clsBankClient::svFailedEmptyObject:
    cout << "\nError account was not saved because it's Empty";
    break;
case clsBankClient::svSucceeded:
    cout << "\nAccount Added Successfully :-) \n";
    NewClient.Print();
    break;
case clsBankClient::svFaildAccountNumberExists:
    cout << "\nError account was not saved because account number is used! \n";
    break;
default:
    break;
}
}

```

```

    }

int main()
{
    AddNewClient();

    system("pause>0");

    return 0;
}

```

## Lesson #4 - Delete Client

```

C:\Users\Acer\source\repos\ConsoleApplication1\x64\Debug\ConsoleAp
Please Enter Account Number: A101

Client Card:

FirstName : Mohammed
LastName : Abu-Hadhoud
Full Name : Mohammed Abu-Hadhoud
Email : Mo@gmail.com
Phone : 799997886
Acc. Number : A101
Password : 1234
Balance : 5000

Are you sure you want to delete this client y/n? y
Client Deleted Successfully :-)

Client Card:

FirstName :
LastName :
Full Name :
Email :
Phone :
Acc. Number :
Password :
Balance : 0

```

لما بتعمل **delete** بتفضي ال **object** وبيمسحه

```

#pragma once
#include <iostream>
#include "clsPerson.h"
#include "clsString.h"
#include <vector>
#include <string>
#include <fstream>
using namespace std;

class clsBankClient:public clsPerson
{

```

ده ال **client** هنعلم عالجديد  
بس

```

private:
    enum enMode {EmptyMode=0,UpdateMode=1,AddNewMode=2};

    //variables
    enMode _Mode;
    string _AccountNumber;
    string _PinCode;
    float _AccountBalance;
    bool _MarkForDelete=false;

    //private client functions
    static clsBankClient _ConvertLineToClientObject(string Line,string
Seperator="#/#") {
        vector<string> vClientData;

        vClientData=clsString::Split(Line,Seperator);

        return clsBankClient(enMode::UpdateMode, vClientData[0],
vClientData[1]
                , vClientData[2], vClientData[3], vClientData[4],
vClientData[5], stod(vClientData[6]));
    }

    static string _ConvertClientObjecttoLine(clsBankClient Client,string
Seperator="#/#") {
        return Client.get_FirstName()
            +Seperator+Client.get_LastName()
            +Seperator+Client.get_Email()
            +Seperator+Client.get_Phone()
            +Seperator+Client._AccountNumber
            +Seperator+Client._PinCode
            +Seperator+to_string(Client._AccountBalance);
    }

    static clsBankClient _GetEmptyClientObject() {
        return clsBankClient(enMode::EmptyMode,"","","","","","",0);
    }

    static vector<clsBankClient>_LoadClientDataFromFile() {
        vector<clsBankClient> vClients;
        fstream MyFile;
        MyFile.open("Clients.txt",ios::in);

        if (MyFile.is_open()) {
            string Line;
            while (getline(MyFile,Line))
            {
                clsBankClient
Client=_ConvertLineToClientObject(Line);
                vClients.push_back(Client);
            }
            MyFile.close();
        }
        return vClients;
    }

    void _AddDataLineToFile(string stDataLine) {
        fstream MyFile;
        MyFile.open("Clients.txt", ios::out | ios::app);

```

زودنا متغير bool اسمه  
mark for delete

```

        if (MyFile.is_open()) {
            MyFile << stDataLine << endl;
            MyFile.close();
        }
    }

static void _SaveClientsDataToFile(vector<clsBankClient> &vClients) {
    fstream MyFile;
    MyFile.open("Clients.txt",ios::out);

    string DataLine;

    if (MyFile.is_open()) {
        for (clsBankClient &C:vClients) {
            if (C.MarkedForDelete() == false) {
                DataLine =
_ConvertClientObjecttoLine(C);
                MyFile << DataLine << endl;
            }
        }
        MyFile.close();
    }

    void _Update() {
        vector<clsBankClient> _vClients;
        _vClients = _LoadClientDataFromFile();

        for (clsBankClient &C:_vClients) {
            if (C.get_AccountNumber()==get_AccountNumber()) {
                C = *this;
                break;
            }
        }
        _SaveClientsDataToFile(_vClients);
    }

    void _AddNew() {
        _AddDataLineToFile(_ConvertClientObjecttoLine(*this));
    }

public :

//constructors
clsBankClient(enMode Mode,string FirstName, string LastName, string
Email, string Phone
, string AccountNumber,string PinCode,float AccountBalance)
:clsPerson(FirstName,LastName,Email,Phone){

    this->_Mode = Mode;
    this->_AccountNumber = AccountNumber;
    this->_PinCode = PinCode;
    this->_AccountBalance = AccountBalance;
}

//setters
void set_Mode(enMode Mode) { this->_Mode = Mode; }
void set_AccountNumber(string AccountNumber) { this-

```

هنا زودنا وقلنا له انه لو ال  
mark for delete مش  
معموله true احفظ الداتا في  
الملف

```

>_AccountNumber = AccountNumber; }
    void set_PinCode(string PinCode) { this->_PinCode = PinCode; }
    void set_AccountBalance(float AccountBalance) { this-
>_AccountBalance = AccountBalance; }

    //getters
    enMode get_Mode() { return this->_Mode; }
    string get_AccountNumber() { return this->_AccountNumber; }
    string get_PinCode() { return this->_PinCode; }
    float get_AccountBalance() { return this->_AccountBalance; }

    //declspec
    __declspec(property(get = get_Mode, put = set_Mode))enMode Mode;
    __declspec(property(get = get_AccountNumber, put =
set_AccountNumber))string AccountNumber;
    __declspec(property(get = get_PinCode, put = set_PinCode))string
PinCode;
    __declspec(property(get = get_AccountBalance, put =
set_AccountBalance))float AccountBalance;

    //public client functions
    bool IsEmpty() { return (this->_Mode == enMode::EmptyMode); }

    void Print() {
        cout << "\nClient Card:";
        cout << "\n-----";
        cout << "\nFirstName :" << this->get_FirstName();
        cout << "\nLastName :" << this->get_LastName();
        cout << "\nFullName :" << this->get_FullName();
        cout << "\nEmail :" << this->get_Email();
        cout << "\nPhone :" << this->get_Phone();
        cout << "\nAcc. Number :" << this->_AccountNumber;
        cout << "\nPassword :" << this->_PinCode;
        cout << "\nBalance :" << this->_AccountBalance;
        cout << "\n-----";
    }

    static clsBankClient Find(string AccountNumber) {

        fstream MyFile;
        MyFile.open("Clients.txt",ios::in);

        if (MyFile.is_open()) {

            string Line;

            while (getline(MyFile,Line))
            {
                clsBankClient Client =
_ConvertLineToClientObject(Line);
                if (Client._AccountNumber ==
AccountNumber) {
                    MyFile.close();
                    return Client;
                }
            }
            return _GetEmptyClientObject();
        }
    }
}

```

```

static clsBankClient Find(string AccountNumber, string PinCode) {

    fstream MyFile;
    MyFile.open("Clients.txt", ios::in);

    if (MyFile.is_open()) {

        string Line;

        while (getline(MyFile, Line))
        {
            clsBankClient Client =
_ConvertLineToClientObject(Line);
            if (Client._AccountNumber ==
AccountNumber&&Client._PinCode==PinCode) {
                MyFile.close();
                return Client;
            }

        }
        return _GetEmptyClientObject();
    }
}

enum enSaveResults { svFailedEmptyObject = 0, svSucceeded = 1,
svFaildAccountNumberExists=2};

enSaveResults save() {
    switch (_Mode)
    {
        case clsBankClient::EmptyMode:
            return enSaveResults::svFailedEmptyObject;
            break;
        case clsBankClient::UpdateMode:
            _Update();
            return enSaveResults::svSucceeded;
            break;
        case clsBankClient::AddNewMode:
            if (clsBankClient::IsClientExist(_AccountNumber)) {
                return
enSaveResults::svFaildAccountNumberExists;
            }
            else {
                _AddNew();
                _Mode = enMode::UpdateMode;
                return enSaveResults::svSucceeded;
            }
            break;

        default:
            break;
    }
}

static bool IsClientExist(string AccountNumber) {

    clsBankClient Client = clsBankClient::Find(AccountNumber);
    return (!Client.IsEmpty());
}

```

و هنا عملنا اسمها function

بنحمل فيه الدتا من **delete**  
 الملف وندور عالعميل اللي  
 رقم حسابه نفس رقم الحساب  
 اللي معانا ولو لقاه بيعمل ال  
**mark for delete**  
 بيساوي **true**

وبعدن بيحفظ البيانات تاني  
 في الملف وبيخلي ال **object**  
 اللي احنا واقفين عليه فاضي

```

static clsBankClient GetAddNewClientObject(string AccountNumber)
{
    return clsBankClient(enMode::AddNewMode, "", "", "", "", 
AccountNumber, "", 0);
}

bool MarkedForDelete() {
    return _MarkForDelete;
}

bool Delete() {
    vector<clsBankClient>_vClients;
    _vClients = _LoadClientDataFromFile();
    for (clsBankClient &C:_vClients)
    {
        if (C.get_AccountNumber() == _AccountNumber) {
            C._MarkForDelete = true;
            break;
        }
    }

    _SaveClientsDataToFile(_vClients);
    *this=_GetEmptyClientObject();
    return true;
}
};
```

ده ال **main**

```

#include <iostream>
#include "clsBankClient.h"
#include "clsInputValidate.h"
using namespace std;

void ReadClientInfo(clsBankClient &Client) {
    cout << "\nEnter FirstName: ";
    Client.FirstName = clsInputValidate::ReadString();

    cout << "\nEnter LastName: ";
    Client.LastName = clsInputValidate::ReadString();

    cout << "\nEnter Email: ";
    Client.Email = clsInputValidate::ReadString();

    cout << "\nEnter Phone: ";
    Client.Phone = clsInputValidate::ReadString();

    cout << "\nEnter PinCode: ";
    Client.PinCode = clsInputValidate::ReadString();

    cout << "\nEnter AccountBalance: ";
    Client.AccountBalance = clsInputValidate::ReadFloatNumber();

}

void UpdateClient() {
    string AccountNumber = "";
    cout << "\nPlease Enter client Account Number: ";
    AccountNumber = clsInputValidate::ReadString();
```

```

while (!clsBankClient::IsClientExist(AccountNumber)) {
    cout << "\nAccount number is not found, choose another one: ";
    AccountNumber = clsInputValidate::ReadString();
}

clsBankClient Client1 = clsBankClient::Find(AccountNumber);
Client1.Print();

cout << "\n\nUpdate Client Info:";
cout << "\n_____ \n";
ReadClientInfo(Client1);

clsBankClient::enSaveResults SaveResult;
SaveResult = Client1.save();

switch (SaveResult)
{
case clsBankClient::svFailedEmptyObject:
    cout << "\nError account was not saved because it's Empty";
    break;
case clsBankClient::svSucceeded:
    cout << "\nAccount Updated Successfully :-) \n";
    Client1.Print();
    break;
default:
    break;
}
}

void AddNewClient() {
    string AccountNumber = "";

    cout << "\nPlease Enter Account Number: ";
    AccountNumber = clsInputValidate::ReadString();

    while (clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nAccount Number Is Already used, Choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }
    clsBankClient NewClient =
    clsBankClient::GetAddNewClientObject(AccountNumber);

    ReadClientInfo(NewClient);

    clsBankClient::enSaveResults SaveResult;

    SaveResult = NewClient.save();

    switch (SaveResult)
    {
case clsBankClient::svFailedEmptyObject:
    cout << "\nError account was not saved because it's Empty";
    break;
case clsBankClient::svSucceeded:
    cout << "\nAccount Added Successfully :-) \n";
    NewClient.Print();
    break;
case clsBankClient::svFaildAccountNumberExists:
    cout << "\nError account was not saved because account number is
}

```

```

used!\n";
    break;
default:
    break;
}

void DeleteClient() {
    string AccountNumber = "";
    cout << "\nPlease enter account number: ";
    AccountNumber = clsInputValidate::ReadString();

    while (!clsBankClient::IsClientExist(AccountNumber)) {
        cout << "\nAccount number is not found, please choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }
    clsBankClient Client1 = clsBankClient::Find(AccountNumber);
    Client1.Print();

    cout << "\nAre you sure you want to delete this client y/n? ";
    char Answer = 'h';
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y') {
        if (Client1.Delete()) {
            cout << "\nClient Deleted Successfully :-)\n";
            Client1.Print();
        }
        else {
            cout << "\nError Client Was not Deleted\n";
        }
    }
}

int main()
{
    DeleteClient();

    system("pause>0");

    return 0;
}

```

هنا عملنا **delete client**  
وعرفنا فيها متغير هنخزن فيه  
ال **account number**  
وبعدين بيخللي اليوزر يدخل  
رقم الحساب ولولقاه

بيدور عليه في الملف وبيطبعه

وبيساله عاوز تحذف ده؟  
لو قاله اه  
بيشغل ال **function** اللي  
اسمها **delete**

## Lesson #5 - List Clients

عاوزين نعمل **show client list** ودي بتعرضلنا بيانات العملاء كلهم

## Client List (12) Client(s).

Account Number	Client Name	Phone	Email	Pin Code	Balance
A103	Adli Haddad	8983883	Adli@gmail.com	1234	555
A104	Jamil Adli	23123123	Jamil@gmail.com	1234	8912
A07	Lina Loay	1234	Lina@gmail.com	125	123
A106	Mazem Kareem	898234	Mazin@gmail.com	1234	8002
A107	Lama Karmi	8983883	Lama@gmail.com	1234	1234
A108	Gandi Omran	1234	Ghandi@gmail.com	1234	7000
A109	Hamed Abbas	8838	H@gmail.com	1234	5000
A110	Hilal Mazin	123443	A@A.com	1234	9000
A111	Majida Yousef	1234	Majida	1234	717
A101	Mohammed Abu-Hadhoud	199192	msaquer@gmail.com	1234	901
A112	Ali Ali	83837	Ali@al.com	1234	5000
A500	Hamed Omar	37732	Om@g.com	1234	1221

عاوزين لو مفيش داتا نكتب جملة معينه بدل مانسيبيها فاضيه كده

<pre> static vector&lt;clsBankClient&gt; GetClientsList() {     return _LoadClientDataFromFile(); }  void PrintClientRecordLine(clsBankClient Client) {     cout &lt;&lt; " " &lt;&lt; setw(15) &lt;&lt; left &lt;&lt; Client.get_AccountNumber();     cout &lt;&lt; " " &lt;&lt; setw(20) &lt;&lt; left &lt;&lt; Client.get_FullName();     cout &lt;&lt; " " &lt;&lt; setw(12) &lt;&lt; left &lt;&lt; Client.Phone;     cout &lt;&lt; " " &lt;&lt; setw(20) &lt;&lt; left &lt;&lt; Client.Email;     cout &lt;&lt; " " &lt;&lt; setw(10) &lt;&lt; left &lt;&lt; Client.PinCode;     cout &lt;&lt; " " &lt;&lt; setw(12) &lt;&lt; left &lt;&lt; Client.AccountBalance;  }  void ShowClientsList() {     vector &lt;clsBankClient&gt; vClients = clsBankClient::GetClientsList();      cout &lt;&lt; "\n\t\t\tClient List (" &lt;&lt; vClients.size() &lt;&lt; ") Client(s).";     cout &lt;&lt; "\n_____";     cout &lt;&lt; " _____ \n" &lt;&lt; endl;      cout &lt;&lt; " " &lt;&lt; left &lt;&lt; setw(15) &lt;&lt; "Accout Number";     cout &lt;&lt; " " &lt;&lt; left &lt;&lt; setw(20) &lt;&lt; "Client Name";     cout &lt;&lt; " " &lt;&lt; left &lt;&lt; setw(12) &lt;&lt; "Phone";     cout &lt;&lt; " " &lt;&lt; left &lt;&lt; setw(20) &lt;&lt; "Email";     cout &lt;&lt; " " &lt;&lt; left &lt;&lt; setw(10) &lt;&lt; "Pin Code";     cout &lt;&lt; " " &lt;&lt; left &lt;&lt; setw(12) &lt;&lt; "Balance";     cout &lt;&lt; " \n_____ ";     cout &lt;&lt; " _____ \n" &lt;&lt; endl;      if (vClients.size() == 0)         cout &lt;&lt; "\t\t\tNo Clients Available In the System!";     else          for (clsBankClient Client : vClients)     { </pre>	<p>ده كلاس ال clsBankClient ده كل اللي زودناه</p> <p>ده ال main ده كل اللي زودناه جمل طباعة عاديه ومتنساش تصيف المكتبه iomanip</p>
---	--

```

PrintClientRecordLine(Client);
cout << endl;
}

cout << "\n_____";
cout << "_____ \n" << endl;
}

```

## Lesson #6 - Total Balances

عاوزين نطبع جدول فيه اسم العميل ورقم الحساب والرصيد بتابعه  
وفي الآخر نحسب مجموع الارصدة بالحروف والأرقام

Balances List (12) Client(s).		
Accout Number	Client Name	Balance
A103	Adli Haddad	555
A104	Jamil Adli	8912
A07	Lina Loay	123
A106	Mazem Kareem	8002
A107	Lama Karmi	1234
A108	Gandi Omran	7000
A109	Hamed Abbas	5000
A110	Hilal Mazin	9000
A111	Majida Yousef	717
A101	Mohammed Abu-Hadhoud	901
A112	Ali Ali	5000
A500	Hamed Omar	1221

Total Balances = 47665 ( Forty Seven Thousands Six Hundreds Sixty Five )
---

```

static string NumberToText(int Number) {
    if (Number == 0) {
        return "";
    }

    if (Number >= 1 && Number <= 19) {
        string arr[] = { "", "One", "Two", "Three", "Four", "Five", "Six", "Seven",
        "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen",
        "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen" };

        return arr[Number] + " ";
    }

    if (Number >= 20 && Number <= 99) {
        string arr[] = {
            "", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" };
        return arr[Number / 10] + " " + NumberToText(Number % 10);
    }

    if (Number >= 100 && Number <= 199)
    {

```

جيـنا في كـلاس ال util وـزودـنا ال function دـي

```

        return "One Hundred " + NumberToText(Number % 100);
    }

    if (Number >= 200 && Number <= 999)
    {
        return NumberToText(Number / 100) + "Hundreds " +
NumberToText(Number % 100);
    }

    if (Number >= 1000 && Number <= 1999)
    {
        return "One Thousand " + NumberToText(Number % 1000);
    }

    if (Number >= 2000 && Number <= 999999)
    {
        return NumberToText(Number / 1000) + "Thousands " +
NumberToText(Number % 1000);
    }

    if (Number >= 1000000 && Number <= 1999999)
    {
        return "One Million " + NumberToText(Number % 1000000);
    }

    if (Number >= 2000000 && Number <= 999999999)
    {
        return NumberToText(Number / 1000000) + "Millions " +
NumberToText(Number % 1000000);
    }

    if (Number >= 1000000000 && Number <= 1999999999)
    {
        return "One Billion " + NumberToText(Number % 1000000000);
    }
    else
    {
        return NumberToText(Number / 1000000000) + "Billions " +
NumberToText(Number % 1000000000);
    }

}

```

```

static double GetTotalBalances()
{
    vector<clsBankClient>vClients = GetClientsList();
    double TotalBalances = 0;

    for (clsBankClient &C:vClients)
        TotalBalances += C.AccountBalance;
    }

    return TotalBalances;
}

```

```

#include <iostream>
#include "clsBankClient.h"
#include "clsInputValidate.h"
#include <iomanip>
#include "clsUtil.h"

```

في كلاس ال client زودنا  
ال function دي  
يدوب بيلف عالعملاء يجمع  
الارصده ويرجعها لك

وده الكود اللي في ال main  
 مجرد جمل طباعه

```

void PrintClientRecordBalanceLine(clsBankClient Client)
{
    cout << " " << setw(15) << left << Client.get_AccountNumber();
    cout << " " << setw(40) << left << Client.get_FullName();
    cout << " " << setw(12) << left << Client.AccountBalance;
}

void ShowTotalBalances()
{
    vector <clsBankClient> vClients = clsBankClient::GetClientsList();

    cout << "\n\t\t\t\t\t\tBalances List (" << vClients.size() << ") Client(s).";
    cout << "\n_____";
    cout << " _____\n" << endl;

    cout << " " << left << setw(15) << "Accout Number";
    cout << " " << left << setw(40) << "Client Name";
    cout << " " << left << setw(12) << "Balance";
    cout << "\n_____";
    cout << " _____\n" << endl;

    double TotalBalances = clsBankClient::GetTotalBalances();

    if (vClients.size() == 0)
        cout << "\t\t\t\tNo Clients Available In the System!";
    else

        for (clsBankClient Client : vClients)
    {
        PrintClientRecordBalanceLine(Client);
        cout << endl;
    }

    cout << "\n_____";
    cout << " _____\n" << endl;
    cout << "\t\t\t\t\t\t Total Balances = " << TotalBalances << endl;
    cout << "\t\t\t\t\t\t (" << clsUtil::NumberToText(TotalBalances) << ")";
}
}

int main()
{
    ShowTotalBalances();
    system("pause>0");
    return 0;
}

```

## Lesson 7 - Main Screen

من هنا هنبدأ نشتغل عال ui اللي هو ال user interface او واجهة المستخدم



## ده کود کلاس ال input validate

```
//ProgrammingAdvices.com
//Mohammed Abu-Hadhoud

#pragma once
#include <iostream>
#include <string>
#include "clsString.h"
#include "clsDate.h"

class clsInputValidate
{
public:
    static bool IsNumberBetween(int Number, int From, int To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsNumberBetween(double Number, double From, double To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }

    static bool IsDateBetween(clsDate Date, clsDate From, clsDate To)
    {
        //Date>=From && Date<=To
        if ((clsDate::IsDate1AfterDate2(Date, From) || clsDate::IsDate1EqualDate2(Date, From))
            &&
            (clsDate::IsDate1BeforeDate2(Date, To) || clsDate::IsDate1EqualDate2(Date, To)))
        {
            return true;
        }

        //Date>=To && Date<=From
        if ((clsDate::IsDate1AfterDate2(Date, To) || clsDate::IsDate1EqualDate2(Date, To))
            &&
            (clsDate::IsDate1BeforeDate2(Date, From) || clsDate::IsDate1EqualDate2(Date, From)))
        {
            return true;
        }

        return false;
    }

    static short ReadShortNumber(string ErrorMessage = "Invalid Number, Enter again\n")
    {

```

```

short Number;
while (!(cin >> Number)) {
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cout << ErrorMessage;
}
return Number;
}

static int ReadIntNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{
    int Number;
    while (!(cin >> Number)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << ErrorMessage;
    }
    return Number;
}

static short ReadShortNumberBetween(short From, short To, string ErrorMessage = "Number is not within range,
Enter again:\n")
{
    int Number = ReadShortNumber();

    while (!IsNumberBetween(Number, From, To))
    {
        cout << ErrorMessage;
        Number = ReadShortNumber();
    }
    return Number;
}

static int ReadIntNumberBetween(int From, int To, string ErrorMessage = "Number is not within range, Enter
again:\n")
{
    int Number = ReadIntNumber();

    while (!IsNumberBetween(Number, From, To))
    {
        cout << ErrorMessage;
        Number = ReadIntNumber();
    }
    return Number;
}

static double ReadFloatNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{
    float Number;
    while (!(cin >> Number)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << ErrorMessage;
    }
    return Number;
}

static double ReadFloatNumberBetween(double From, double To, string ErrorMessage = "Number is not within
range, Enter again:\n")
{

```

```

float Number = ReadFloatNumber();

while (!IsNumberBetween(Number, From, To)) {
    cout << ErrorMessage;
    Number = ReadDblNumber();
}
return Number;
}

static double ReadDblNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{
    double Number;
    while (!(cin >> Number)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << ErrorMessage;
    }
    return Number;
}

static double ReadDblNumberBetween(double From, double To, string ErrorMessage = "Number is not within
range, Enter again:\n")
{
    double Number = ReadDblNumber();

    while (!IsNumberBetween(Number, From, To)) {
        cout << ErrorMessage;
        Number = ReadDblNumber();
    }
    return Number;
}

static bool IsValideDate(clsDate Date)
{
    return clsDate::IsValidDate(Date);
}

static string ReadString()
{
    string S1 = "";
    // Usage of std::ws will extract allthe whitespace character
    getline(cin >> ws, S1);
    return S1;
}

};


```

بعد كده هنجي لكود الشاشه الرئيسيه والكود هو نفسه اللي كنا عاملينه قبل كده  
بس هنجي عند كل function خاصه بشاشه معينه ونهنط الكود ده

```

static void _ShowAllClientsScreen()
{
    cout << "\nClient List Screen Will be here...\n";
}

```

```
#pragma once
#include <iostream>
```

```
#include "clsScreen.h"
#include "clsInputValidate.h"
#include <iomanip>

using namespace std;

class clsMainScreen :protected clsScreen
{

private:
    enum enMainMenuOptions {
        eListClients = 1, eAddNewClient = 2, eDeleteClient = 3,
        eUpdateClient = 4, eFindClient = 5, eShowTransactionsMenue = 6,
        eManageUsers = 7, eExit = 8
    };

    static short _ReadMainMenueOption()
    {
        cout << setw(37) << left << "" << "Choose what do you want to do? [1 to 8]? ";
        short Choice = clsInputValidate::ReadShortNumberBetween(1, 8, "Enter Number between 1 to 8? ");
        return Choice;
    }

    static void _GoBackToMainMenue()
    {
        cout << setw(37) << left << "" << "\n\tPress any key to go back to Main Menue...\n";

        system("pause>0");
        ShowMainMenue();
    }

    static void _ShowAllClientsScreen()
    {
        cout << "\nClient List Screen Will be here...\n";
    }

    static void _ShowAddNewClientsScreen()
    {
        cout << "\nAdd New Client Screen Will be here...\n";
    }

    static void _ShowDeleteClientScreen()
    {
        cout << "\nDelete Client Screen Will be here...\n";
    }

    static void _ShowUpdateClientScreen()
    {
        cout << "\nUpdate Client Screen Will be here...\n";
    }

    static void _ShowFindClientScreen()
    {
```

```

cout << "\nFind Client Screen Will be here...\n";
}

static void _ShowTransactionsMenue()
{
    cout << "\nTransactions Menue Will be here...\n";
}

static void _ShowManageUsersMenue()
{
    cout << "\nUsers Menue Will be here...\n";
}

static void _ShowEndScreen()
{
    cout << "\nEnd Screen Will be here...\n";
}

static void _PerfromMainMenueOption(enMainMenueOptions MainMenueOption)
{
    switch (MainMenueOption)
    {
        case enMainMenueOptions::eListClients:
        {
            system("cls");
            _ShowAllClientsScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eAddNewClient:
        {
            system("cls");
            _ShowAddNewClientsScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eDeleteClient:
        {
            system("cls");
            _ShowDeleteClientScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eUpdateClient:
        {
            system("cls");
            _ShowUpdateClientScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eFindClient:
        {
            system("cls");
            _ShowFindClientScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eShowTransactionsMenue:
        {
            system("cls");
            _ShowTransactionsMenue();
        }
    }
}

```

```
_GoBackToMainMenue();
break;

case enMainMenuOptions::eManageUsers:
system("cls");
>ShowManageUsersMenue();
_GoBackToMainMenue();
break;

case enMainMenuOptions::eExit:
system("cls");
>ShowEndScreen();
//Login();

break;
}

}

public:

static void ShowMainMenue()
{

system("cls");
_DrawScreenHeader("\t\tMain Screen","");

cout << setw(37) << left << "" << "=====Main Menue===== \n";
cout << setw(37) << left << "" << "\t\tMain Menue\n";
cout << setw(37) << left << "" << "=====Main Menue===== \n";
cout << setw(37) << left << "" << "\t[1] Show Client List.\n";
cout << setw(37) << left << "" << "\t[2] Add New Client.\n";
cout << setw(37) << left << "" << "\t[3] Delete Client.\n";
cout << setw(37) << left << "" << "\t[4] Update Client Info.\n";
cout << setw(37) << left << "" << "\t[5] Find Client.\n";
cout << setw(37) << left << "" << "\t[6] Transactions.\n";
cout << setw(37) << left << "" << "\t[7] Manage Users.\n";
cout << setw(37) << left << "" << "\t[8] Logout.\n";
cout << setw(37) << left << "" << "===== \n";

_PerfromMainMenuOption((enMainMenuOptions)_ReadMainMenuOption());
}

};

};


```

وفي ال main هنشغل الشاشه

```
#include <iostream>
#include "clsMainScreen.h"

int main()

{

clsMainScreen::ShowMainMenue();

system("pause>0");
return 0;
}
```

```
}
```

## Lesson 8 - List Clients Screen

هنا هنعمل الشاشه بتاعت show client list

هنعمل كلاس خاص بالشاشه دي وبعدين نروح نشغلها من الشاشه الرئيسيه

ده الكود بتاع الشاشه

```
#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsBankClient.h"
#include <iomanip>

class clsClientListScreen :protected clsScreen
{
private:
    static void PrintClientRecordLine(clsBankClient Client)
    {
        cout << setw(8) << left << "" << "|" << setw(15) << left << Client.get_AccountNumber();
        cout << "|" << setw(20) << left << Client.get_FullName();
        cout << "|" << setw(12) << left << Client.Phone;
        cout << "|" << setw(20) << left << Client.Email;
        cout << "|" << setw(10) << left << Client.PinCode;
        cout << "|" << setw(12) << left << Client.AccountBalance;
    }

public:
    static void ShowClientsList()
    {
        vector <clsBankClient> vClients = clsBankClient::GetClientsList();
        string Title = "\t Client List Screen";
        string SubTitle = "\t (" + to_string(vClients.size()) + ") Client(s).";
        _DrawScreenHeader(Title, SubTitle);

        cout << setw(8) << left << "" << "\n\t_____"; 
        cout << "_____ \n" << endl;

        cout << setw(8) << left << "" << "|" << left << setw(15) << "Accout Number";
        cout << "|" << left << setw(20) << "Client Name";
        cout << "|" << left << setw(12) << "Phone";
        cout << "|" << left << setw(20) << "Email";
        cout << "|" << left << setw(10) << "Pin Code";
        cout << "|" << left << setw(12) << "Balance";
        cout << setw(8) << left << "" << "\n\t_____"; 
        cout << "_____ \n" << endl;

        if (vClients.size() == 0)
            cout << "\t\l\l\lNo Clients Available In the System!";
        else
    }
}
```

```

for (clsBankClient Client : vClients)
{
    PrintClientRecordLine(Client);
    cout << endl;
}

cout << setw(8) << left << "" << "\n\t_____";
cout << "_____ \n" << endl;

}
};

```

وده التعديل اللي عملناه في الشاشه الرئيسيه

```

static void _ShowAllClientsScreen()
{
    // cout << "\nClient List Screen Will be here...\n";
    clsClientListScreen::ShowClientsList();

}

```

## Lesson 9 - Add New Client Screen

هنعمل الشاشه بتاعت اضافه عميل جديد  
 ال function بتاعت الطباعه اللي ي كلاس ال client شيلناها وحطيناها هنا عشان نفل الاكواود  
 الخاصه بال ui عن الاكواود الخاصه بالبيانات

```

#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"
#include <iomanip>

class clsAddNewClientScreen : protected clsScreen
{
private:
    static void _ReadClientInfo(clsBankClient& Client)
    {
        cout << "\nEnter FirstName: ";
        Client.FirstName = clsInputValidate::ReadString();

        cout << "\nEnter LastName: ";
        Client.LastName = clsInputValidate::ReadString();

        cout << "\nEnter Email: ";
        Client.Email = clsInputValidate::ReadString();

        cout << "\nEnter Phone: ";
        Client.Phone = clsInputValidate::ReadString();

        cout << "\nEnter PinCode: ";
        Client.PinCode = clsInputValidate::ReadString();

        cout << "\nEnter Account Balance: ";
        Client.AccountBalance = clsInputValidate::ReadFloatNumber();
    }

    static void _PrintClient(clsBankClient Client)

```

```

{
    cout << "\nClient Card:";  

    cout << "\n_____";  

    cout << "\nFirstName : " << Client.FirstName;  

    cout << "\nLastName : " << Client.LastName;  

    cout << "\nFull Name : " << Client.get_FullName();  

    cout << "\nEmail : " << Client.Email;  

    cout << "\nPhone : " << Client.Phone;  

    cout << "\nAcc. Number : " << Client.get_AccountNumber();  

    cout << "\nPassword : " << Client.PinCode;  

    cout << "\nBalance : " << Client.AccountBalance;  

    cout << "\n_____ \n";
}

public:

static void ShowAddNewClientScreen()
{
    _DrawScreenHeader("\t Add New Client Screen", "");

    string AccountNumber = "";

    cout << "\nPlease Enter Account Number: ";
    AccountNumber = clsInputValidate::ReadString();
    while (clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nAccount Number Is Already Used, Choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }

    clsBankClient NewClient = clsBankClient::GetAddNewClientObject(AccountNumber);

    _ReadClientInfo(NewClient);

    clsBankClient::enSaveResults SaveResult;

    SaveResult = NewClient.save();

    switch (SaveResult)
    {
        case clsBankClient::enSaveResults::svSucceeded:
        {
            cout << "\nAccount Added SuccessFully :-)\n";
            _PrintClient(NewClient);
            break;
        }
        case clsBankClient::enSaveResults::svFailedEmptyObject:
        {
            cout << "\nError account was not saved because it's Empty";
            break;
        }
        case clsBankClient::enSaveResults::svFaildAccountNumberExists:
        {
            cout << "\nError account was not saved because account number is used!\n";
            break;
        }
    }
}

```

```
    }
}
}

};
```

وده التعديل اللي عملناه في كود الشاشه الرئيسيه

```
static void _ShowAddNewClientsScreen()
{
    // cout << "\nAdd New Client Screen Will be here...\n";
    clsAddNewClientScreen::ShowAddNewClientScreen();

}
```

## Lesson 10 - Delete Client Screen

شاشة الحذف

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsPerson.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"

class clsDeleteClientScreen :protected clsScreen
{

private:
    static void _PrintClient(clsBankClient Client)
    {
        cout << "\nClient Card:";
        cout << "\n_____";
        cout << "\nFirstName : " << Client.FirstName;
        cout << "\nLastName : " << Client.LastName;
        cout << "\nFull Name : " << Client.get_FullName();
        cout << "\nEmail : " << Client.Email;
        cout << "\nPhone : " << Client.Phone;
        cout << "\nAcc. Number : " << Client.get_AccountNumber();
        cout << "\nPassword : " << Client.PinCode;
        cout << "\nBalance : " << Client.AccountBalance;
        cout << "\n_____ \n";
    }

public:
    static void ShowDeleteClientScreen()
    {

        _DrawScreenHeader("\tDelete Client Screen", "");

        string AccountNumber = "";

        cout << "\nPlease Enter Account Number: ";
        AccountNumber = clsInputValidate::ReadString();
        while (!clsBankClient::IsClientExist(AccountNumber))
        {
            cout << "\nAccount number is not found, choose another one: ";
            AccountNumber = clsInputValidate::ReadString();
        }
    }
}
```

```

clsBankClient Client1 = clsBankClient::Find(AccountNumber);
_PrintClient(Client1);

cout << "\nAre you sure you want to delete this client y/n? ";

char Answer = 'n';
cin >> Answer;

if (Answer == 'y' || Answer == 'Y')
{

    if (Client1.Delete())
    {
        cout << "\nClient Deleted Successfully :-)\n";
        _PrintClient(Client1);
    }
    else
    {
        cout << "\nError Client Was not Deleted\n";
    }
}
};

};

```

وده التعديل اللي عملناه في كود الشاشه الرئيسيه

```

static void _ShowDeleteClientScreen()
{
    //cout << "\nDelete Client Screen Will be here...\n";
    clsDeleteClientScreen::ShowDeleteClientScreen();
}

```

## Lesson 11 - Update Client Screen

شاشة ال update

```

#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsPerson.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"

class clsUpdateClientScreen :protected clsScreen

{
private:

    static void _PrintClient(clsBankClient Client)
    {
        cout << "\nClient Card:";

        cout << "\n_____";

        cout << "\nFirstName : " << Client.FirstName;
        cout << "\nLastName : " << Client.LastName;
        cout << "\nFull Name : " << Client.get_FullName();
        cout << "\nEmail : " << Client.Email;
        cout << "\nPhone : " << Client.Phone;
        cout << "\nAcc. Number : " << Client.get_AccountNumber();
        cout << "\nPassword : " << Client.PinCode;
    }
};

```

```

cout << "\nBalance : " << Client.AccountBalance;
cout << "\n_____ \n";

}

static void ReadClientInfo(clsBankClient& Client)
{
    cout << "\nEnter FirstName: ";
    Client.FirstName = clsInputValidate::ReadString();

    cout << "\nEnter LastName: ";
    Client.LastName = clsInputValidate::ReadString();

    cout << "\nEnter Email: ";
    Client.Email = clsInputValidate::ReadString();

    cout << "\nEnter Phone: ";
    Client.Phone = clsInputValidate::ReadString();

    cout << "\nEnter PinCode: ";
    Client.PinCode = clsInputValidate::ReadString();

    cout << "\nEnter Account Balance: ";
    Client.AccountBalance = clsInputValidate::ReadFloatNumber();
}

public:

static void ShowUpdateClientScreen()
{
    _DrawScreenHeader("\tUpdate Client Screen", "");

    string AccountNumber = "";

    cout << "\nPlease Enter client Account Number: ";
    AccountNumber = clsInputValidate::ReadString();

    while (!clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nAccount number is not found, choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }

    clsBankClient Client1 = clsBankClient::Find(AccountNumber);

    _PrintClient(Client1);

    cout << "\nAre you sure you want to update this client y/n? ";

    char Answer = 'n';
    cin >> Answer;

    if (Answer == 'y' || Answer == 'Y')
    {

        cout << "\n\nUpdate Client Info:";
        cout << "\n_____ \n";

```

```

ReadClientInfo(Client1);

clsBankClient::enSaveResults SaveResult;

SaveResult = Client1.save();

switch (SaveResult)
{
case clsBankClient::enSaveResults::svSucceeded:
{
    cout << "\nAccount Updated Successfully :-)\n";

    _PrintClient(Client1);
    break;
}
case clsBankClient::enSaveResults::svFailedEmptyObject:
{
    cout << "\nError account was not saved because it's Empty";
    break;
}
}
}
}
};


```

وده التعديل اللي عملناه في كود الشاشه الرئيسيه

```

static void _ShowUpdateClientScreen()
{
//cout << "\nUpdate Client Screen Will be here...\n";
clsUpdateClientScreen::ShowUpdateClientScreen();

}

```

## Lesson 12 - Find Client Screen

شاشة البحث

```

#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsPerson.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"

class clsFindClientScreen :protected clsScreen
{

private:
    static void _PrintClient(clsBankClient Client)
    {
        cout << "\nClient Card:";

        cout << "\n_____";
        cout << "\nFirstName : " << Client.FirstName;
        cout << "\nLastName : " << Client.LastName;
        cout << "\nFull Name : " << Client.get_FullName();
        cout << "\nEmail : " << Client.Email;
        cout << "\nPhone : " << Client.Phone;
        cout << "\nAcc. Number : " << Client.get_AccountNumber();
    }
}

```

```

cout << "\nPassword : " << Client.PinCode;
cout << "\nBalance : " << Client.AccountBalance;
cout << "\n_____ \n";
}

public:

static void ShowFindClientScreen()
{
    _DrawScreenHeader("\tFind Client Screen","");
    string AccountNumber;
    cout << "\nPlease Enter Account Number: ";
    AccountNumber = clsInputValidate::ReadString();
    while (!clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nAccount number is not found, choose another one: ";
        AccountNumber = clsInputValidate::ReadString();
    }

    clsBankClient Client1 = clsBankClient::Find(AccountNumber);

    if (!Client1.IsEmpty())
    {
        cout << "\nClient Found :-)\n";
    }
    else
    {
        cout << "\nClient Was not Found :-(\n";
    }

    _PrintClient(Client1);
}
};


```

وده التعديل اللي عملناه في كود الشاشه الرئيسيه

```

static void _ShowFindClientScreen()
{
    // cout << "\nFind Client Screen Will be here...\n";
    clsFindClientScreen::ShowFindClientScreen();

}

```

## Lesson 13 - Transactions Screen

هنا هنعمل ال sub menu بتاعت ال transaction menu  
ده الكود بتاع ال transaction menu

```

#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsInputValidate.h"
#include <iomanip>

using namespace std;

class clsTransactionsScreen :protected clsScreen
{

```

```

private:
    enum enTransactionsMenueOptions {
        eDeposit = 1, eWithdraw = 2,
        eShowTotalBalance = 3, eShowMainMenue = 4
    };

    static short ReadTransactionsMenueOption()
    {
        cout << setw(37) << left << "" << "Choose what do you want to do? [1 to 4]? ";
        short Choice = clsInputValidate::ReadShortNumberBetween(1, 4, "Enter Number between 1 to 4? ");
        return Choice;
    }

    static void _ShowDepositScreen()
    {
        cout << "\n Deposit Screen will be here.\n";
    }

    static void _ShowWithdrawScreen()
    {
        cout << "\n Withdraw Screen will be here.\n";
    }

    static void _ShowTotalBalancesScreen()
    {
        cout << "\n Balances Screen will be here.\n";
    }

    static void _GoBackToTransactionsMenue()
    {
        cout << "\n\nPress any key to go back to Transactions Menue... ";
        system("pause>0");
        ShowTransactionsMenue();
    }

    static void _PerformTransactionsMenueOption(enTransactionsMenueOptions TransactionsMenueOption)
    {
        switch (TransactionsMenueOption)
        {
            case enTransactionsMenueOptions::eDeposit:
            {
                system("cls");
                _ShowDepositScreen();
                _GoBackToTransactionsMenue();
                break;
            }

            case enTransactionsMenueOptions::eWithdraw:
            {
                system("cls");
                _ShowWithdrawScreen();
                _GoBackToTransactionsMenue();
                break;
            }

            case enTransactionsMenueOptions::eShowTotalBalance:

```

```

    {
        system("cls");
        _ShowTotalBalancesScreen();
        _GoBackToTransactionsMenue();
        break;
    }

    case enTransactionsMenueOptions::eShowMainMenu:
    {
        //do nothing here the main screen will handle it :-);
    }
}

public:

static void ShowTransactionsMenue()
{
    system("cls");
    _DrawScreenHeader("\t Transactions Screen", "");

    cout << setw(37) << left << "=====\\n";
    cout << setw(37) << left << "\\t Transactions Menue\\n";
    cout << setw(37) << left << "=====\\n";
    cout << setw(37) << left << "\\t[1] Deposit.\\n";
    cout << setw(37) << left << "\\t[2] Withdraw.\\n";
    cout << setw(37) << left << "\\t[3] Total Balances.\\n";
    cout << setw(37) << left << "\\t[4] Main Menue.\\n";
    cout << setw(37) << left << "=====\\n";

    _PerformTransactionsMenueOption((enTransactionsMenueOptions)ReadTransactionsMenueOption());
}
};

};


```

وده التعديل اللي عملناه في كود الشاشه الرئيسيه

```

static void _ShowTransactionsMenue()
{
    // cout << "\\nTransactions Menue Will be here...\\n";
    clsTransactionsScreen::ShowTransactionsMenue();

}

```

## Lesson 14 - Deposit Screen

هنعمل ال deposit screen اللي في ال transactions menu بنفس الطرق اللي فاتت

```

Please enter AccountNumber? a
Client with [a] does not exist.

Please enter AccountNumber? A101

Client Card:

FirstName      : Mohammed
LastName       : Abu-Hadhoud
Full Name     : Mohammed Abu-Hadhoud
Email          : msaqer@gmail.com
Phone          : 199192
Acc. Number   : A101
Password       : 1234
Balance        : 9000

Please enter deposit amount? 2000

Are you sure you want to perform this transaction? y

Amount Deposited Successfully.

New Balance Is: 11000

```

قبل مانعمل الشاشه هندخل علي كلاس ال client ونحط فيه function خاصه بالايادع لانها خاصه  
بالداتا ملهاش علاقه بال ui  
فجيـنا جـوه كـلاس ال client وزـودـنا عـلـيـه عـمـلـيـه الإـيدـاع وـالـسـحب

```

void Deposit(double Amount)
{
    _AccountBalance += Amount;
    save();
}

void Withdraw(double Amount)
{
    _AccountBalance -= Amount;
    save();
}

```

ده الكود بتاع ال deposit screen

```

#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"

class clsDepositScreen : protected clsScreen
{

```

private:

```
static void _PrintClient(clsBankClient Client)
{
    cout << "\nClient Card:";
    cout << "\n_____";
    cout << "\nFirstName : " << Client.FirstName;
    cout << "\nLastName : " << Client.LastName;
    cout << "\nFull Name : " << Client.get_FullName();
    cout << "\nEmail : " << Client.Email;
    cout << "\nPhone : " << Client.Phone;
    cout << "\nAcc. Number : " << Client.get_AccountNumber();
    cout << "\nPassword : " << Client.PinCode;
    cout << "\nBalance : " << Client.AccountBalance;
    cout << "\n_____ \n";
}

static string _ReadAccountNumber()
{
    string AccountNumber = "";
    cout << "\nPlease enter AccountNumber? ";
    cin >> AccountNumber;
    return AccountNumber;
}
```

public:

```
static void ShowDepositScreen()
{
    _DrawScreenHeader("\t Deposit Screen","");
    string AccountNumber = _ReadAccountNumber();

    while (!clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nClient with [" << AccountNumber << "] does not exist.\n";
        AccountNumber = _ReadAccountNumber();
    }

    clsBankClient Client1 = clsBankClient::Find(AccountNumber);
    _PrintClient(Client1);

    double Amount = 0;
    cout << "\nPlease enter deposit amount? ";
    Amount = clsInputValidate::ReadDblNumber();

    cout << "\nAre you sure you want to perform this transaction? ";
    char Answer = 'n';
    cin >> Answer;

    if (Answer == 'Y' || Answer == 'y')
    {
        Client1.Deposit(Amount);
        cout << "\nAmount Deposited Successfully.\n";
        cout << "\nNew Balance Is: " << Client1.AccountBalance;
    }
}
```

```

    else
    {
        cout << "\nOperation was cancelled.\n";
    }

}

};

}

```

وده التعديل اللي عملناه في شاشة ال transactions

```

static void _ShowDepositScreen()
{
    //cout << "\n Deposit Screen will be here.\n";
    clsDepositScreen::ShowDepositScreen();
}

```

## Lesson 15 - Withdraw Screen

هنعمل تعديل على ال function بقاعدت ال withdraw بحيث انه مايقدرش يسحب اكتر من رصيده

```

bool Withdraw(double Amount)
{
    if (Amount > _AccountBalance)
    {
        return false;
    }
    else
    {
        _AccountBalance -= Amount;
        save();
        return true;
    }
}

```

ده الكود بقاعد شاشة السحب

```

#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"

class clsWithdrawScreen : protected clsScreen
{
private:

    static void _PrintClient(clsBankClient Client)
    {
        cout << "\nClient Card:";

        cout << "\n_____";
        cout << "\nFirstName : " << Client.FirstName;
        cout << "\nLastName : " << Client.LastName;
        cout << "\nFull Name : " << Client.get_FullName();
        cout << "\nEmail : " << Client.Email;
        cout << "\nPhone : " << Client.Phone;
        cout << "\nAcc. Number : " << Client.get_AccountNumber();
        cout << "\nPassword : " << Client.PinCode;
        cout << "\nBalance : " << Client.AccountBalance;
        cout << "\n_____ \n";
    }
}

```

```

static string _ReadAccountNumber()
{
    string AccountNumber = "";
    cout << "\nPlease enter AccountNumber? ";
    cin >> AccountNumber;
    return AccountNumber;
}

public:

static void ShowWithdrawScreen()
{
    _DrawScreenHeader("\t Withdraw Screen","");
    string AccountNumber = _ReadAccountNumber();

    while (!clsBankClient::IsClientExist(AccountNumber))
    {
        cout << "\nClient with [" << AccountNumber << "] does not exist.\n";
        AccountNumber = _ReadAccountNumber();
    }

    clsBankClient Client1 = clsBankClient::Find(AccountNumber);
    _PrintClient(Client1);

    double Amount = 0;
    cout << "\nPlease enter Withdraw amount? ";
    Amount = clsInputValidate::ReadDblNumber();

    cout << "\nAre you sure you want to perform this transaction? ";
    char Answer = 'n';
    cin >> Answer;

    if (Answer == 'Y' || Answer == 'y')
    {
        if (Client1.Withdraw(Amount))
        {
            cout << "\nAmount Withdraw Successfully.\n";
            cout << "\nNew Balance Is: " << Client1.AccountBalance;
        }
        else
        {
            cout << "\nCannot withdraw, Insufficient Balance!\n";
            cout << "\nAmout to withdraw is: " << Amount;
            cout << "\nYour Balance is: " << Client1.AccountBalance;
        }
    }
    else
    {
        cout << "\nOperation was cancelled.\n";
    }
}
};


```

وذه التعديل على شاشة ال transactions

```
static void _ShowWithdrawScreen()
```

```
{
//cout << "\n Withdraw Screen will be here.\n";
clsWithdrawScreen::ShowWithdrawScreen();
}
```

## Lesson 16 - Total Balances Screen

د کود شاشه ال total balances

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsBankClient.h"
#include <iomanip>
#include "clsUtil.h"

class clsTotalBalancesScreen : protected clsScreen
{
private:
    static void PrintClientRecordBalanceLine(clsBankClient Client)
    {
        cout << setw(25) << left << "" << "|" " << setw(15) << left << Client.get_AccountNumber();
        cout << "|" " << setw(40) << left << Client.get_FullName();
        cout << "|" " << setw(12) << left << Client.AccountBalance;
    }

public:
    static void ShowTotalBalances()
    {
        vector <clsBankClient> vClients = clsBankClient::GetClientsList();

        string Title = "\t Balances List Screen";
        string SubTitle = "\t (" + to_string(vClients.size()) + ") Client(s).";

        _DrawScreenHeader(Title, SubTitle);

        cout << setw(25) << left << "" << "\n\t\t_____";
        cout << " _____\n" << endl;

        cout << setw(25) << left << "" << "|" " << left << setw(15) << "Accout Number";
        cout << "|" " << left << setw(40) << "Client Name";
        cout << "|" " << left << setw(12) << "Balance";
        cout << setw(25) << left << "" << "\t\t_____";
        cout << " _____\n" << endl;

        double TotalBalances = clsBankClient::GetTotalBalances();

        if (vClients.size() == 0)
            cout << "\t\t\tNo Clients Available In the System!";
        else

            for (clsBankClient Client : vClients)
            {
                PrintClientRecordBalanceLine(Client);
                cout << endl;
            }

        cout << setw(25) << left << "" << "\n\t\t_____";
    }
}
```



```
static void _ShowDeleteUserScreen()
{
    cout << "\nDelete User Screen Will Be Here.\n";
}

static void _ShowUpdateUserScreen()
{
    cout << "\nUpdate User Screen Will Be Here.\n";
}

static void _ShowFindUserScreen()
{
    cout << "\nFind User Screen Will Be Here.\n";
}

static void _PerformManageUsersMenueOption(enManageUsersMenueOptions ManageUsersMenueOption)
{
    switch (ManageUsersMenueOption)
    {
        case enManageUsersMenueOptions::eListUsers:
        {
            system("cls");
            _ShowListUsersScreen();
            _GoBackToManageUsersMenue();
            break;
        }

        case enManageUsersMenueOptions::eAddNewUser:
        {
            system("cls");
            _ShowAddNewUserScreen();
            _GoBackToManageUsersMenue();
            break;
        }

        case enManageUsersMenueOptions::eDeleteUser:
        {
            system("cls");
            _ShowDeleteUserScreen();
            _GoBackToManageUsersMenue();
            break;
        }

        case enManageUsersMenueOptions::eUpdateUser:
        {
            system("cls");
            _ShowUpdateUserScreen();
            _GoBackToManageUsersMenue();
            break;
        }

        case enManageUsersMenueOptions::eFindUser:
        {
            system("cls");

            _ShowFindUserScreen();
        }
    }
}
```

```

        _GoBackToManageUsersMenue();
        break;
    }

    case enManageUsersMenueOptions::eMainMenue:
    {
        //do nothing here the main screen will handle it :-);
    }
}

}

public:

static void ShowManageUsersMenue()
{
    system("cls");
    _DrawScreenHeader("\t Manage Users Screen", "");

    cout << setw(37) << left << "" << "=====\\n";
    cout << setw(37) << left << "" << "\\t\\t Manage Users Menue\\n";
    cout << setw(37) << left << "" << "=====\\n";
    cout << setw(37) << left << "" << "\\t[1] List Users.\\n";
    cout << setw(37) << left << "" << "\\t[2] Add New User.\\n";
    cout << setw(37) << left << "" << "\\t[3] Delete User.\\n";
    cout << setw(37) << left << "" << "\\t[4] Update User.\\n";
    cout << setw(37) << left << "" << "\\t[5] Find User.\\n";
    cout << setw(37) << left << "" << "\\t[6] Main Menue.\\n";
    cout << setw(37) << left << "" << "=====\\n";

    _PerformManageUsersMenueOption((enManageUsersMenueOptions)ReadManageUsersMenueOption());
}
};

};

```

وده التعديل على الشاشه الرئيسيه

```

static void _ShowManageUsersMenue()
{
    // cout << "\\nUsers Menue Will be here...\\n";
    clsManageUsersScreen::ShowManageUsersMenue();

}

```

## Lesson 18 - Prepare User Class

هنعمل كلاس اليوزر وبيورث من ال person وبنعمله بنفس الطريقه بتاعت الكلاس بتاع ال client ده كلاس اليوزر

```

#pragma once
#include<iostream>
#include<fstream>
#include"clsPerson.h"
#include"clsString.h"
#include <string>

using namespace std;

```

```

class clsUser:public clsPerson
{
private:

    enum enMode {EmptyMode=0,UpdateMode=1,AddMode=2};

    enMode _Mode;
    string _UserName;
    string _PassWord;
    int _Permissions;
    bool _MarkedForDelete = false;

    //private functions
    static clsUser _ConvertLineToUserObject(string Line,string Seperator="#//#") {
        vector<string>vUserData =clsString::Split(Line,Seperator);
        clsUser User(enMode::UpdateMode, vUserData[0],vUserData[1], vUserData[2], vUserData[3],
vUserData[4], vUserData[5], stoi(vUserData[6]));
        return User;
    }

    static string _ConvertFromUserObjectToLine(clsUser User, string Seperaor = "#//#") {

        return User.FirstName + Seperaor
            + User.LastName + Seperaor
            + User.Email + Seperaor
            + User.Phone + Seperaor
            + User.UserName + Seperaor
            + User.PassWord + Seperaor
            + to_string(User.Permissions);
    }

    static vector<clsUser> _LoadUsersDataFromFile() {
        vector<clsUser>vUsers;
        fstream MyFile;
        MyFile.open("Users.txt",ios::in);

        if (MyFile.is_open()) {
            string Line = "";
            while (getline(MyFile,Line)) {
                vUsers.push_back(_ConvertLineToUserObject(Line));
            }
            MyFile.close();
        }
        return vUsers;
    }

    static void _SaveUsersDataToFile(vector<clsUser>vUsers) {
        fstream MyFile;
        MyFile.open("Users.txt", ios::out);

        if (MyFile.is_open()) {
            for (clsUser &User:vUsers) {
                if (User._MarkedForDelete==false) {
                    MyFile << _ConvertFromUserObjectToLine(User) << endl;
                }
            }
            MyFile.close();
        }
    }
}

```

```

static void _AddDataLineToFile(string stDataLine) {
    fstream MyFile;
    MyFile.open("Users.txt", ios::out | ios::app);

    if (MyFile.is_open()) {
        MyFile << stDataLine << endl;
        MyFile.close();
    }
}

void _Update() {
    vector<clsUser> vUsers = _LoadUsersDataFromFile();

    for (clsUser &User:vUsers) {
        if (User.UserName==UserName) {
            User = *this;
            break;
        }
    }
    _SaveUsersDataToFile(vUsers);
}

void _AddNew() {
    _AddDataLineToFile(_ConvertFromUserObjectToLine(*this));
}

static clsUser _GetEmptyUserObject() {
    return clsUser(enMode::EmptyMode, "", "", "", "", "", "", "", 0);
}

```

public:

```

//Constructor
clsUser(enMode Mode,string FirstName,string LastName,string Email, string Phone,string UserName
, string Password,int Permissions):clsPerson(FirstName,LastName,Email,Phone) {

    this->_Mode = Mode;
    this->_UserName = UserName;
    this->_PassWord = Password;
    this->_Permissions = Permissions;
}

//check Functions
bool IsEmpty() {
    return (_Mode == enMode::EmptyMode);
}

bool MarkedForDelete() {
    return _MarkedForDelete;
}

static bool IsUserExist(string UserName)
{

    clsUser User = clsUser::Find(UserName);
    return (!User.IsEmpty());
}

//setters

```

```

void set_UserName(string UserName) {
    this->_UserName = UserName;
}

void set_PassWord(string PassWord) {
    this->_PassWord = PassWord;
}

void set_Permissions(int Permissions) {
    this->_Permissions = Permissions;
}

//Getters
string get_UserName() { return _UserName; }
string get_PassWord() { return _PassWord; }
int get_Permissions() { return _Permissions; }

//declspec
_declspec(property(get = get_UserName, put = set_UserName))string UserName;
_declspec(property(get = get_PassWord, put = set_PassWord))string PassWord;
_declspec(property(get = get_Permissions, put = set_Permissions))int Permissions;

//public functions
static clsUser Find(string UserName) {
    fstream MyFile;
    MyFile.open("Users.txt", ios::in);

    if (MyFile.is_open()) {
        string Line = "";
        while (getline(MyFile, Line)) {
            clsUser User = _ConvertLineToUserObject(Line);
            if (User.UserName == UserName) {
                MyFile.close();
                return User;
            }
        }
        MyFile.close();
    }
    return _GetEmptyUserObject();
}

static clsUser Find(string UserName, string PassWord) {
    fstream MyFile;
    MyFile.open("Users.txt", ios::in);

    if (MyFile.is_open()) {
        string Line = "";
        while (getline(MyFile, Line)) {
            clsUser User = _ConvertLineToUserObject(Line);
            if (User.UserName == UserName && User.PassWord == PassWord) {
                MyFile.close();
                return User;
            }
        }
        MyFile.close();
    }
    return _GetEmptyUserObject();
}

```

```

enum enSaveResults { svFaildEmptyObject = 0, svSucceeded = 1, svFaildUserExists = 2 };

enSaveResults Save() {
    switch (_Mode)
    {
        case clsUser::EmptyMode:
            if (IsEmpty())
                return svFaildEmptyObject;
            break;
        case clsUser::UpdateMode:
            _Update();
            return svSucceeded;
            break;
        case clsUser::AddMode:
            if (clsUser::IsUserExist(this->UserName)) {
                return enSaveResults::svFaildUserExists;
            }
            else {
                _AddNew();
                this->_Mode = UpdateMode;
                return svSucceeded;
            }
            break;
        default:
            break;
    }
}

bool Delete() {
    vector<clsUser> vUsers=_LoadUsersDataFromFile();

    for (clsUser& U : vUsers)
    {
        if (U.UserName == _UserName)
        {
            U._MarkedForDelete = true;
            break;
        }
    }

    _SaveUsersDataToFile(vUsers);

    *this = _GetEmptyUserObject();

    return true;
}

static clsUser GetAddNewUserObject(string UserName)
{
    return clsUser(enMode::AddMode, "", "", "", "", UserName, "", 0);
}

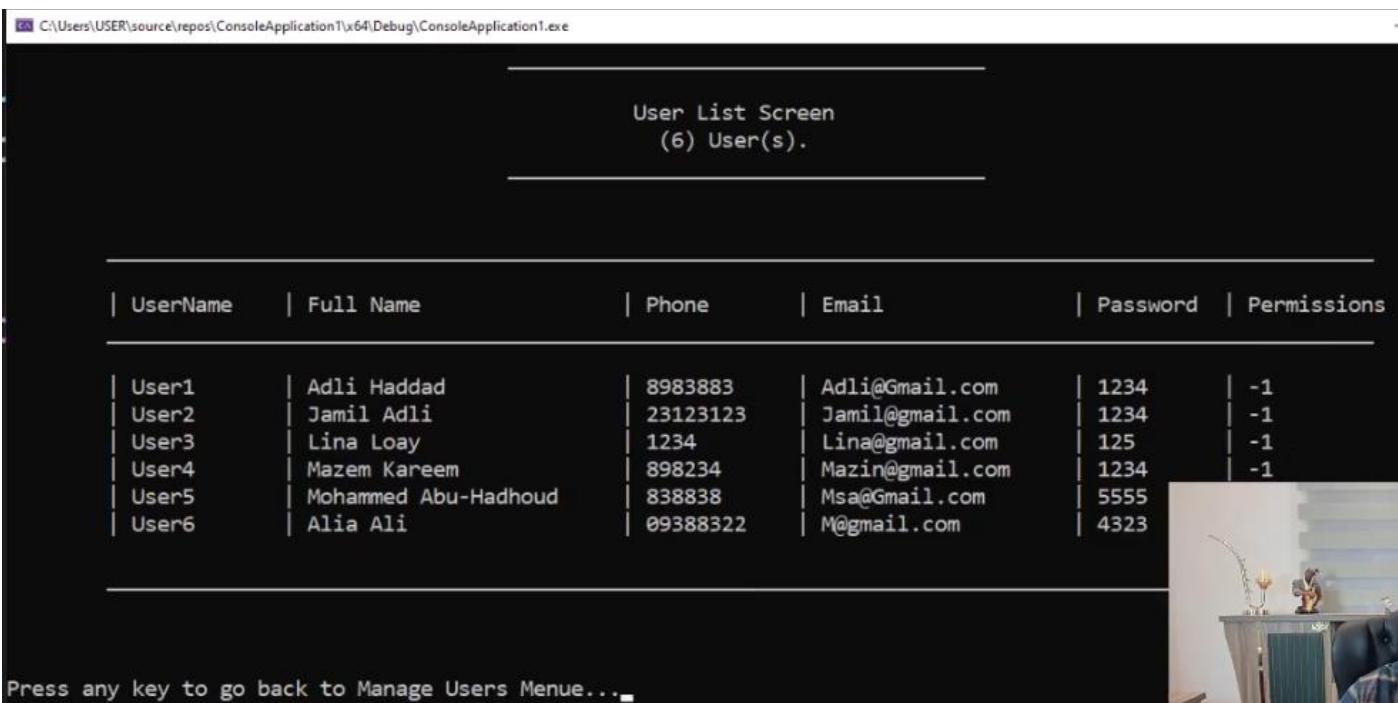
static vector <clsUser> GetUsersList()
{
    return _LoadUsersDataFromFile();
}

};

```

## Lesson 19 - List Users Screen

هنعمل الشاشه بتاعت list users



UserName	Full Name	Phone	Email	Password	Permissions
User1	Adli Haddad	8983883	Adli@Gmail.com	1234	-1
User2	Jamil Adli	23123123	Jamil@gmail.com	1234	-1
User3	Lina Loay	1234	Lina@gmail.com	125	-1
User4	Mazem Kareem	898234	Mazin@gmail.com	1234	-1
User5	Mohammed Abu-Hadhoud	838838	Msa@gmail.com	5555	-1
User6	Alia Ali	09388322	M@gmail.com	4323	-1

ده الكود بتاع شاشة ال list users

```
#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsUser.h"
#include <iomanip>

class clsListUsersScreen :protected clsScreen
{
private:
    static void _PrintUserRecordLine(clsUser User)
    {
        cout << setw(8) << left << "" << "|" " << setw(12) << left << User.UserName;
        cout << "|" " << setw(25) << left << User.get_FullName();
        cout << "|" " << setw(12) << left << User.Phone;
        cout << "|" " << setw(20) << left << User.Email;
        cout << "|" " << setw(10) << left << User.PassWord;
        cout << "|" " << setw(12) << left << User.Permissions;
    }

public:
    static void ShowUsersList()
    {
        vector <clsUser> vUsers = clsUser::GetUsersList();

        string Title = "\t User List Screen";
        string SubTitle = "\t (" + to_string(vUsers.size()) + ") User(s)";

        _DrawScreenHeader(Title, SubTitle);

        cout << setw(8) << left << "" << "\n\t";
    }
}
```

```
cout << " _____ \n" << endl;

cout << setw(8) << left << "" << "| " << left << setw(12) << "UserName";
cout << "| " << left << setw(25) << "Full Name";
cout << "| " << left << setw(12) << "Phone";
cout << "| " << left << setw(20) << "Email";
cout << "| " << left << setw(10) << "Password";
cout << "| " << left << setw(12) << "Permissions";
cout << setw(8) << left << "" << "\n\t_____";
cout << " _____ \n" << endl;

if (vUsers.size() == 0)
    cout << "\t\ta\tNo Users Available In the System!";
else

    for (clsUser User : vUsers)
    {

        _PrintUserRecordLine(User);
        cout << endl;
    }

    cout << setw(8) << left << "" << "\n\t_____";
    cout << " _____ \n" << endl;
}

};

وذه التعديل على شاشة ال manage user
```

```
static void _ShowListUsersScreen()
{
//cout << "\nList Users Screen Will Be Here.\n";
clsListUsersScreen::ShowUsersList();

}
```

## Lesson 20 - Add New User Screen

ده التعديل اللي هنعمله على كلاس اليوزر في ال public  
ده عشان اللي هيدخل الداتا يختار منهم

```
enum enPermissions {
    eAll = -1, pListClients = 1, pAddNewClient = 2, pDeleteClient = 4,
    pUpdateClients = 8, pFindClient = 16, pTransactions = 32, pManageUsers = 64
};
```

ده كود شاشة ال add new user

```
#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsUser.h"
#include "clsInputValidate.h"
#include <iomanip>

class clsAddNewUserScreen : protected clsScreen
{
private:

    static void _ReadUserInfo(clsUser& User)
    {
        cout << "\nEnter FirstName: ";
```

```

User.FirstName = clsInputValidate::ReadString();

cout << "\nEnter LastName: ";
User.LastName = clsInputValidate::ReadString();

cout << "\nEnter Email: ";
User.Email = clsInputValidate::ReadString();

cout << "\nEnter Phone: ";
User.Phone = clsInputValidate::ReadString();

cout << "\nEnter Password: ";
User.PassWord = clsInputValidate::ReadString();

cout << "\nEnter Permission: ";
User.Permissions = _ReadPermissionsToSet();
}

static void _PrintUser(clsUser User)
{
    cout << "\nUser Card:";

    cout << "\n_____";
    cout << "\nFirstName : " << User.FirstName;
    cout << "\nLastName : " << User.LastName;
    cout << "\nFull Name : " << User.get_FullName();
    cout << "\nEmail : " << User.Email;
    cout << "\nPhone : " << User.Phone;
    cout << "\nUser Name : " << User.UserName;
    cout << "\nPassword : " << User.PassWord;
    cout << "\nPermissions : " << User.Permissions;
    cout << "\n_____ \n";
}

static int _ReadPermissionsToSet()
{
    int Permissions = 0;
    char Answer = 'n';

    cout << "\nDo you want to give full access? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        return -1;
    }

    cout << "\nDo you want to give access to : \n ";

    cout << "\nShow Client List? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {

        Permissions += clsUser::enPermissions::pListClients;
    }

    cout << "\nAdd New Client? y/n? ";
}

```

```

    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pAddNewClient;
    }

    cout << "\nDelete Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pDeleteClient;
    }

    cout << "\nUpdate Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pUpdateClients;
    }

    cout << "\nFind Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pFindClient;
    }

    cout << "\nTransactions? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pTransactions;
    }

    cout << "\nManage Users? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pManageUsers;
    }

    return Permissions;
}

public:

static void ShowAddNewUserScreen()
{
    _DrawScreenHeader("\t Add New User Screen","");
    string UserName = "";

    cout << "\nPlease Enter UserName: ";
    UserName = clsInputValidate::ReadString();
    while (clsUser::IsUserExist(UserName))
    {
        cout << "\nUserName Is Already Used, Choose another one: ";
        UserName = clsInputValidate::ReadString();
}

```

```
}
```

```
clsUser NewUser = clsUser::GetAddNewUserObject(UserName);
```

```
_ReadUserInfo(NewUser);
```

```
clsUser::enSaveResults SaveResult;
```

```
SaveResult = NewUser.Save();
```

```
switch (SaveResult)
```

```
{
```

```
case clsUser::enSaveResults::svSucceeded:
```

```
{
```

```
cout << "\nUser Added Successfully :-)\n";
```

```
_PrintUser(NewUser);
```

```
break;
```

```
}
```

```
case clsUser::enSaveResults::svFaildEmptyObject:
```

```
{
```

```
cout << "\nError User was not saved because it's Empty";
```

```
break;
```

```
}
```

```
case clsUser::enSaveResults::svFaildUserExists:
```

```
{
```

```
cout << "\nError User was not saved because UserName is used!\n";
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
};
```

وذه التعديل على شاشة ال manage user

```
static void _ShowAddNewUserScreen()
```

```
{
```

```
// cout << "\nAdd New User Screen Will Be Here.\n";
```

```
clsAddNewUserScreen::ShowAddNewUserScreen();
```

```
}
```

## Lesson 21 - Delete User Screen

ده الكود بتاع الشاشه

```
#pragma once
```

```
#include <iostream>
```

```
#include "clsScreen.h"
```

```
#include "clsPerson.h"
```

```
#include "clsUser.h"
```

```
#include "clsInputValidate.h"
```

```
class clsDeleteUserScreen :protected clsScreen
```

```
{
```

```
private:
```

```
    static void _PrintUser(clsUser User)
```

```

{
    cout << "\nUser Card:";
    cout << "\n_____";
    cout << "\nFirstName : " << User.FirstName;
    cout << "\nLastName : " << User.LastName;
    cout << "\nFull Name : " << User.get_FullName();
    cout << "\nEmail : " << User.Email;
    cout << "\nPhone : " << User.Phone;
    cout << "\nUser Name : " << User.UserName;
    cout << "\nPassword : " << User.PassWord;
    cout << "\nPermissions : " << User.Permissions;
    cout << "\n_____\\n";
}

public:
    static void ShowDeleteUserScreen()
    {
        _DrawScreenHeader("\tDelete User Screen","");
        string UserName = "";
        cout << "\nPlease Enter UserName: ";
        UserName = clsInputValidate::ReadString();
        while (!clsUser::IsUserExist(UserName))
        {
            cout << "\nUser is not found, choose another one: ";
            UserName = clsInputValidate::ReadString();
        }
        clsUser User1 = clsUser::Find(UserName);
        _PrintUser(User1);

        cout << "\nAre you sure you want to delete this User y/n? ";
        char Answer = 'n';
        cin >> Answer;

        if (Answer == 'y' || Answer == 'Y')
        {
            if (User1.Delete())
            {
                cout << "\nUser Deleted Successfully :-)\\n";
                _PrintUser(User1);
            }
            else
            {
                cout << "\nError User Was not Deleted\\n";
            }
        }
    }
};

```

وَهُوَ التَّعْدِيلُ عَلَى الشَّاشَةِ manage users

```

static void _ShowDeleteUserScreen()
{
    // cout << "\nDelete User Screen Will Be Here.\\n";
    clsDeleteUserScreen::ShowDeleteUserScreen();
}

```

```
}
```

## Lesson 22 - Update User Screen

ده الكود بناء الشاشه

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsPerson.h"
#include "clsUser.h"
#include "clsInputValidate.h"

class clsUpdateUserScreen :protected clsScreen

{
private:

    static void _ReadUserInfo(clsUser& User)
    {
        cout << "\nEnter FirstName: ";
        User.FirstName = clsInputValidate::ReadString();

        cout << "\nEnter LastName: ";
        User.LastName = clsInputValidate::ReadString();

        cout << "\nEnter Email: ";
        User.Email = clsInputValidate::ReadString();

        cout << "\nEnter Phone: ";
        User.Phone = clsInputValidate::ReadString();

        cout << "\nEnter Password: ";
        User.PassWord = clsInputValidate::ReadString();

        cout << "\nEnter Permission: ";
        User.Permissions = _ReadPermissionsToSet();
    }

    static void _PrintUser(clsUser User)
    {
        cout << "\nUser Card:";

        cout << "\n_____";
        cout << "\nFirstName : " << User.FirstName;
        cout << "\nLastName : " << User.LastName;
        cout << "\nFull Name : " << User.get_FullName();
        cout << "\nEmail : " << User.Email;
        cout << "\nPhone : " << User.Phone;
        cout << "\nUser Name : " << User.UserName;
        cout << "\nPassword : " << User.PassWord;
        cout << "\nPermissions : " << User.Permissions;
        cout << "\n_____ \n";
    }

    static int _ReadPermissionsToSet()
    {
        int Permissions = 0;
```

```
char Answer = 'n';

cout << "\nDo you want to give full access? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    return -1;
}

cout << "\nDo you want to give access to : \n ";

cout << "\nShow Client List? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{

    Permissions += clsUser::enPermissions::pListClients;
}

cout << "\nAdd New Client? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pAddNewClient;
}

cout << "\nDelete Client? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pDeleteClient;
}

cout << "\nUpdate Client? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pUpdateClients;
}

cout << "\nFind Client? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pFindClient;
}

cout << "\nTransactions? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pTransactions;
}

cout << "\nManage Users? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
```

```

    Permissions += clsUser::enPermissions::pManageUsers;
}

return Permissions;

}

public:

static void ShowUpdateUserScreen()
{
    _DrawScreenHeader("\tUpdate User Screen", "");

    string UserName = "";

    cout << "\nPlease Enter User UserName: ";
    UserName = clsInputValidate::ReadString();

    while (!clsUser::IsUserExist(UserName))
    {
        cout << "\nAccount number is not found, choose another one: ";
        UserName = clsInputValidate::ReadString();
    }

    clsUser User1 = clsUser::Find(UserName);

    _PrintUser(User1);

    cout << "\nAre you sure you want to update this User y/n? ";

    char Answer = 'n';
    cin >> Answer;

    if (Answer == 'y' || Answer == 'Y')
    {

        cout << "\n\nUpdate User Info:";
        cout << "\n_____ \n";

        _ReadUserInfo(User1);

        clsUser::enSaveResults SaveResult;

        SaveResult = User1.Save();

        switch (SaveResult)
        {
        case clsUser::enSaveResults::svSucceeded:
        {
            cout << "\nUser Updated Successfully :-) \n";

            _PrintUser(User1);
            break;
        }
        case clsUser::enSaveResults::svFaildEmptyObject:
        {
            cout << "\nError User was not saved because it's Empty";
            break;
        }
    }
}

```

```

        }

    }

}

};

}

```

وذه التعديل على الشاشه manage users

```

static void _ShowUpdateUserScreen()
{
    // cout << "\nUpdate User Screen Will Be Here.\n";
    clsUpdateUserScreen::ShowUpdateUserScreen();
}

```

## Lesson 23 - Find User Screen

ده الكود بتاع الشاشه

```

#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsPerson.h"
#include "clsUser.h"
#include "clsInputValidate.h"

class clsFindUserScreen :protected clsScreen
{

private:
    static void _PrintUser(clsUser User)
    {
        cout << "\nUser Card:";

        cout << "\n_____";
        cout << "\nFirstName : " << User.FirstName;
        cout << "\nLastName : " << User.LastName;
        cout << "\nFull Name : " << User.get_FullName();
        cout << "\nEmail : " << User.Email;
        cout << "\nPhone : " << User.Phone;
        cout << "\nUserName : " << User.UserName;
        cout << "\nPassword : " << User.PassWord;
        cout << "\nPermissions : " << User.Permissions;
        cout << "\n_____ \n";
    }

public:
    static void ShowFindUserScreen()
    {

        _DrawScreenHeader("\t Find User Screen","");
        string UserName;
        cout << "\nPlease Enter UserName: ";
        UserName = clsInputValidate::ReadString();
        while (!clsUser::IsUserExist(UserName))
        {
            cout << "\nUser is not found, choose another one: ";
            UserName = clsInputValidate::ReadString();
        }
    }
}

```

```

    }

    clsUser User1 = clsUser::Find(UserName);

    if (!User1.IsEmpty())
    {
        cout << "\nUser Found :-)\n";
    }
    else
    {
        cout << "\nUser Was not Found :-(\n";
    }

    _PrintUser(User1);

}

};

}

```

## وده التعديل على الشاشه manage users

```

static void _ShowFindUserScreen()
{
    //cout << "\nFind User Screen Will Be Here.\n";
    clsFindUserScreen::ShowFindUserScreen();
}

```

## Lesson 24 - Login

هنعمل شاشة تسجيل الدخول وهنبدأ بيها البرنامج بدل الشاشه الرئيسيه لو دخل البيانات صح يدور عاليوزر ويعرفه علي مستوى السيستم كل هيعمله global ولو دخل البيانات غلط هيدخل في while loop لحد مايدخلها صح عشان تعرف global user بتعرف فيه object من النوع user واي كلاس تحتاجه هيعمل include للملف ده

## ده الكود بتاع ال global user (header file)

```

#pragma once
#include <iostream>
#include "clsUser.h"
clsUser CurrentUser = clsUser::Find("", "");

```

## ده الكود بتاع شاشة ال login

```

#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsUser.h"
#include <iomanip>
#include "clsMainScreen.h"
#include "Global.h"

class clsLoginScreen :protected clsScreen
{
private:

    static void _Login()
    {
        bool LoginFaild = false;

```

```

string Username, Password;
do
{
    if (LoginFaild)
    {
        cout << "\nInvlaid Username/Password!\n\n";
    }

    cout << "Enter Username? ";
    cin >> Username;

    cout << "Enter Password? ";
    cin >> Password;

    CurrentUser = clsUser::Find(Username, Password);

    LoginFaild = CurrentUser.IsEmpty();

} while (LoginFaild);

clsMainScreen::ShowMainMenue();

}

public:

static void ShowLoginScreen()
{
    system("cls");
    _DrawScreenHeader("\t Login Screen","");
    _Login();

}

};


```

وده التعديل على ال main

```

#include <iostream>
#include "clsLoginScreen.h"

int main()

{
    clsLoginScreen::ShowLoginScreen();

    system("pause>0");
    return 0;
}

```

## Lesson 25 - Logout

ال log out كل اللي بيعمله انه بيفضي ال object اللي معمول global ده التعديل عالشاشة الرئيسية ماتنساش تعمل دي

#include "Global.h"

```
/* static void _ShowEndScreen()
```

```

{
    cout << "\nEnd Screen Will be here...\n";

}*/



static void _Logout()
{

    CurrentUser = clsUser::Find("", "");
    //then it will go back to main function.
}

case enMainMenueOptions::eExit:
    system("cls");
    _Logout();
    //Login();

    break;

```

## Lesson 26 - Show Login Screen at Logout

هنا احنا عايزين بدل ماندوس logout ونخرج من البرنامج خالص  
 لا احنا عايزينه نفتح شاشة ال login  
 طيب فيه حد ممكن يقولك اتنا نكتب الكود كده

```

static void _Logout()
{

    CurrentUser = clsUser::Find("", "");
    clsLoginScreen::ShowLoginScreen();
    //then it will go back to main function.
}

```

احنا في الكود اللي فات جينا جوه ال logout function بتاعت ال login screen فيها ال logout لو جيت عملت كده البرنامج مش هيشتغل وهضربي طيب ليه؟

قالك بسبب حاجه اسمها circular reference ودي منو عه في البرمجه يعني ايه الكلام ده؟

عشان نفهمها تعالى نبص عالمصار اللي بيمشي فيه البرنامج في حالة اتنا خلينا البرنامج يقفل لما نعمل ال log out وفي حالة الكود اللي كتبناه

في الأول البرنامج كان بيمشي كده

Main Function >> Login Screen >> Main Menu Screen  
 >> Logout Function >> Main menu Screen >> Login Screen  
 >> Main Function

فكان لما بيدخل ينفذ function معينه كان بيرجع تاني لمكانه الأولاني عشان ينفذ بقى الكود

طيب تعالى بقى نشوف بعد التعديل اللي عملناه

Main Function >> Login Screen >> Main Menu Screen

>> Logout Function >> Login Screen >> Main Menu Screen >> Logout Function >> Login Screen >> Main Menu Screen Logout Function >> Login Screen >> Main Menu Screen

وهكذا الي مالانهايه ومش هيكون فيه نقطه عودة بينتهى فيها البرنامج زي الاول يعني هتلاقيه بيفتح stacks عمال علي بطال طبيعي يضرب منك كمان انت لما وصلت لشاشة ال login وبعدها فتحت القائمه الرئيسيه ال compiler هيكون واقف على شاشة ال login ومستني الكود اللي جوه القائمه الرئيسيه يخلص عشان ينفذ باقي الكود اللي في شاشة ال login فلما انت تيجي جوه ال logout تطلب منه يرجع يشغل شاشة ال login تاني أصبحت شاشة ال login مستتبه ال logout تخلص واصبح ال login مستني شاشة ال logout تخلص واصبح عندك circular reference طيب الحل ايه؟

قالك انه الحل انك تعمل while loop في ال main نفسه بحيث انه لما ييجي يصل للمرحله دي  
Main Function >> Login Screen >> Main Menu Screen  
>> Logout Function >> Main menu Screen >> Login Screen  
>> Main Function

يعيد تشغيل شاشة تسجيل الدخول تاني زي كده

```
#include <iostream>
#include "clsLoginScreen.h"

int main()

{
    while (true) { clsLoginScreen::ShowLoginScreen(); }

    system("pause>0");
    return 0;
}
```

## Lesson 27 - Apply Permissions

هنستغل على ال permissions او حاجه هنعملها هنروح عند الكلاس user ونحط الكود ده شرحناه قبل كده

```
bool CheckAccessPermission(enPermissions Permission)
{
    if (this->Permissions == enPermissions::eAll)
        return true;

    if ((Permission & this->Permissions) == Permission)
        return true;
    else
        return false;
}
```

بعجين احنا عايزين شاشة ال access denied تكون ظاهره في كل الشاشات لو اليوزر ماعندوش صلاحيه ليها



```

cout << " " << left << setw(20) << "Email";
cout << " " << left << setw(10) << "Pin Code";
cout << " " << left << setw(12) << "Balance";
cout << setw(8) << left << "" << "\n\t_____";
cout << " _____\n" << endl;

if (vClients.size() == 0)
    cout << "\t\t\tNo Clients Available In the System!";
else

    for (clsBankClient Client : vClients)
    {

        PrintClientRecordLine(Client);
        cout << endl;
    }

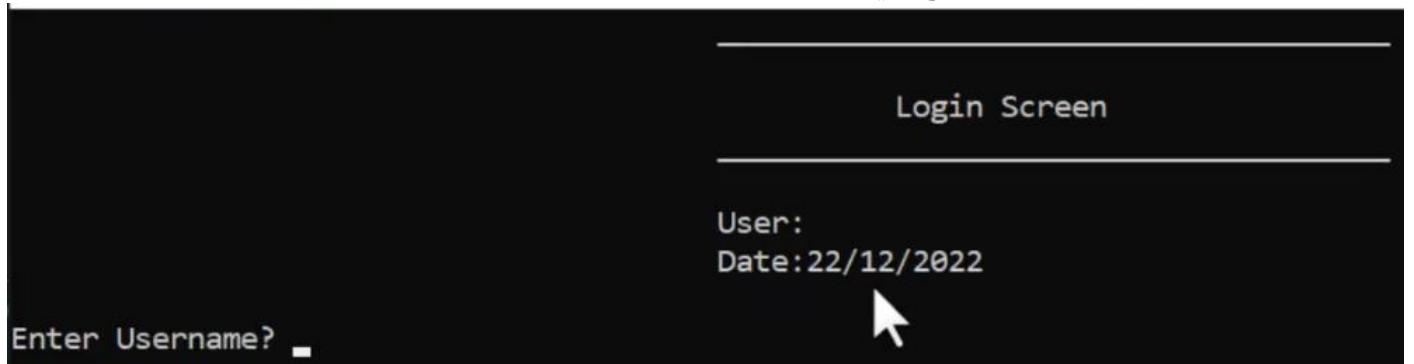
    cout << setw(8) << left << "" << "\n\t_____";
    cout << " _____\n" << endl;
}

```

انا مش هحط باقي الشاشات عشان مااكررش بدون داعي  
وماتنساش تضيف ال user للكلas global

### Add Date and Logged In User to All Screens

عاوزين نضيف اسم اليوزر والتاريخ في كل الشاشات





```

{
    if (!CurrentUser.CheckAccessPermission(Permission))
    {
        cout << "\t\t\t\t\t";
        cout << "\n\n\t\t\t\t Access Denied! Contact your Admin.";
        cout << "\n\t\t\t\t\t\n\n";
        return false;
    }
    else
    {
        return true;
    }
}

```

### Lock the System After 3 failed Logins

عاوزين لو الايوزر عمل login غلط 3 مرات يطلع من السيستم

Microsoft Visual Studio Debug Console

User:  
Date: 22/12/2022

Enter Username? a

Enter Password? a

Invlaid Username/Password!

You have 2 Trials to login.

Enter Username? a

Enter Password? a

Invlaid Username/Password!

You have 1 Trials to login.

Enter Username? a

Enter Password? a

Invlaid Username/Password!

You have 0 Trials to login.



Your are Locked after 3 faild trails

C:\Users\USER\source\repos\ConsoleApplication1\x64\Debug\Cons  
To automatically close the console when debugging stops, enab

## Solution

```
#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsUser.h"
#include <iomanip>
#include "clsMainScreen.h"
#include "Global.h"

class clsLoginScreen :protected clsScreen
{
private:
```

```

static bool _Login()
{
    bool LoginFaild = false;
    short FaildLoginCount = 0;

    string Username, Password;
    do
    {

        if (LoginFaild)
        {
            FaildLoginCount++;

            cout << "\nInvlaid Username/Password!";
            cout << "\nYou have " << (3 - FaildLoginCount)
                << " Trial(s) to login.\n\n";
        }

        if (FaildLoginCount == 3)
        {
            cout << "\nYour are Locked after 3 faild trails \n\n";
            return false;
        }

        cout << "Enter Username? ";
        cin >> Username;

        cout << "Enter Password? ";
        cin >> Password;

        CurrentUser = clsUser::Find(Username, Password);

        LoginFaild = CurrentUser.IsEmpty();

    } while (LoginFaild);

    clsMainScreen::ShowMainMenue();
    return true;
}

public:

static bool ShowLoginScreen()
{
    system("cls");
    _DrawScreenHeader("\t Login Screen","");
    return _Login();
}

};

#include <iostream>
#include "clsLoginScreen.h"

int main()

{
    while (true)

```

```

{
    if (!clsLoginScreen::ShowLoginScreen())
    {
        break;
    }

}

return 0;
}

```

## Register Logins In a Log File

محتاجين نعمل ملف يكون فيه كل تاريخ ووقت كل عملية دخول علي السيستم



## Solution

ده اللي زودناه في كلاس ال date

```

static string GetSystemDateTimeString()
{
    //system datetime string
    time_t t = time(0);
    tm* now = localtime(&t);

    short Day, Month, Year, Hour, Minute, Second;

    Year = now->tm_year + 1900;
    Month = now->tm_mon + 1;
    Day = now->tm_mday;
    Hour = now->tm_hour;
    Minute = now->tm_min;
    Second = now->tm_sec;

    return (to_string(Day) + "/" + to_string(Month) + "/"
            + to_string(Year) + " - "
            + to_string(Hour) + ":" + to_string(Minute)
            + ":" + to_string(Second));
}

```

وده اللي زودناه في كلاس ال user

```

//private functions
string _PrepareLoginRecord(string Seperator = "#/#")
{
    string LoginRecord = "";
    LoginRecord += clsDate::GetSystemDateTimeString() + Seperator;
    LoginRecord += UserName + Seperator;
}

```

```

        LoginRecord += PassWord + Seperator;
        LoginRecord += to_string(Permissions);
        return LoginRecord;
    }

    //public functions
void RegisterLogIn()
{
    string stDataLine = _PrepareLogInRecord();

    fstream MyFile;
    MyFile.open("LoginRegister.txt", ios::out | ios::app);

    if (MyFile.is_open())
    {

        MyFile << stDataLine << endl;

        MyFile.close();
    }
}

```

وده اللي زودناه في شاشة ال login

```

static bool _Login()
{
    bool LoginFaild = false;
    short FaildLoginCount = 0;

    string Username, Password;
    do
    {

        if (LoginFaild)
        {
            FaildLoginCount++;

            cout << "\nInvalid Username/Password!";
            cout << "\nYou have " << (3 - FaildLoginCount)
                << " Trial(s) to login.\n\n";
        }

        if (FaildLoginCount == 3)
        {
            cout << "\nYour are Locked after 3 faild trails \n\n";
            return false;
        }

        cout << "Enter Username? ";
        cin >> Username;

        cout << "Enter Password? ";
        cin >> Password;

        CurrentUser = clsUser::Find(Username, Password);

        LoginFaild = CurrentUser.IsEmpty();
    } while (LoginFaild);
}

```

```
    CurrentUser.RegisterLogin();
    clsMainScreen::ShowMainMenue();
    return true;
}
```

وماتنساش السطر ده في كلاس ال user  
#include "clsDate.h"

## Show Login Register Screen

عاوزين نعمل شاشه تعرضلنا ال login



Date/Time	UserName	Password	Permissions
22/12/2022 - 18:8:3	User1	1234	0
22/12/2022 - 18:8:28	User2	1234	-1
22/12/2022 - 18:8:57	User4	1234	-1
22/12/2022 - 18:9:15	User1	1234	0
25/12/2022 - 12:40:12	User2	1234	-1
25/12/2022 - 12:54:25	User2	1234	-1

 At the bottom, it says 'Press any key to go back to Main Menue...'."/>

## Solution

الفكرة هنا انك بتجمع المتغيرات في **structure** داخل كلاس ال **user**  
دي الاكواد اللي حصل فيها تعديل و هعلم عالتتعديل  
ده كلاس ال **user**

```
#pragma once
#include<iostream>
#include<fstream>
#include"clsPerson.h"
#include"clsString.h"
#include"clsDate.h"
#include <string>

using namespace std;

class clsUser:public clsPerson
{
private:

    enum enMode {EmptyMode=0,UpdateMode=1,AddMode=2};

    enMode _Mode;
    string _UserName;
    string _PassWord;
    int _Permissions;
    bool _MarkedForDelete = false;

    struct stLoginRegisterRecord;

    //private functions
    static stLoginRegisterRecord _ConvertLoginRegisterLineToRecord(string Line, string Seperator = "#/#")
    {
        stLoginRegisterRecord LoginRegisterRecord;
        vector <string> LoginRegisterDataLine = clsString::Split(Line, Seperator);
    }
}
```

```

        LoginRegisterRecord.DateTime = LoginRegisterDataLine[0];
        LoginRegisterRecord.UserName = LoginRegisterDataLine[1];
        LoginRegisterRecord.Password = LoginRegisterDataLine[2];
        LoginRegisterRecord.Permissions = stoi(LoginRegisterDataLine[3]);

        return LoginRegisterRecord;
    }

    string _PrepareLogInRecord(string Seperator = "#//#")
    {
        string LoginRecord = "";
        LoginRecord += clsDate::GetSystemDateTimeString() + Seperator;
        LoginRecord += UserName + Seperator;
        LoginRecord += PassWord + Seperator;
        LoginRecord += to_string(Permissions);
        return LoginRecord;
    }

    static clsUser _ConvertLineToUserObject(string Line, string Seperator = "#//#") {
        vector<string> vUserData = clsString::Split(Line, Seperator);
        clsUser User(enMode::UpdateMode, vUserData[0], vUserData[1], vUserData[2], vUserData[3],
vUserData[4], vUserData[5], stoi(vUserData[6]));
        return User;
    }

    static string _ConvertFromUserObjectToLine(clsUser User, string Seperaor = "#//#") {

        return User.FirstName + Seperaor
            + User.LastName + Seperaor
            + User.Email + Seperaor
            + User.Phone + Seperaor
            + User.UserName + Seperaor
            + User.PassWord + Seperaor
            + to_string(User.Permissions);
    }

    static vector<clsUser> _LoadUsersDataFromFile() {
        vector<clsUser> vUsers;
        fstream MyFile;
        MyFile.open("Users.txt", ios::in);

        if (MyFile.is_open()) {
            string Line = "";
            while (getline(MyFile, Line)) {
                vUsers.push_back(_ConvertLineToUserObject(Line));
            }
            MyFile.close();
        }
        return vUsers;
    }

    static void _SaveUsersDataToFile(vector<clsUser> vUsers) {
        fstream MyFile;
        MyFile.open("Users.txt", ios::out);

        if (MyFile.is_open()) {
            for (clsUser &User: vUsers) {
                if (User._MarkedForDelete == false) {
                    MyFile << _ConvertFromUserObjectToLine(User) << endl;
                }
            }
        }
    }
}

```

```

        }
    }
    MyFile.close();
}

static void _AddDataLineToFile(string stDataLine) {
    fstream MyFile;
    MyFile.open("Users.txt", ios::out | ios::app);

    if (MyFile.is_open()) {
        MyFile << stDataLine << endl;
        MyFile.close();
    }
}

void _Update() {
    vector<clsUser> vUsers = _LoadUsersDataFromFile();

    for (clsUser &User:vUsers) {
        if (User.UserName==UserName) {
            User = *this;
            break;
        }
    }
    _SaveUsersDataToFile(vUsers);
}

void _AddNew() {
    _AddDataLineToFile(_ConvertFromUserObjectToLine(*this));
}

static clsUser _GetEmptyUserObject() {
    return clsUser(enMode::EmptyMode, "", "", "", "", "", "", 0);
}

public:

enum enPermissions {
    eAll = -1, pListClients = 1, pAddNewClient = 2, pDeleteClient = 4,
    pUpdateClients = 8, pFindClient = 16, pTransactions = 32, pManageUsers = 64
};

struct stLoginRegisterRecord
{
    string DateTime;
    string UserName;
    string Password;
    int Permissions;
};

//Constructor
clsUser(enMode Mode,string FirstName,string LastName,string Email, string Phone,string UserName
, string Password,int Permissions):clsPerson(FirstName,LastName,Email,Phone) {

    this->_Mode = Mode;
    this->_UserName = UserName;
    this->_PassWord = Password;
}

```

```

        this->_Permissions = Permissions;
    }

//check Functions
bool IsEmpty() {
    return (_Mode == enMode::EmptyMode);
}

bool MarkedForDelete() {
    return _MarkedForDelete;
}

static bool IsUserExist(string UserName)
{
    clsUser User = clsUser::Find(UserName);
    return (!User.IsEmpty());
}

//setters
void set_UserName(string UserName) {
    this->_UserName = UserName;
}

void set_PassWord(string PassWord) {
    this->_PassWord = PassWord;
}

void set_Permissions(int Permissions) {
    this->_Permissions = Permissions;
}

//Getters
string get_UserName() { return _UserName; }
string get_PassWord() { return _PassWord; }
int get_Permissions() { return _Permissions; }

//declspec
_declspec(property(get = get_UserName, put = set_UserName))string UserName;
_declspec(property(get = get_PassWord, put = set_PassWord))string PassWord;
_declspec(property(get = get_Permissions, put = set_Permissions))int Permissions;

//public functions
static clsUser Find(string UserName) {
    fstream MyFile;
    MyFile.open("Users.txt", ios::in);

    if (MyFile.is_open()) {
        string Line = "";
        while (getline(MyFile,Line)) {
            clsUser User = _ConvertLineToUserObject(Line);
            if (User.UserName==UserName) {
                MyFile.close();
                return User;
            }
        }
        MyFile.close();
    }
    return _GetEmptyUserObject();
}

```

```

}

static clsUser Find(string UserName,string PassWord) {
    fstream MyFile;
    MyFile.open("Users.txt", ios::in);

    if (MyFile.is_open()) {
        string Line = "";
        while (getline(MyFile, Line)) {
            clsUser User = _ConvertLineToUserObject(Line);
            if (User.UserName == UserName&&User.PassWord==PassWord) {
                MyFile.close();
                return User;
            }
        }
        MyFile.close();
    }
    return _GetEmptyUserObject();
}

enum enSaveResults { svFaildEmptyObject = 0, svSucceeded = 1, svFaildUserExists = 2 };

enSaveResults Save() {
    switch (_Mode)
    {
    case clsUser::EmptyMode:
        if (IsEmpty())
            return svFaildEmptyObject;
        break;
    case clsUser::UpdateMode:
        _Update();
        return svSucceeded;
        break;
    case clsUser::AddMode:
        if (clsUser::IsUserExist(this->UserName)) {
            return enSaveResults::svFaildUserExists;
        }
        else {
            _AddNew();
            this->_Mode = UpdateMode;
            return svSucceeded;
        }
        break;
    default:
        break;
    }
}

bool Delete() {
    vector<clsUser>vUsers=_LoadUsersDataFromFile();

    for (clsUser& U : vUsers)
    {
        if (U.UserName == _UserName)
        {
            U._MarkedForDelete = true;
            break;
        }
    }
}

```

```

        }

        _SaveUsersDataToFile(vUsers);

        *this = _GetEmptyUserObject();

        return true;
    }

    static clsUser GetAddNewUserObject(string UserName)
    {
        return clsUser(enMode::AddMode, "", "", "", "", UserName, "", 0);
    }

    static vector <clsUser> GetUsersList()
    {
        return _LoadUsersDataFromFile();
    }

    bool CheckAccessPermission(enPermissions Permission)
    {
        if (this->Permissions == enPermissions::eAll)
            return true;

        if ((Permission & this->Permissions) == Permission)
            return true;
        else
            return false;
    }

    void RegisterLogIn()
    {

        string stDataLine = _PrepareLogInRecord();

        fstream MyFile;
        MyFile.open("LoginRegister.txt", ios::out | ios::app);

        if (MyFile.is_open())
        {

            MyFile << stDataLine << endl;

            MyFile.close();
        }
    }

    static vector <stLoginRegisterRecord> GetLoginRegisterList()
    {
        vector <stLoginRegisterRecord> vLoginRegisterRecord;

        fstream MyFile;
        MyFile.open("LoginRegister.txt", ios::in); //read Mode

        if (MyFile.is_open())
        {

            string Line;

```

```
    stLoginRegisterRecord LoginRegisterRecord;

    while (getline(MyFile, Line))
    {
        LoginRegisterRecord = _ConvertLoginRegisterLineToRecord(Line);

        vLoginRegisterRecord.push_back(LoginRegisterRecord);

    }

    MyFile.close();

}

return vLoginRegisterRecord;

}

};
```

## وڈے کوڈ ال register screen

```
#pragma once

#include <iostream>
#include "clsScreen.h"
#include <iomanip>
#include <fstream>
#include "clsUser.h"

class clsLoginRegisterScreen :protected clsScreen
{
private:
    static void PrintLoginRegisterRecordLine(clsUser::stLoginRegisterRecord LoginRegisterRecord)
    {
        cout << setw(8) << left << "" << "|" << setw(35) << left << LoginRegisterRecord.DateTime;
        cout << "|" << setw(20) << left << LoginRegisterRecord.UserName;
        cout << "|" << setw(20) << left << LoginRegisterRecord.Password;
        cout << "|" << setw(10) << left << LoginRegisterRecord.Permissions;
    }

public:
    static void ShowLoginRegisterScreen()
    {
        vector <clsUser::stLoginRegisterRecord> vLoginRegisterRecord = clsUser::GetLoginRegisterList();

        string Title = "\tLogin Register List Screen";
        string SubTitle = "\t\t(" + to_string(vLoginRegisterRecord.size()) + ") Record(s).";

        _DrawScreenHeader(Title, SubTitle);

        cout << setw(8) << left << "" << "\n\t_____";
        cout << "\n" << endl;
    }
}
```

```

cout << setw(8) << left << "" << "|" " << left << setw(35) << "Date/Time";
cout << "|" " << left << setw(20) << "UserName";
cout << "|" " << left << setw(20) << "Password";
cout << "|" " << left << setw(10) << "Permissions";
cout << setw(8) << left << "" << "\n\t____"; 
cout << "_____\n" << endl;

if (vLoginRegisterRecord.size() == 0)
    cout << "\t\t\t\tNo Logins Available In the System!";
else

    for (clsUser::stLoginRegisterRecord Record : vLoginRegisterRecord)
    {

        PrintLoginRegisterRecordLine(Record);
        cout << endl;
    }

    cout << setw(8) << left << "" << "\n\t____"; 
    cout << "_____\n" << endl;

}
};


```

## وده کود ال main menu screen

```

#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsInputValidate.h"
#include <iomanip>
#include "clsClientListScreen.h"
#include "clsAddNewClientScreen.h"
#include "clsDeleteClientScreen.h"
#include "clsTransactionsScreen.h"
#include "clsFindClientScreen.h"
#include "clsUpdateClientScreen.h"
#include "clsManageUsersScreen.h"
#include "Global.h"
#include "clsLoginRegisterScreen.h"

using namespace std;

class clsMainScreen :protected clsScreen
{

private:
    enum enMainMenueOptions {
        eListClients = 1, eAddNewClient = 2, eDeleteClient = 3,
        eUpdateClient = 4, eFindClient = 5, eShowTransactionsMenue = 6,
        eManageUsers = 7, eLoginRegister=8,eExit = 9
    };

    static short _ReadMainMenueOption()
    {
        cout << setw(37) << left << "" << "Choose what do you want to do? [1 to 9]? ";
        short Choice = clsInputValidate::ReadShortNumberBetween(1, 9, "Enter Number between 1 to 9? ");
        return Choice;
    }
};


```

```
}

static void _GoBackToMainMenue()
{
    cout << setw(37) << left << "" << "\n\tPress any key to go back to Main Menue...\n";

    system("pause>0");
    ShowMainMenue();
}

static void _ShowAllClientsScreen()
{
    // cout << "\nClient List Screen Will be here...\n";
    clsClientListScreen::ShowClientsList();

}

static void _ShowAddNewClientsScreen()
{
    // cout << "\nAdd New Client Screen Will be here...\n";
    clsAddNewClientScreen::ShowAddNewClientScreen();

}

static void _ShowDeleteClientScreen()
{
    //cout << "\nDelete Client Screen Will be here...\n";
    clsDeleteClientScreen::ShowDeleteClientScreen();

}

static void _ShowUpdateClientScreen()
{
    //cout << "\nUpdate Client Screen Will be here...\n";
    clsUpdateClientScreen::ShowUpdateClientScreen();

}

static void _ShowFindClientScreen()
{
    // cout << "\nFind Client Screen Will be here...\n";
    clsFindClientScreen::ShowFindClientScreen();

}

static void _ShowTransactionsMenue()
{
    // cout << "\nTransactions Menue Will be here...\n";
    clsTransactionsScreen::ShowTransactionsMenue();

}

static void _ShowManageUsersMenue()
{
    // cout << "\nUsers Menue Will be here...\n";
    clsManageUsersScreen::ShowManageUsersMenue();

}
```

```

static void _ShowLoginRegisterScreen() {
    // cout << "\nLogin Register Menue Will be here...\n";

    clsLoginRegisterScreen::ShowLoginRegisterScreen();
}

/* static void _ShowEndScreen()
{
    cout << "\nEnd Screen Will be here...\n";

} */

static void _Logout()
{

    CurrentUser = clsUser::Find("", "");
    //then it will go back to main function.
}

static void _PerfromMainMenueOption(enMainMenueOptions MainMenueOption)
{
    switch (MainMenueOption)
    {
        case enMainMenueOptions::eListClients:
        {
            system("cls");
            _ShowAllClientsScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eAddNewClient:
        {
            system("cls");
            _ShowAddNewClientsScreen();
            _GoBackToMainMenue();
            break;
        }

        case enMainMenueOptions::eDeleteClient:
        {
            system("cls");
            _ShowDeleteClientScreen();
            _GoBackToMainMenue();
            break;
        }

        case enMainMenueOptions::eUpdateClient:
        {
            system("cls");
            _ShowUpdateClientScreen();
            _GoBackToMainMenue();
            break;
        }

        case enMainMenueOptions::eFindClient:
        {
            system("cls");
            _ShowFindClientScreen();
            _GoBackToMainMenue();
            break;
        }

        case enMainMenueOptions::eShowTransactionsMenue:
        {
            system("cls");
            _ShowTransactionsMenue();
            _GoBackToMainMenue();
        }
    }
}

```

```

break;

case enMainMenueOptions::eManageUsers:
    system("cls");
    _ShowManageUsersMenue();
    _GoBackToMainMenue();
    break;

case enMainMenueOptions::eLoginRegister:
    system("cls");
    _ShowLoginRegisterScreen();
    _GoBackToMainMenue();
    break;

case enMainMenueOptions::eExit:
    system("cls");
    _Logout();
    //Login();

    break;
}

}

public:

static void ShowMainMenue()
{
    system("cls");
    _DrawScreenHeader("\t\tMain Screen","");
    cout << setw(37) << left << "=====\\n";
    cout << setw(37) << left << "  Main Menue\\n";
    cout << setw(37) << left << "=====\\n";
    cout << setw(37) << left << "\t[1] Show Client List.\\n";
    cout << setw(37) << left << "\t[2] Add New Client.\\n";
    cout << setw(37) << left << "\t[3] Delete Client.\\n";
    cout << setw(37) << left << "\t[4] Update Client Info.\\n";
    cout << setw(37) << left << "\t[5] Find Client.\\n";
    cout << setw(37) << left << "\t[6] Transactions.\\n";
    cout << setw(37) << left << "\t[7] Manage Users.\\n";
    cout << setw(37) << left << "\t[8] Login Register.\\n";
    cout << setw(37) << left << "\t[9] Logout.\\n";
    cout << setw(37) << left << "=====\\n";

    _PerfromMainMenueOption((enMainMenueOptions)_ReadMainMenueOption());
}
};

}

```

## Permission to Show Log Register Screen

عاوزين نضيف صلاحيه للوصول لشاشة ال login register

## Solution

ده التعديل على كلاس ال user

```

enum enPermissions {
    eAll = -1, pListClients = 1, pAddNewClient = 2, pDeleteClient = 4,
    pUpdateClients = 8, pFindClient = 16, pTransactions = 32, pManageUsers = 64
    ,pLoginRegister=128
};

```

## وده التعديل على شاشه ال add user وال update user

```

static int _ReadPermissionsToSet()
{
    int Permissions = 0;
    char Answer = 'n';

    cout << "\nDo you want to give full access? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        return -1;
    }

    cout << "\nDo you want to give access to : \n ";

    cout << "\nShow Client List? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {

        Permissions += clsUser::enPermissions::pListClients;
    }

    cout << "\nAdd New Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pAddNewClient;
    }

    cout << "\nDelete Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pDeleteClient;
    }

    cout << "\nUpdate Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pUpdateClients;
    }

    cout << "\nFind Client? y/n? ";
    cin >> Answer;
    if (Answer == 'y' || Answer == 'Y')
    {
        Permissions += clsUser::enPermissions::pFindClient;
    }
}

```

```

cout << "\nTransactions? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pTransactions;
}

cout << "\nManage Users? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pManageUsers;
}

cout << "\nShow Login Register? y/n? ";
cin >> Answer;
if (Answer == 'y' || Answer == 'Y')
{
    Permissions += clsUser::enPermissions::pLoginRegister;
}

return Permissions;
}

```

وده التعديل على شاشة ال login register

```

static void ShowLoginRegisterScreen()
{
    if (!CheckAccessRights(clsUser::enPermissions::pLoginRegister))
    {
        return;// this will exit the function and it will not continue
    }
}

```

### Transfer Screen

عاوزين نعمل في قائمة ال transactions خاصية اننا نحول من حساب عيل لحساب عميل تاني

## Transactions Screen

User: User2

Date: 25/12/2022

### Transactions Menue

- [1] Deposit.
- [2] Withdraw.
- [3] Total Balances.
- [4] Transfer.
- [5] Main Menue.

Choose what do you want to do? [1 to 5]? 

## Transfer Screen

User: User2

Date: 25/12/2022

Please Enter Account Number to Transfer From: A101

Client Card:

Full Name : Mohammed Abu-Hadhoud

Acc. Number : A101

Balance : 5000

Please Enter Account Number to Transfer From:

Please Enter Account Number to Transfer From: A108

Client Card:

---

Full Name : Gandi Omran  
Acc. Number : A108  
Balance : 8000

---

Enter Transfer Amount?

Enter Transfer Amount? 10000

Amount Exceeds the available Balance, Enter another Amount ? Copyright 2022

Amount Exceeds the available Balance, Enter another Amount ? 1000

Are you sure you want to perform this operation? y/n? y

Transfer done successfully

Client Card:

---

Full Name : Mohammed Abu-Hadhoud  
Acc. Number : A101  
Balance : 4000

---

Client Card:

---

Full Name : Gandi Omran  
Acc. Number : A108  
Balance : 9000

---

Press any key to go back to Transaction's Menue... 

Copyright 2022

## Solution

ده التعديل على ال **clsBankClient**

```
bool Transfer(float Amount, clsBankClient& DestinationClient)
{
    if (Amount > AccountBalance)
    {
        return false;
    }

    Withdraw(Amount);
    DestinationClient.Deposit(Amount);
    return true;
}
```

ده كود شاشة ال **transfer**

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsPerson.h"
#include "clsBankClient.h"
#include "clsInputValidate.h"

class clsTransferScreen :protected clsScreen
{

private:
    static void _PrintClient(clsBankClient Client)
    {
        cout << "\nClient Card:"; 
        cout << "\n_____ \n";
        cout << "\nFull Name : " << Client.get_FullName();
        cout << "\nAcc. Number : " << Client.get_AccountNumber();
        cout << "\nBalance : " << Client.AccountBalance;
        cout << "\n_____ \n";

    }

    static string _ReadAccountNumber()
    {
        string AccountNumber;
        cout << "\nPlease Enter Account Number to Transfer From: ";
        AccountNumber = clsInputValidate::ReadString();
        while (!clsBankClient::IsClientExist(AccountNumber))
        {
            cout << "\nAccount number is not found, choose another one: ";
            AccountNumber = clsInputValidate::ReadString();
        }
        return AccountNumber;
    }

    static float ReadAmount(clsBankClient SourceClient)
    {
        float Amount;

        cout << "\nEnter Transfer Amount? ";

        Amount = clsInputValidate::ReadFloatNumber();

        while (Amount > SourceClient.AccountBalance)
        {
```

```

cout << "\nAmount Exceeds the available Balance, Enter another Amount ? ";
Amount = clsInputValidate::ReadDbiNumber();
}

return Amount;
}

public:

static void ShowTransferScreen()
{

_DrawScreenHeader("\tTransfer Screen","");
clsBankClient SourceClient = clsBankClient::Find(_ReadAccountNumber());

_PrintClient(SourceClient);

clsBankClient DestinationClient = clsBankClient::Find(_ReadAccountNumber());

_PrintClient(DestinationClient);

float Amount = ReadAmount(SourceClient);

cout << "\nAre you sure you want to perform this operation? y/n? ";
char Answer = 'n';
cin >> Answer;
if (Answer == 'Y' || Answer == 'y')
{
    if (SourceClient.Transfer(Amount, DestinationClient))
    {
        cout << "\nTransfer done successfully\n";
    }
    else
    {
        cout << "\nTransfer Faild \n";
    }
}

_PrintClient(SourceClient);
_PrintClient(DestinationClient);

}

};

}

```

وهي التعديلات على كود قائمة ال transactions

```

enum enTransactionsMenueOptions {
    eDeposit = 1, eWithdraw = 2,
    eShowTotalBalance = 3,eTransfer=4 ,eShowMainMenue = 5
};

static void _ShowTransferScreen()
{
    clsTransferScreen::ShowTransferScreen();
}

case enTransactionsMenueOptions::eTransfer:
{
    system("cls");
    _ShowTransferScreen();
    _GoBackToTransactionsMenue();
}

```

```

        break;
    }

static void ShowTransactionsMenue()
{
    if (!CheckAccessRights(clsUser::enPermissions::pTranactions))
    {
        return;// this will exit the function and it will not continue
    }

    system("cls");
    _DrawScreenHeader("\t Transactions Screen","");
}

cout << setw(37) << left << "=====\n";
cout << setw(37) << left << "\t\t Transactions Menue\n";
cout << setw(37) << left << "=====\n";
cout << setw(37) << left << "\t[1] Deposit.\n";
cout << setw(37) << left << "\t[2] Withdraw.\n";
cout << setw(37) << left << "\t[3] Total Balances.\n";
cout << setw(37) << left << "\t[4] Transfer.\n";
cout << setw(37) << left << "\t[5] Main Menue.\n";
cout << setw(37) << left << "=====\n";

_PerformTransactionsMenueOption((enTransactionsMenueOptions)ReadTransactionsMenueOption());
}

```

## Create Transfer Log

عايزين نخزن عمليات التحويل بين الحسابات في ملف



```

TransferLog - Notepad
File Edit Format View Help
25/12/2022 - 14:42:21##A101##A108##400.00000##3600.00000##9400.00000##User2
25/12/2022 - 14:47:52##A108##A109##1000.00000##8400.00000##6000.00000##User2

```

البيانات بالترتيب (التاريخ والوقت – المحول منه-المحول اليه – قيمة التحويل – رصيد المحول منه  
بعد التحويل – رصيد المحول اليه بعد التحويل- اليوزر اللي عمل التحويل)

## Solution

ده التعديل على ال `clsBankClient`

```

string _PrepareTransferLogRecord(float Amount, clsBankClient DestinationClient,
                                 string UserName, string Seperator = "###")
{
    string TransferLogRecord = "";
    TransferLogRecord += clsDate::GetSystemDateTimeString() + Seperator;
    TransferLogRecord += get_AccountNumber() + Seperator;
    TransferLogRecord += DestinationClient.get_AccountNumber() + Seperator;
    TransferLogRecord += to_string(Amount) + Seperator;
    TransferLogRecord += to_string(AccountBalance) + Seperator;
    TransferLogRecord += to_string(DestinationClient.AccountBalance) + Seperator;
    TransferLogRecord += UserName;
    return TransferLogRecord;
}

void _RegisterTransferLog(float Amount, clsBankClient DestinationClient, string UserName)
{
    string stDataLine = _PrepareTransferLogRecord(Amount, DestinationClient, UserName);
}

```

```

fstream MyFile;
MyFile.open("TransferLog.txt", ios::out | ios::app);

if (MyFile.is_open())
{
    MyFile << stDataLine << endl;

    MyFile.close();
}

}

bool Transfer(float Amount, clsBankClient& DestinationClient, string UserName)
{
    if (Amount > AccountBalance)
    {
        return false;
    }

    Withdraw(Amount);
    DestinationClient.Deposit(Amount);
    _RegisterTransferLog(Amount, DestinationClient, UserName);

    return true;
}

```

وذه التعديل على شاشة ال transfer

```
if (SourceClient.Transfer(Amount, DestinationClient, CurrentUser.UserName))
```

**Show Transfers Log Screen**

## Transactions Screen

User: User2

Date: 25/12/2022

### Transactions Menue

- [1] Deposit.
- [2] Withdraw.
- [3] Total Balances.
- [4] Transfer.
- [5] Transfer Log.**
- [6] Main Menue.

Choose what do you want to do? [1 to 6]?

Transfer Log List Screen  
(2) Record(s).

User: User2  
Date: 25/12/2022



Date/Time	s.Acct	d.Acct	Amount	s.Balance	d.Balance	User
25/12/2022 - 14:42:21	A101	A108	400	3600	9400	User2
25/12/2022 - 14:47:52	A108	A109	1000	8400	6000	User2

Press any key to go back to Transactions Menue...

## Solution

د. التعديل على bank client

```
static stTrnsferLogRecord _ConvertTransferLogLineToRecord(string Line, string Seperator = "#/#")  
{
```

```

stTrnsferLogRecord TrnsferLogRecord;

vector <string> vTrnsferLogRecordLine = clsString::Split(Line, Seperator);
TrnsferLogRecord.DateTime = vTrnsferLogRecordLine[0];
TrnsferLogRecord.SourceAccountNumber = vTrnsferLogRecordLine[1];
TrnsferLogRecord.DestinationAccountNumber = vTrnsferLogRecordLine[2];
TrnsferLogRecord.Amount = stod(vTrnsferLogRecordLine[3]);
TrnsferLogRecord.srcBalanceAfter = stod(vTrnsferLogRecordLine[4]);
TrnsferLogRecord.destBalanceAfter = stod(vTrnsferLogRecordLine[5]);
TrnsferLogRecord.UserName = vTrnsferLogRecordLine[6];

return TrnsferLogRecord;
}

public :
struct stTrnsferLogRecord
{
    string DateTime;
    string SourceAccountNumber;
    string DestinationAccountNumber;
    float Amount;
    float srcBalanceAfter;
    float destBalanceAfter;
    string UserName;
};

static vector <stTrnsferLogRecord> GetTransfersLogList()
{
    vector <stTrnsferLogRecord> vTransferLogRecord;

    fstream MyFile;
    MyFile.open("TransfersLog.txt", ios::in); //read Mode

    if (MyFile.is_open())
    {

        string Line;

        stTrnsferLogRecord TransferRecord;

        while (getline(MyFile, Line))
        {

            TransferRecord = _ConvertTransferLogLineToRecord(Line);

            vTransferLogRecord.push_back(TransferRecord);
        }
    }

    MyFile.close();
}

return vTransferLogRecord;
}

```

التعديل على قائمة ال transactions

```

enum enTransactionsMenueOptions {
    eDeposit = 1, eWithdraw = 2,
}

```

```

eShowTotalBalance = 3, eTransfer = 4, eTransferLog = 5,
eShowMainMenue = 6

};

static short ReadTransactionsMenueOption()
{
    cout << setw(37) << left << "" << "Choose what do you want to do? [1 to 6]? ";
    short Choice = clsInputValidate::ReadShortNumberBetween(1, 6, "Enter Number between 1 to 6? ");
    return Choice;
}

static void _ShowTransferLogScreen()
{
    //cout << "\n Transfer Screen will be here.\n";
    clsTransferLogScreen::ShowTransferLogScreen();
}

case enTransactionsMenueOptions::eTransferLog:
{
    system("cls");
    _ShowTransferLogScreen();
    _GoBackToTransactionsMenue();
    break;
}

static void ShowTransactionsMenue()
{
    if (!CheckAccessRights(clsUser::enPermissions::pTranactions))
    {
        return;// this will exit the function and it will not continue
    }

    system("cls");
    _DrawScreenHeader("\tTransactions Screen", "");

    cout << setw(37) << left << "" << "=====\\n";
    cout << setw(37) << left << "" << "\\t\\t Transactions Menue\\n";
    cout << setw(37) << left << "" << "=====\\n";
    cout << setw(37) << left << "" << "\\t[1] Deposit.\\n";
    cout << setw(37) << left << "" << "\\t[2] Withdraw.\\n";
    cout << setw(37) << left << "" << "\\t[3] Total Balances.\\n";
    cout << setw(37) << left << "" << "\\t[4] Transfer.\\n";
    cout << setw(37) << left << "" << "\\t[5] Transfer Log.\\n";
    cout << setw(37) << left << "" << "\\t[6] Main Menue.\\n";
    cout << setw(37) << left << "" << "=====\\n";

    _PerformTransactionsMenueOption((enTransactionsMenueOptions)ReadTransactionsMenueOption());
}

```

ده کود شاشة ال transfer log

```

#pragma once

#include <iostream>
#include "clsScreen.h"
#include <iomanip>
#include <fstream>
#include "clsBankClient.h"

class clsTransferLogScreen :protected clsScreen

```

```

{

private:

static void PrintTransferLogRecordLine(clsBankClient::stTrnsferLogRecord TransferLogRecord)
{
    cout << setw(8) << left << "" << "|" << setw(23) << left << TransferLogRecord.DateTime;
    cout << "|" << setw(8) << left << TransferLogRecord.SourceAccountNumber;
    cout << "|" << setw(8) << left << TransferLogRecord.DestinationAccountNumber;
    cout << "|" << setw(8) << left << TransferLogRecord.Amount;
    cout << "|" << setw(10) << left << TransferLogRecord.srcBalanceAfter;
    cout << "|" << setw(10) << left << TransferLogRecord.destBalanceAfter;
    cout << "|" << setw(8) << left << TransferLogRecord.UserName;

}

public:

static void ShowTransferLogScreen()
{
    vector <clsBankClient::stTrnsferLogRecord> vTransferLogRecord = clsBankClient::GetTransfersLogList();

    string Title = "\tTransfer Log List Screen";
    string SubTitle = "\t (" + to_string(vTransferLogRecord.size()) + ") Record(s)";

    _DrawScreenHeader(Title, SubTitle);

    cout << setw(8) << left << "" << "\n\t_____"; 
    cout << " _____\n" << endl;

    cout << setw(8) << left << "" << "|" << left << setw(23) << "Date/Time";
    cout << "|" << left << setw(8) << "s.Acct";
    cout << "|" << left << setw(8) << "d.Acct";
    cout << "|" << left << setw(8) << "Amount";
    cout << "|" << left << setw(10) << "s.Balance";
    cout << "|" << left << setw(10) << "d.Balance";
    cout << "|" << left << setw(8) << "User";

    cout << setw(8) << left << "" << "\n\t_____"; 
    cout << " _____\n" << endl;

    if (vTransferLogRecord.size() == 0)
        cout << "\t\t\tNo Transfers Available In the System!";
    else

        for (clsBankClient::stTrnsferLogRecord Record : vTransferLogRecord)
        {

            PrintTransferLogRecordLine(Record);
            cout << endl;
        }

        cout << setw(8) << left << "" << "\n\t_____"; 
        cout << " _____\n" << endl;
}

}

```

```
};
```

## Encrypt Password In File

عاوز يشفر الباسورد اللي في ملف ال user.txt

```
*Users - Notepad
File Edit Format View Help
Hilal#//#Ali#/##Hilal@Gmail.com#/##83983948#/##User1#/##1234#/##0
Jamil#//#Adli#/##Jamil@gmail.com#/##23123123#/##User2#/##1234#/##-1
Lina#/##Loay#/##Lina@gmail.com#/##1234#/##User3#/##1234#/##-1
Mazem#/##Kareem#/##Mazin@gmail.com#/##898234#/##User4#/##1234#/##-1
Alia#/##Ali#/##M@gmail.com#/##09388322#/##User6#/##1234#/##-1
Ola#/##Ahmad#/##Ola@gmail.com#/##838837#/##User8#/##1234#/##7
```

لو الباسورد مثلا 1234 انت تخزنه علي انه مثلا 4567 بس جوه السيستم تتعامل معاه علي انه 1234 وتعمل الحوار ده في كل ملف محفوظ فيه الباسورد

### Solution

كل اللي هتعمله انك هتتحجي عند أي FUNCTION بتخزن باسورد وتعمله ENCRYPT وتعمله DECRYPT بتقرا الباسورد من الملف وتعمله وتحجي عند أي FUNCTION

ده التعديل علي كلاس ال UTIL

```
static string EncryptText(string Text, short EncryptionKey = 2)
{
    for (int i = 0; i <= Text.length(); i++)
    {
        Text[i] = char((int)Text[i] + EncryptionKey);
    }
    return Text;
}

static string DecryptText(string Text, short EncryptionKey = 2)
{
    for (int i = 0; i <= Text.length(); i++)
    {
        Text[i] = char((int)Text[i] - EncryptionKey);
    }
    return Text;
}
```

وده التعديل علي كلاس ال USER

```
//private functions
static stLoginRegisterRecord _ConvertLoginRegisterLineToRecord(string Line, string Seperator = "#//#")
{
    stLoginRegisterRecord LoginRegisterRecord;
```

```

vector <string> LoginRegisterDataLine = clsString::Split(Line, Seperator);
LoginRegisterRecord.DateTime = LoginRegisterDataLine[0];
LoginRegisterRecord.UserName = LoginRegisterDataLine[1];
LoginRegisterRecord.Password = clsUtil::DecryptText(LoginRegisterDataLine[2]);
LoginRegisterRecord.Permissions = stoi(LoginRegisterDataLine[3]);

return LoginRegisterRecord;
}

string _PrepareLogInRecord(string Seperator = "#//#")
{
    string LoginRecord = "";
    LoginRecord += clsDate::GetSystemDateTimeString() + Seperator;
    LoginRecord += UserName + Seperator;
    //here we encrypt store the encrypted Password not the real one.
    LoginRecord += clsUtil::EncryptText(PassWord) + Seperator;
    LoginRecord += to_string(Permissions);
    return LoginRecord;
}

static clsUser _ConvertLineToUserObject(string Line, string Seperator = "#//#") {
    vector<string> vUserData;
    vUserData = clsString::Split(Line, Seperator);

    return clsUser(enMode)::UpdateMode, vUserData[0], vUserData[1], vUserData[2],
        vUserData[3], vUserData[4], clsUtil::DecryptText(vUserData[5]), stoi(vUserData[6]));
}

static string _ConvertFromUserObjectToLine(clsUser User, string Seperator = "#//#") {

    string UserRecord = "";
    UserRecord += User.FirstName + Seperator;
    UserRecord += User.LastName + Seperator;
    UserRecord += User.Email + Seperator;
    UserRecord += User.Phone + Seperator;
    UserRecord += User.UserName + Seperator;
    //here we encrypt store the encrypted Password not the real one.
    UserRecord += clsUtil::EncryptText(User.PassWord) + Seperator;
    UserRecord += to_string(User.Permissions);

    return UserRecord;
}

```

### Abstract Class/Interface Practical Example

ال developer abstract class interface هوا عقد بين أستاذ محمد أبو هدهود وبينكك و لازم تلبي كل الشروط الموجودة فيه لو ماعملتش الشروط دي ال compiler مش هي عمل للكود compile

لو جيت قولتلك ان كلاس ال person يكون عندي send sms و send emails و send fax وال الثقه بيني وبينك موجوده بس ممكن تغلط في انك ماتكتبشي الاسامي بالطريقه اللي انا عايزها او تعمل مش من النوع اللي انا عايزه او تديها اسمي غلط parameters

```
#pragma once
#include <iostream>
#include <string>
using namespace std;

class InterfaceCommunication
{
public:
    virtual void SendEmail(string Title, string Body) = 0;
    virtual void SendFax(string Title, string Body) = 0;
    virtual void SendSMS(string Title, string Body) = 0;
};

};
```

أي حد هيرث منه لازم يعمل الـ functions اللي موجوده فيه زي كده

```
#pragma once
#include <iostream>
#include "InterfaceCommunication.h"
using namespace std;

class clsPerson : public InterfaceCommunication
{
private:
    string _FirstName;
    string _LastName;
    string _Email;
    string _Phone;

public:
    //constructors
    clsPerson(string FirstName, string LastName, string Email, string Phone) {
        this->_FirstName = FirstName;
        this->_LastName = LastName;
        this->_Email = Email;
        this->_Phone = Phone;
    }

    //setters
    void set_FirstName (string FirstName) { this->_FirstName = FirstName; }
    void set_LastName (string LastName) { this->_LastName = LastName; }
    void set_Email(string Email) { this->_Email = Email; }
    void set_Phone(string Phone) { this->_Phone = Phone; }

    //getters
    string get_FirstName() { return this->_FirstName; }
    string get_LastName() { return this->_LastName; }
    string get_Email() { return this->_Email; }
    string get_Phone() { return this->_Phone; }
    string get_FullName() { return this->_FirstName + " " + this->_LastName; }

    //declspec
    __declspec(property(get = get_FirstName, put = set_FirstName)) string FirstName;
    __declspec(property(get = get_LastName, put = set_LastName)) string LastName;
    __declspec(property(get = get_Email, put = set_Email)) string Email;
    __declspec(property(get = get_Phone, put = set_Phone)) string Phone;

    //person functions
};
```

```

void Print()
{
    cout << "\nInfo:";

    cout << "\n_____";
    cout << "\nFirstName: " << this->_FirstName;
    cout << "\nLastName : " << this->_LastName;
    cout << "\nFull Name: " << this->get_FullName();
    cout << "\nEmail : " << this->_Email;
    cout << "\nPhone : " << this->_Phone;
    cout << "\n_____\\n";
}

//////////implementation of interface

void SendEmail(string Title, string Body)
{

}

void SendFax(string Title, string Body)
{

}

void SendSMS(string Title, string Body)
{

};


```

## Project Overview

هعمل مشروع صغير عباره عن currency exchange بس هيكون من ضمن مشروع البنك  
هبيك رؤية عنه بس ماتطبقش غير من الدرس اللي بعده

## Main Screen

User: User2  
Date: 26/12/2022

### Main Menue

- [1] Show Client List.
- [2] Add New Client.
- [3] Delete Client.
- [4] Update Client Info.
- [5] Find Client.
- [6] Transactions.
- [7] Manage Users.
- [8] Login Register.
- [9] Currency Exchange.
- [10] Logout.

Choose what do you want to do? [1 to 10]?

## Currency Exchange Main Screen



User: User2

Date: 26/12/2022

### Currency Exchange Menue

- [1] List Currencies.
- [2] Find Currency.
- [3] Update Rate.
- [4] Currency Calculator.
- [5] Main Menue.

Choose what do you want to do? [1 to 5]?

Currencies List Screen  
(222) Currency.

User: User2  
Date: 26/12/2022



Course #11  
OOP as it Should Be  
(Applications)

Country	Code	Name	Rate/(1\$)
United States of America	USD	US Dollar	1
Afghanistan	AFN	Afghanistan Afghani	87.48
Albania	ALL	Albania Lek(e)	109.01
Algeria	DZD	Algerian Dinar	137.046
American Samoa	USD	US Dollar	1
France	EUR	Euro	0.9
Angola	AOA	Angolan Kwanza	504.734
Anguilla	XCD	E.C. Dollar	
Antigua and Barbuda	XCD	E.C. Dollar	
Argentina	ARS	Argentine Peso	
Armenia	AMD	Armenian Dram	
Aruba	AWG	Aruban Guilder	
Australia	AUD	Australian Dollar	
Austria	EUR	Euro	0.958
Azerbaijan	AZN	Azerbaijan Manat	1.694



---

### Find Currency Screen

---

User: User2  
Date: 26/12/2022

Find By: [1] Code or [2] Country ? 1

Please Enter CurrencyCode: jjj

Currency Was not Found :-(

Press any key to go back to Currencies Menue...

---

### Find Currency Screen

---

User: User2  
Date: 26/12/2022

Find By: [1] Code or [2] Country ? 1

Please Enter CurrencyCode: jod

Currency Found : - )

Currency Card:

---

Country : Jordan  
Code : JOD  
Name : Jordanian Dinar  
Rate(1\$) = : 0.708

---

Press any key to go back to Currencies Menue...

---

Find Currency Screen

---

User: User2  
Date: 26/12/2022

Find By: [1] Code or [2] Country ? 2

Please Enter Country Name: egypt

Currency Found :-)

Currency Card:

---

Country : Egypt  
Code : EGP  
Name : Egyptian Pound  
Rate(1\$) = : 24.592

---

Press any key to go back to Currencies Menue...

## Update Currency Screen

User: User2  
Date: 26/12/2022

Please Enter Currency Code: jod

Currency Card:

Country : Jordan  
Code : JOD  
Name : Jordanian Dinar  
Rate(1\$) = : 0.708



Are you sure you want to update the rate of this Currency y/n? y

Update Currency Rate:

Enter New Rate: .

Are you sure you want to update the rate of this Currency y/n? y

Update Currency Rate:

Enter New Rate: .9

Currency Rate Updated Successfully :-)

Currency Card:

Country : Jordan  
Code : JOD  
Name : Jordanian Dinar  
Rate(1\$) = : 0.9

Press any key to go back to Currencies Menue...■

Copyright 2022

## Update Currency Screen

User: User2  
Date: 26/12/2022

Please Enter Currency1 Code:

jod

Please Enter Currency2 Code:

usd

Enter Amount to Exchange: 100

Convert From:

---

Country : Jordan  
Code : JOD  
Name : Jordanian Dinar  
Rate(1\$) = : 0.9

---

100 JOD = 111.111 USD

Do you want to perform another calculation? y/n ?

Copyright 2022

C:\Users\USER\source\repos\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe

Enter Amount to Exchange: 100

Convert From:

---

Country	:	Jordan
Code	:	JOD
Name	:	Jordanian Dinar
Rate(1\$) =	:	0.9

---

100 JOD = 111.111 USD

Converting from USD to:

To:

---

Country	:	Egypt
Code	:	EGP
Name	:	Egyptian Pound
Rate(1\$) =	:	24.592

---

100 JOD = 2732.44 EGP

Do you want to perform another calculation? y/n ? n

Do you want to perform another calculation? y/n ? n

### **Prepare Currency Object**

هعمل كلاس ال currency  
ده شكل ملف ال text اللي فيه الداتا بتاعت اسعار العملات

## Currencies - Notepad

File Edit Format View Help

United States of America##USD##US Dollar##1.000000  
Afghanistan##AFN##Afghanistan Afghani##87.480003  
Albania##ALL##Albania Lek(e)##109.010002  
Algeria##DZD##Algerian Dinar##137.046005  
American Samoa##USD##US Dollar##1.000000  
France##EUR##Euro##0.900000  
Angola##AOA##Angolan Kwanza##504.734009  
Anguilla##XCD##E.C. Dollar##2.700000  
Antigua and Barbuda##XCD##E.C. Dollar##2.700000  
Argentina##ARS##Argentine Peso##171.744995  
Armenia##AMD##Armenian Dram##391.700012  
Aruba##AWG##Aruban Guilder##1.790000  
Australia##AUD##Australian Dollar##1.451000  
Austria##EUR##Euro##0.938000  
Azerbaijan##AZN##Azerbaijan Manat##1.694000  
Bahamas##BSD##Bahamian Dollar##1.000000  
Bahrain##BHD##Bahraini Dinar##0.377000  
Bangladesh##BDT##Bangladesh Taka##101.000000  
Barbados##BBD##Barbados Dollar##2.000000  
Belarus##BYN##Belarusian Ruble##2.478000  
Belgium##EUR##Euro##0.938000  
Belize##BZD##Belize Dollar##2.000000  
Benin##XOF##CFA Franc##615.401978  
Bermuda##BMD##Bermuda Dollar##1.000000  
Bhutan##BTN##Bhutan Ngultrum##82.800003  
Bolivia##BOB##Bolivia Boliviano##6.852000  
Bosnia and Herzegovina##BAM##Bosnia and Herzegovina Convertible Mark##1.835000  
Botswana##BWP##Botswana Pula##12.953000  
Brazil##BRL##Brazilian Real##5.261000  
Brunei##BND##Brunei Dollar##1.344000  
Bulgaria##BGN##Bulgarian Lev##1.835000  
Burkina Faso##XOF##CFA Franc##615.401978  
Burundi##BIF##Burundi Franc##2043.123047  
Cambodia##KHR##Cambodian Riel##4130.000000  
Cameroon##XAF##CFA Franc##615.401978

الملف عباره عن اسم الدوله وکود العمله واسم العمله وسعر التحويل  
كل المطلوب مني هنا اني اعدل ال rate مش عايز اعدل حاجه تانيه  
ده کلاس ال currency

```
#pragma once
#include<iostream>
#include<fstream>
#include <vector>
#include"clsString.h"
#include <string>

using namespace std;

class clsCurrency
{
private:
    enum enMode { EmptyMode=0,UpdateMode=1};
```

```

//private variables
enMode _Mode;
string _Country;
string _CurrencyCode;
string _CrruncyName;
float _Rate;

//private functions
static clsCurrency _ConvertLineToCurrencyObject(string Line, string Seperator = "#/#") {
    vector<string> vCurrencyData=clsString::Split(Line,Seperator);
    return
clsCurrency(enMode::UpdateMode,vCurrencyData[0],vCurrencyData[1],vCurrencyData[2],stof(vCurrencyData[3]));

}

static string _ConvertCurrencyObjectToLine(clsCurrency Currency,string Seperator="#/#") {
    return Currency.get_Country() + Seperator
        + Currency.get_CurrencyCode() + Seperator
        + Currency.get_CurrencyName() + Seperator
        + to_string(Currency.get_Rate());
}

static vector<clsCurrency> _LoadCurrencysDataFromFile() {
    vector<clsCurrency> vCurrencys;

    fstream MyFile;
    MyFile.open("Currencies.txt", ios::in);

    if (MyFile.is_open()) {
        string Line;

        while (getline(MyFile,Line)) {
            vCurrencys.push_back(_ConvertLineToCurrencyObject(Line));
        }
        MyFile.close();
    }
    return vCurrencys;
}

static void _SaveCurrencyDataToFile(vector<clsCurrency> vCurrencys) {
    fstream MyFile;
    MyFile.open("Currencies.txt",ios::out);

    if (MyFile.is_open()) {
        for (clsCurrency Currency:vCurrencys) {
            MyFile << _ConvertCurrencyObjectToLine(Currency) << endl;
        }
        MyFile.close();
    }
}

void _Update() {
    vector<clsCurrency>vCurrencys = _LoadCurrencysDataFromFile();
    for (clsCurrency &Currency:vCurrencys) {
        if (Currency.get_CurrencyCode() == this->_CurrencyCode) {
            Currency = *this;
        }
    }
    _SaveCurrencyDataToFile(vCurrencys);
}

```

```

    }

    static clsCurrency getEmptyCurrencyObject() {
        return clsCurrency(enMode::EmptyMode,"","","","",0);
    }

public:
    //constructors
    clsCurrency(enMode Mode,string Country,string CurrencyCode,string currencyName,float Rate) {
        this->_Mode = Mode;
        this->_Country = Country;
        this->_CurrencyCode = CurrencyCode;
        this->_CrruncyName = currencyName;
        this->_Rate = Rate;
    }

    //setters
    void UpdateRate(float Rate) { this->_Rate = Rate;
    _Update();
};

    //getters
    enMode get_Mode() { return this->_Mode; }
    string get_Country(){ return this->_Country; }
    string get_CurrencyCode() { return this->_CurrencyCode; }
    string get_CurrencyName() { return this->_CrruncyName; }
    float get_Rate() { return this->_Rate; }

    //Public functions
    bool IsEmpty() { return (this->_Mode==enMode::EmptyMode); }

    static clsCurrency FindByCode(string CurrencyCode) {
        CurrencyCode = clsString::UpperAllString(CurrencyCode);
        fstream Myfile;
        Myfile.open("Currencies.txt",ios::in);

        if (Myfile.is_open()) {
            string Line;
            while (getline(Myfile,Line)) {
                clsCurrency Currency= _ConvertLineToCurrencyObject(Line);
                if (Currency.get_CurrencyCode() ==CurrencyCode ) {
                    Myfile.close();
                    return Currency;
                }
            }
        }
        return getEmptyCurrencyObject();
    }

    static clsCurrency FindByCountry(string Country) {
        Country = clsString::UpperAllString(Country);
        fstream Myfile;
        Myfile.open("Currencies.txt", ios::in);

        if (Myfile.is_open()) {
            string Line;
            while (getline(Myfile, Line)) {
                clsCurrency Currency = _ConvertLineToCurrencyObject(Line);
                if (clsString::UpperAllString(Currency.get_Country()) == Country) {
                    Myfile.close();

```

```

        return Currency;
    }

}

return getEmptyCurrencyObject();
}

static bool IsCurrencyExist(string CurrencyCode) {
    clsCurrency c1 = FindByCode(CurrencyCode);
    return (!c1.IsEmpty());
}

static vector<clsCurrency> GetCurrenciesList() {
    return _LoadCurrenciesDataFromFile();
}

};

```

وهنا بنجرب الكود في ال main كبداية

```

#include <iostream>
#include "clsCurrency.h"

static void _PrintCurrency(clsCurrency Currency)
{
    cout << "\nCurrency Card:\n";
    cout << "_____ \n";
    cout << "\nCountry : " << Currency.get_Country();
    cout << "\nCode : " << Currency.get_CurrencyCode();
    cout << "\nName : " << Currency.get_CurrencyName();
    cout << "\nRate(1$) = :" << Currency.get_Rate();

    cout << "\n_____ \n";
}

int main()
{
    clsCurrency Currency1 = Currency1.FindByCode("jod");

    if (Currency1.IsEmpty())
    {
        cout << "\nCurrency Is Not Found!\n";
    }
    else
    {
        _PrintCurrency(Currency1);
    }

    clsCurrency Currency2 = Currency2.FindByCountry("Egypt");

    if (Currency2.IsEmpty())
    {
        cout << "\nCurrency Is Not Found!\n";
    }
    else
    {

```

```

    _PrintCurrency(Currency2);
}

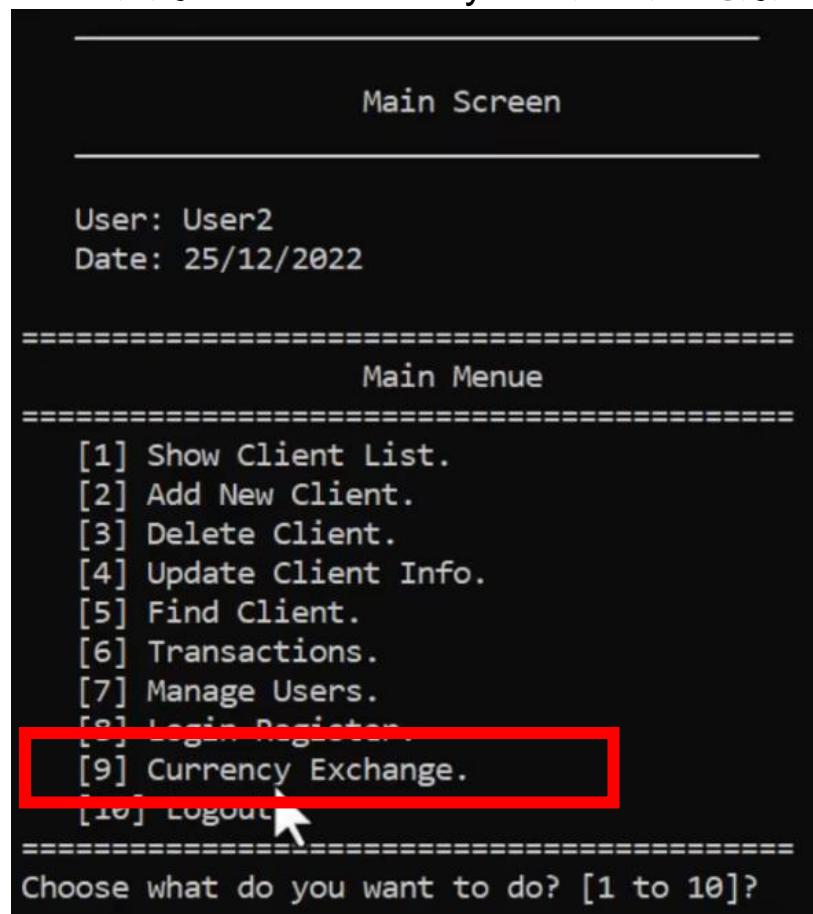
cout << "Currency1 after updating Rate:\n";
Currency1.UpdateRate(0.71);
_PrintCurrency(Currency1);

return 0;
}

```

## Currency Main Screen

عايزين نضيف قايمه ال currency للشاشة الرئيسيه



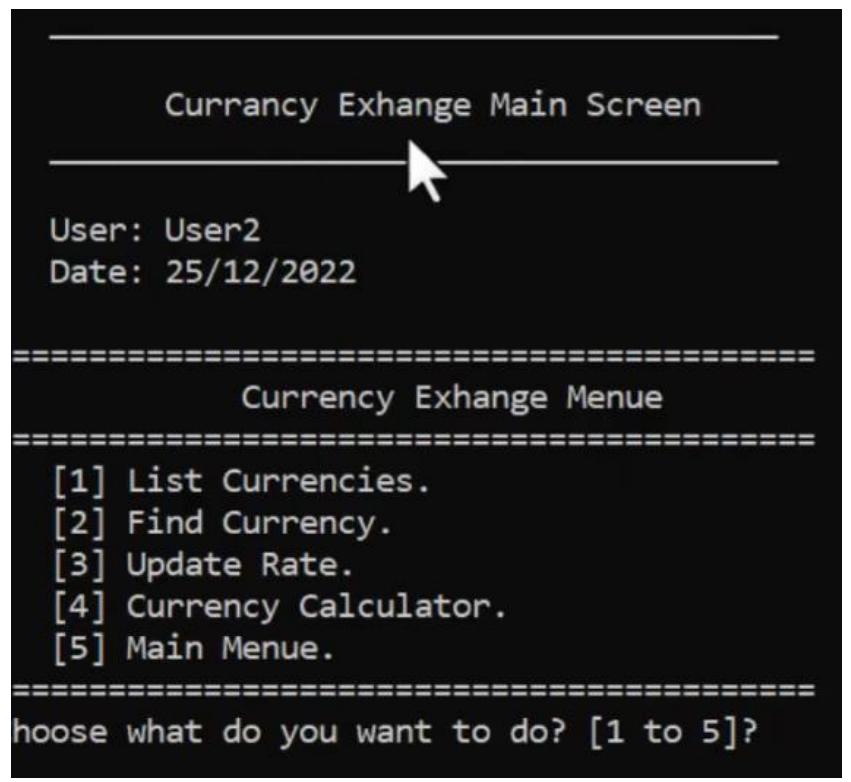
```

Main Screen

User: User2
Date: 25/12/2022

=====
Main Menue
=====
[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Transactions.
[7] Manage Users.
[8] Login Register.
[9] Currency Exchange.
[10] Logout
=====
Choose what do you want to do? [1 to 10]?

```



List Find Currency Screen Will Be Here.

Press any key to go back to Currencies Menue...

## Solution

دہ کوڈ شاشہ ال currency exchange

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsInputValidate.h"
#include <iomanip>

using namespace std;

class clsCurrencyExchangeMainScreen :protected clsScreen
{

private:
    enum enCurrenciesMainMenueOptions {
        eListCurrencies = 1, eFindCurrency = 2, eUpdateCurrencyRate = 3,
        eCurrencyCalculator = 4, eMainMenue = 5
    };

    static short ReadCurrenciesMainMenueOptions()
    {
        cout << setw(37) << left << "" << "Choose what do you want to do? [1 to 5]? ";
        short Choice = clsInputValidate::ReadShortNumberBetween(1, 5, "Enter Number between 1 to 5?");
        return Choice;
    }

    static void _GoBackToCurrenciesMenue()
    {
        cout << "\n\nPress any key to go back to Currencies Menue...";
    }
}
```

```
        system("pause>0");
        ShowCurrenciesMenue();
    }

    static void _ShowCurrenciesListScreen()
    {
        cout << "\nCurriencies List Screen Will Be Here.\n";
    }

    static void _ShowFindCurrencyScreen()
    {
        cout << "\nFind Currency Screen Will Be Here.\n";
    }

    static void _ShowUpdateCurrencyRateScreen()
    {
        cout << "\nUpdate Currency Rate Screen Will Be Here.\n";
    }

    static void _ShowCurrencyCalculatorScreen()
    {
        cout << "\nCurrency Calculator Screen Will Be Here.\n";
    }

    static void _PerformCurrenciesMainMenuOptions(enCurrenciesMainMenuOptions CurrenciesMainMenuOptions)
    {

        switch (CurrenciesMainMenuOptions)
        {
            case enCurrenciesMainMenuOptions::eListCurrencies:
            {
                system("cls");
                _ShowCurrenciesListScreen();
                _GoBackToCurrenciesMenue();
                break;
            }

            case enCurrenciesMainMenuOptions::eFindCurrency:
            {
                system("cls");
                _ShowFindCurrencyScreen();
                _GoBackToCurrenciesMenue();
                break;
            }

            case enCurrenciesMainMenuOptions::eUpdateCurrencyRate:
            {
                system("cls");
                _ShowUpdateCurrencyRateScreen();
                _GoBackToCurrenciesMenue();
                break;
            }

            case enCurrenciesMainMenuOptions::eCurrencyCalculator:
            {
                system("cls");
                _ShowCurrencyCalculatorScreen();
            }
        }
    }
}
```

```

        _GoBackToCurrenciesMenue();
        break;
    }

    case enCurrenciesMainMenuOptions::eMainMenu:
    {
        //do nothing here the main screen will handle it :-);
    }
}

public:

static void ShowCurrenciesMenue()
{
    system("cls");
    _DrawScreenHeader(" Currency Exchange Main Screen","");
}

cout << setw(37) << left << "=====\n";
cout << setw(37) << left << " \t\t Currency Exchange Menue\n";
cout << setw(37) << left << "=====\n";
cout << setw(37) << left << " \t[1] List Currencies.\n";
cout << setw(37) << left << " \t[2] Find Currency.\n";
cout << setw(37) << left << " \t[3] Update Rate.\n";
cout << setw(37) << left << " \t[4] Currency Calculator.\n";
cout << setw(37) << left << " \t[5] Main Menue.\n";
cout << setw(37) << left << "=====\n";

_PerformCurrenciesMainMenuOptions((enCurrenciesMainMenuOptions)ReadCurrenciesMainMenuOptions());
}
};

};


```

ده کود شاشة ال main screen

```

#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsInputValidate.h"
#include "clsClientListScreen.h"
#include "clsAddNewClientScreen.h"
#include "clsDeleteClientScreen.h"
#include "clsUpdateClientScreen.h"
#include "clsFindClientScreen.h"
#include "clsTransactionsScreen.h"
#include "clsManageUsersScreen.h"
#include "clsLoginScreen.h"
#include "Global.h"
#include "clsLoginRegisterScreen.h"
#include "clsCurrencyExchangeMainScreen.h"

using namespace std;

class clsMainScreen :protected clsScreen
{

private:
    enum enMainMenuOptions {
        eListClients = 1, eAddNewClient = 2, eDeleteClient = 3,

```

```

eUpdateClient = 4, eFindClient = 5, eShowTransactionsMenue = 6,
eManageUsers = 7, eLoginRegister = 8, eCurrnicyExchange = 9, eExit = 10
};

static short _ReadMainMenuOption()
{
    cout << setw(37) << left << "" << "Choose what do you want to do? [1 to 10]? ";
    short Choice = clsInputValidate::ReadShortNumberBetween(1, 10, "Enter Number between 1 to 10? ");
    return Choice;
}

static void _GoBackToMainMenu()
{
    cout << setw(37) << left << "" << "\n\tPress any key to go back to Main Menue...\n";

    system("pause>0");
    ShowMainMenu();
}

static void _ShowAllClientsScreen()
{
    // cout << "\nClient List Screen Will be here...\n";
    clsClientListScreen::ShowClientsList();

}

static void _ShowAddNewClientsScreen()
{
    // cout << "\nAdd New Client Screen Will be here...\n";
    clsAddNewClientScreen::ShowAddNewClientScreen();

}

static void _ShowDeleteClientScreen()
{
    //cout << "\nDelete Client Screen Will be here...\n";
    clsDeleteClientScreen::ShowDeleteClientScreen();

}

static void _ShowUpdateClientScreen()
{
    //cout << "\nUpdate Client Screen Will be here...\n";
    clsUpdateClientScreen::ShowUpdateClientScreen();

}

static void _ShowFindClientScreen()
{
    // cout << "\nFind Client Screen Will be here...\n";
    clsFindClientScreen::ShowFindClientScreen();
}

static void _ShowTransactionsMenue()
{
    // cout << "\nTransactions Menue Will be here...\n";
    clsTransactionsScreen::ShowTransactionsMenue();
}

static void _ShowManageUsersMenue()

```

```

{
    // cout << "\nUsers Menue Will be here...\n";
    clsManageUsersScreen::ShowManageUsersMenue();
}

static void _ShowLoginRegisterScreen()
{
    // cout << "\nLogin Register Will be here...\n";
    clsLoginRegisterScreen::ShowLoginRegisterScreen();
}

static void _ShowCurrencyExchangeMainScreen()
{
    clsCurrencyExchangeMainScreen::ShowCurrenciesMenue();
}

static void _Logout()
{
    CurrentUser = clsUser::Find("", "");
    //then it will go back to main function.
}

static void _PerfromMainMenueOption(enMainMenueOptions MainMenueOption)
{
    switch (MainMenueOption)
    {
        case enMainMenueOptions::eListClients:
        {
            system("cls");
            _ShowAllClientsScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eAddNewClient:
        {
            system("cls");
            _ShowAddNewClientsScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eDeleteClient:
        {
            system("cls");
            _ShowDeleteClientScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eUpdateClient:
        {
            system("cls");
            _ShowUpdateClientScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eFindClient:
        {
            system("cls");
            _ShowFindClientScreen();
            _GoBackToMainMenue();
            break;
        }
        case enMainMenueOptions::eShowTransactionsMenue:
        {
            system("cls");
            _ShowTransactionsMenue();
        }
    }
}

```

```

_GoBackToMainMenue();
break;

case enMainMenueOptions::eManageUsers:
system("cls");
>ShowManageUsersMenue();
_GoBackToMainMenue();
break;

case enMainMenueOptions::eLoginRegister:
system("cls");
>ShowLoginRegisterScreen();
_GoBackToMainMenue();
break;

case enMainMenueOptions::eCurrencyExchange:
system("cls");
>ShowCurrencyExchangeMainScreen();
_GoBackToMainMenue();
break;

case enMainMenueOptions::eExit:
system("cls");
Logout();
break;
}

}

public:

static void ShowMainMenue()
{
system("cls");
_DrawScreenHeader("\t\tMain Screen","");
cout << setw(37) << left << "=====Main Menue=====\n";
cout << setw(37) << left << " \t\tMain Menue\n";
cout << setw(37) << left << "=====Main Menue=====\n";
cout << setw(37) << left << "\t[1] Show Client List.\n";
cout << setw(37) << left << "\t[2] Add New Client.\n";
cout << setw(37) << left << "\t[3] Delete Client.\n";
cout << setw(37) << left << "\t[4] Update Client Info.\n";
cout << setw(37) << left << "\t[5] Find Client.\n";
cout << setw(37) << left << "\t[6] Transactions.\n";
cout << setw(37) << left << "\t[7] Manage Users.\n";
cout << setw(37) << left << "\t[8] Login Register.\n";
cout << setw(37) << left << "\t[9] Currency Exchange.\n";
cout << setw(37) << left << "\t[10] Logout.\n";
cout << setw(37) << left << "=====Main Menue=====\n";

_PerformMainMenueOption((enMainMenueOptions)_ReadMainMenueOption());
}

};


```

## Currencies List Screen

## عاوزين نعمل شاشة list currencies

C:\Users\USER\source\repos\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe

### Currencies List Screen (222) Currency.

User: User2  
Date: 26/12/2022



Country	Code	Name	Rate/(1\$)
United States of America	USD	US Dollar	1
Afghanistan	AFN	Afghanistan Afghani	87.48
Albania	ALL	Albania Lek(e)	109.01
Algeria	DZD	Algerian Dinar	137.046
American Samoa	USD	US Dollar	1
France	EUR	Euro	0.9
Angola	AOA	Angolan Kwanza	504.734
Anguilla	XCD	E.C. Dollar	
Antigua and Barbuda	XCD	E.C. Dollar	
Argentina	ARS	Argentine Peso	
Armenia	AMD	Armenian Dram	
Aruba	AWG	Aruban Guilder	
Australia	AUD	Australian Dollar	
Austria	EUR	Euro	
Azerbaijan	AZN	Azerbaijan Manat	1.694

## Solution

ده كود الشاشه

```
#pragma once

#include <iostream>
#include "clsScreen.h"
#include "clsCurrency.h"
#include <iomanip>

class clsListCurrenciesScreen :protected clsScreen
{
private:
    static void PrintCurrencyRecordLine(clsCurrency Currency)
    {
        cout << setw(8) << left << "" << "|" << setw(30) << left << Currency.get_Country();
        cout << "|" << setw(8) << left << Currency.get_CurrencyCode();
        cout << "|" << setw(45) << left << Currency.get_CurrencyName();
        cout << "|" << setw(10) << left << Currency.get_Rate();
    }

public:
    static void ShowCurrenciesList()
    {
```

```

vector <clsCurrency> vCurrencys = clsCurrency::GetCurrenciesList();
string Title = "\t Currencies List Screen";
string SubTitle = "\t (" + to_string(vCurrencys.size()) + ") Currency.';

_DrawScreenHeader(Title, SubTitle);
cout << setw(8) << left << "" << "\n\t_____"; 
cout << "_____ \n" << endl;

cout << setw(8) << left << "" << "|" << left << setw(30) << "Country";
cout << "|" << left << setw(8) << "Code";
cout << "|" << left << setw(45) << "Name";
cout << "|" << left << setw(10) << "Rate/(1$)";
cout << setw(8) << left << "" << "\n\t_____"; 
cout << "_____ \n" << endl;

if (vCurrencys.size() == 0)
    cout << "\t\t\tNo Currencies Available In the System!";
else

    for (clsCurrency Currency : vCurrencys)
    {

        PrintCurrencyRecordLine(Currency);
        cout << endl;
    }

    cout << setw(8) << left << "" << "\n\t_____"; 
    cout << "_____ \n" << endl;
}

};


```

وذه التعديل على كود القائمه

```

static void _ShowCurrenciesListScreen()
{
    // cout << "\nCurrencies List Screen Will Be Here.\n";
    clsListCurrenciesScreen::ShowCurrenciesList();
}

```

## Find Currency

عاوزين نعمل شاشة ال find currency

## Find Currency Screen

User: User2  
Date: 26/12/2022

Find By: [1] Code or [2] Country ? 2

Please Enter Country Name: egypt

Currency Found :-)

Currency Card:

---

Country : Egypt  
Code : EGP  
Name : Egyptian Pound  
Rate(1\$) = : 24.592

---

Press any key to go back to Currencies Menue...

## Solution

دہ کوڈ الشاشہ

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsCurrency.h"
#include "clsInputValidate.h"

class clsFindCurrencyScreen :protected clsScreen
{
private:
    static void _PrintCurrency(clsCurrency Currency)
    {
        cout << "\nCurrency Card:\n";
        cout << "_____ \n";
        cout << "\nCountry : " << Currency.get_Country();
        cout << "\nCode : " << Currency.get_CurrencyCode();
        cout << "\nName : " << Currency.get_CurrencyName();
        cout << "\nRate(1$) = : " << Currency.get_Rate();

        cout << "\n_____ \n";
    }

    static void _ShowResults(clsCurrency Currency)
    {
        if (!Currency.IsEmpty())
        {
            cout << "\nCurrency Found :-)\n";
            _PrintCurrency(Currency);
        }
    }
}
```

```

else
{
    cout << "\nCurrency Was not Found :-(\n";
}
}

public:

static void ShowFindCurrencyScreen()
{
    _DrawScreenHeader("\t Find Currency Screen","");
    cout << "\nFind By: [1] Code or [2] Country ? ";
    short Answer = 1;

    cin >> Answer;

    if (Answer == 1)
    {
        string CurrencyCode;
        cout << "\nPlease Enter CurrencyCode: ";
        CurrencyCode = clsInputValidate::ReadString();
        clsCurrency Currency = clsCurrency::FindByCode(CurrencyCode);
        _ShowResults(Currency);
    }
    else
    {
        string Country;
        cout << "\nPlease Enter Country Name: ";
        Country = clsInputValidate::ReadString();
        clsCurrency Currency = clsCurrency::FindByCountry(Country);
        _ShowResults(Currency);
    }
}
};

};

}

```

وده التعديل على كود القائمه

```

static void _ShowFindCurrencyScreen()
{
    // cout << "\nFind Currency Screen Will Be Here.\n";
    clsFindCurrencyScreen::ShowFindCurrencyScreen();
}

```

### Update Currency Rate

عاوزين نعمل شاشة ال update

Please Enter Currency Code: jod

Currency Card:

---

Country : Jordan  
 Code : JOD  
 Name : Jordanian Dinar  
 Rate(1\$) = : 0.71

---

Are you sure you want to update the rate of this Currency y/n? y

Update Currency Rate:

---

Enter New Rate: -

Enter New Rate: 0.708

Currency Rate Updated Successfully :-)

Currency Card:

---

Country : Jordan  
 Code : JOD  
 Name : Jordanian Dinar  
 Rate(1\$) = : 0.708

---

Press any key to go back to Currencies Menue...

### Solution

دہ کوڈ الشاشہ

```
#pragma once
#include <iostream>
#include "clsScreen.h"
#include "clsCurrency.h"
#include "clsInputValidate.h"

class clsUpdateCurrencyRateScreen :protected clsScreen
{
private:

    static float _ReadRate()
    {
        cout << "\nEnter New Rate: ";
        float NewRate = 0;
```

```

NewRate = clsInputValidate::ReadFloatNumber();
return NewRate;
}

static void _PrintCurrency(clsCurrency Currency)
{
    cout << "\nCurrency Card:\n";
    cout << "_____ \n";
    cout << "\nCountry : " << Currency.get_Country();
    cout << "\nCode : " << Currency.get_CurrencyCode();
    cout << "\nName : " << Currency.get_CurrencyName();
    cout << "\nRate(1$) = : " << Currency.get_Rate();

    cout << "\n_____ \n";
}

public:

static void ShowUpdateCurrencyRateScreen()
{
    _DrawScreenHeader("\tUpdate Currency Screen","");
    string CurrencyCode = "";

    cout << "\nPlease Enter Currency Code: ";
    CurrencyCode = clsInputValidate::ReadString();

    while (!clsCurrency::IsCurrencyExist(CurrencyCode))
    {
        cout << "\nCurrency is not found, choose another one: ";
        CurrencyCode = clsInputValidate::ReadString();
    }

    clsCurrency Currency = clsCurrency::FindByCode(CurrencyCode);
    _PrintCurrency(Currency);

    cout << "\nAre you sure you want to update the rate of this Currency y/n? ";

    char Answer = 'n';
    cin >> Answer;

    if (Answer == 'y' || Answer == 'Y')
    {
        cout << "\n\nUpdate Currency Rate:";
        cout << "\n_____ \n";

        Currency.UpdateRate(_ReadRate());

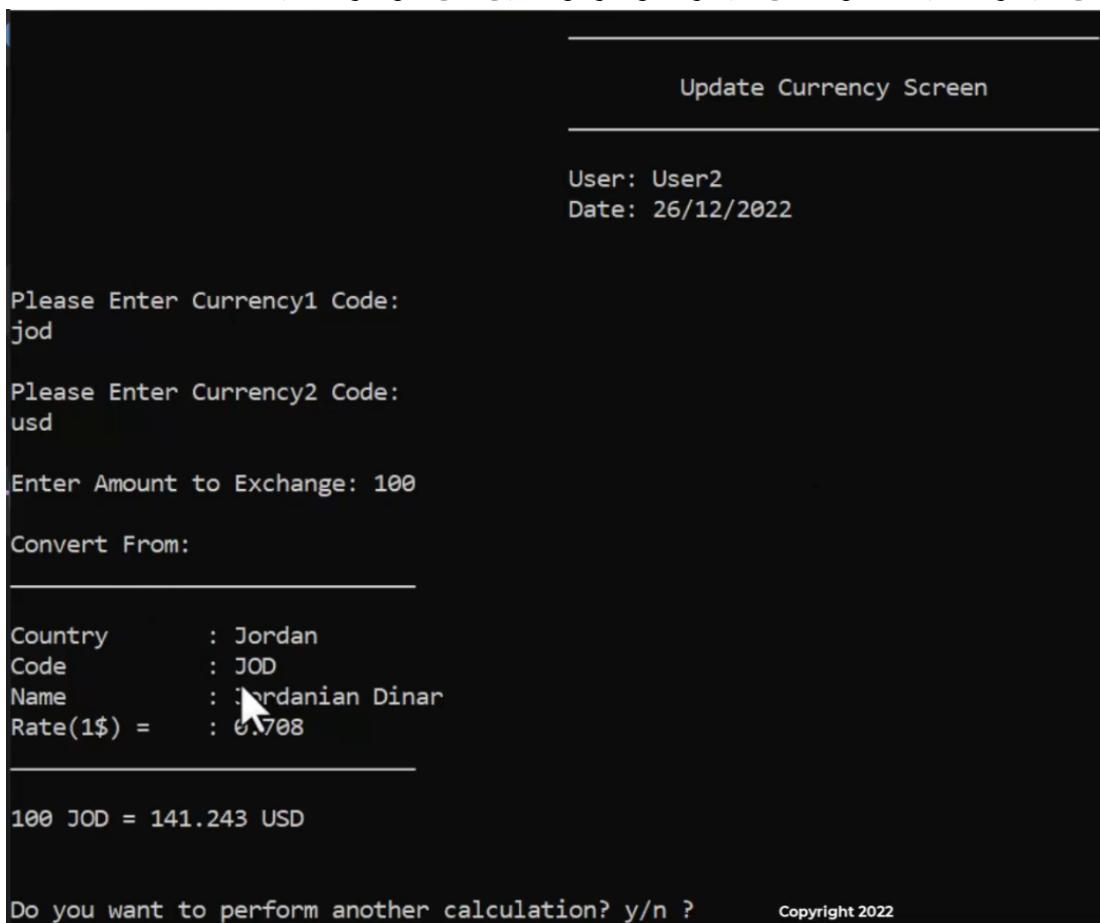
        cout << "\nCurrency Rate Updated Successfully :-) \n";
        _PrintCurrency(Currency);
    }
}
};


```

```
static void _ShowUpdateCurrencyRateScreen()
{
    // cout << "\nUpdate Currency Rate Screen Will Be Here.\n";
    clsUpdateCurrencyRateScreen::ShowUpdateCurrencyRateScreen();
}
```

## Currency Calculator Screen

عاوزين نعمل شاشة بتحول من أي عمله لاي عمله  
لو حولت من دينار لجنيه بتحول من دينار لدولار وبعدين من دولار لجنيه



```

Please Enter Currency2 Code:
egp

Enter Amount to Exchange: 100

Convert From:
_____
Country      : Jordan
Code         : JOD
Name        : Jordanian Dinar
Rate(1$) =   : 0.708
_____
100 JOD = 141.243 USD

Converting from USD to:

To:
_____
Country      : Egypt
Code         : EGP
Name        : Egyptian Pound
Rate(1$) =   : 24.592
_____
100 JOD = 3473.45 EGP

Do you want to perform another calculation? y/n ? 

```

### Solution

ده الی زوـنـاه عـلـی كـلاـس الـ **clsCurrency**

```

float ConvertToUSD(float Amount)
{
    return (float)(Amount / get_Rate());
}

float ConvertToOtherCurrency(float Amount, clsCurrency Currency2)
{
    float AmountInUSD = ConvertToUSD(Amount);

    if (Currency2.get_CurrencyCode() == "USD")
    {
        return AmountInUSD;
    }

    return (float)(AmountInUSD * Currency2.get_Rate());
}

```

ده كـوـد الشـاشـه

```

#pragma once
#include <iostream>
#include "clsScreen.h"

```

```

#include "clsCurrency.h"
#include "clsInputValidate.h"

class clsCurrencyCalculatorScreen :protected clsScreen

{
private:

    static float _ReadAmount()
    {
        cout << "\nEnter Amount to Exchange: ";
        float Amount = 0;

        Amount = clsInputValidate::ReadFloatNumber();
        return Amount;
    }

    static clsCurrency _GetCurrency(string Message)
    {

        string CurrencyCode;
        cout << Message << endl;

        CurrencyCode = clsInputValidate::ReadString();

        while (!clsCurrency::IsCurrencyExist(CurrencyCode))
        {
            cout << "\nCurrency is not found, choose another one: ";
            CurrencyCode = clsInputValidate::ReadString();
        }

        clsCurrency Currency = clsCurrency::FindByCode(CurrencyCode);
        return Currency;
    }

    static void _PrintCurrencyCard(clsCurrency Currency, string Title = "Currency Card:")
    {

        cout << "\n" << Title << "\n";
        cout << "_____ \n";
        cout << "\nCountry : " << Currency.get_Country();
        cout << "\nCode : " << Currency.get_CurrencyCode();
        cout << "\nName : " << Currency.get_CurrencyName();
        cout << "\nRate(1$) = : " << Currency.get_Rate();
        cout << "\n_____ \n\n";

    }

    static void _PrintCalculationsResults(float Amount, clsCurrency Currency1, clsCurrency Currency2)
    {

        _PrintCurrencyCard(Currency1, "Convert From:");

        float AmountInUSD = Currency1.ConvertToUSD(Amount);

        cout << Amount << " " << Currency1.get_CurrencyCode()
        << " = " << AmountInUSD << " USD\n";
    }
}

```

```

if (Currency2.get_CurrencyCode() == "USD")
{
    return;
}

cout << "\nConverting from USD to:\n";

_PrintCurrencyCard(Currency2, "To:");

float AmountInCurrency2 = Currency1.ConvertToOtherCurrency(Amount, Currency2);

cout << Amount << " " << Currency1.get_CurrencyCode()
<< " = " << AmountInCurrency2 << " " << Currency2.get_CurrencyCode();

}

public:

static void ShowCurrencyCalculatorScreen()
{
    char Continue = 'y';

    while (Continue == 'y' || Continue == 'Y')
    {
        system("cls");

        _DrawScreenHeader("\tUpdate Currency Screen","");
        clsCurrency CurrencyFrom = _GetCurrency("\nPlease Enter Currency1 Code: ");
        clsCurrency CurrencyTo = _GetCurrency("\nPlease Enter Currency2 Code: ");
        float Amount = _ReadAmount();

        _PrintCalculationsResults(Amount, CurrencyFrom, CurrencyTo);

        cout << "\n\nDo you want to perform another calculation? y/n ? ";
        cin >> Continue;
    }
}
};

}

```

وده التعديل على القيمه

```

case enCurrenciesMainMenueOptions::eCurrencyCalculator:
{
    system("cls");
    _ShowCurrencyCalculatorScreen();
    _GoBackToCurrenciesMenue();
    break;
}

```

## Template Functions

ال **template functions** بتتوفر عليك كتابه الكود فلو عندك مثلا **function max** اسمها **مره** بتأخذ **int** ومره **float** ومره **short** بدل مانكتب الكود 3 او اربع مرات قالك لا انت تكتب الكود مره واحده وال **data type** تخليه

طريقتها انك تكتب كلمة **template** وبعدها العلامتين دول **<>** وبينهم بتكتب **typename** وبعدها بتكتب **T** ودي بتعبر عن ال **data type** اللي خليناه متغير وبعدين **T** وبتكتب ال **function** عادي **bas** أي **parameter** بتاخدها هتكون من النوع **T**

```
template <typename T> T myMax(T Number1, T Number2) { ...  
  
int main()  
{  
    cout << myMax<int>(3, 7) << endl; // Call myMax for int  
  
    cout << myMax<double>(5.3, 4.2)  
        << endl; // call myMax for double  
  
    cout << myMax<char>('a', 'b')  
        << endl; // call myMax for char  
  
    return 0;  
}
```

```
template <typename T> T myMax(T Number1, T Number2)  
{  
    return (Number1 > Number2) ? Number1 : Number2;  
}
```

### Template Classes

زي ما عرفت انك تقدر تخلی نوع ال **function** الكلاس برضه تقدر تخلی المتغيرات اللي جواه من نوع متغير ومثال عليها هو ال **vector** لما كنت بتعرفه كنت بتعرف النوع بين اقواس **<>**

طريقته انك بتحطي قبل الكلاس وبتكتب **template <class T>** واي متغير بيجي لك بتعرفه من النوع **t**

```
#include <iostream>
using namespace std;

#include <iostream>
using namespace std;

template<class T>
class Calculator {

private:
    T Number1, Number2;

public:
    Calculator(T n1, T n2) { ... }

    void PrintResults() { ... }

    T Add() { ... }
}
```

Copyright 2022

```
T Add()
{
    return Number1 + Number2;
}

T Subtract()
{
    return Number1 - Number2;
}

T Multiply()
{
    return Number1 * Number2;
}

T Divide()
{
    return Number1 / Number2;
}
```

Copyright 2022

```

int main() {
    Calculator<int> intCalc(2, 1);
    Calculator<float> floatCalc(2.4, 1.2);

    cout << "Int results:" << endl;
    intCalc.PrintResults();

    cout << endl
        << "Float results:" << endl;
    floatCalc.PrintResults();

    return 0;
}

```

Copyright 2022

## Homework

ارجو الذهاب لمكتبه

clsInputValidationn

واختصار الفنكشنز باستخدام

Template Function

```

#pragma once
#include <iostream>
#include <string>
#include "clsString.h"
#include "clsDate.h"

class clsInputValidateTemplate
{
public:

    template<typename T> static bool IsNumberBetween(T Number, T From, T To)
    {
        if (Number >= From && Number <= To)
            return true;
        else
            return false;
    }
}

```

```

static bool IsDateBetween(clsDate Date, clsDate From, clsDate To)
{
    //Date>=From && Date<=To
    if ((clsDate::IsDate1AfterDate2(Date, From) || clsDate::IsDate1EqualDate2(Date, From))
        &&
        (clsDate::IsDate1BeforeDate2(Date, To) || clsDate::IsDate1EqualDate2(Date, To)))
    )
    {
        return true;
    }

    //Date>=To && Date<=From
    if ((clsDate::IsDate1AfterDate2(Date, To) || clsDate::IsDate1EqualDate2(Date, To))
        &&
        (clsDate::IsDate1BeforeDate2(Date, From) || clsDate::IsDate1EqualDate2(Date, From)))
    )
    {
        return true;
    }

    return false;
}

```

```

template<typename T>static T ReadNumber(string ErrorMessage = "Invalid Number, Enter again\n")
{

```

```

    T Number;
    while (!(cin >> Number)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << ErrorMessage;
    }
    return Number;
}

```

```

template<typename T>static T ReadNumberBetween(T From, T To, string ErrorMessage = "Number is not within
range, Enter again:\n")
{

```

```

    T Number = ReadShortNumber();

    while (!IsNumberBetween(Number, From, To))
    {
        cout << ErrorMessage;
        Number = ReadShortNumber();
    }
    return Number;
}

```

```

static bool IsValidDate(clsDate Date)
{
    return clsDate::IsValidDate(Date);
}

```

```

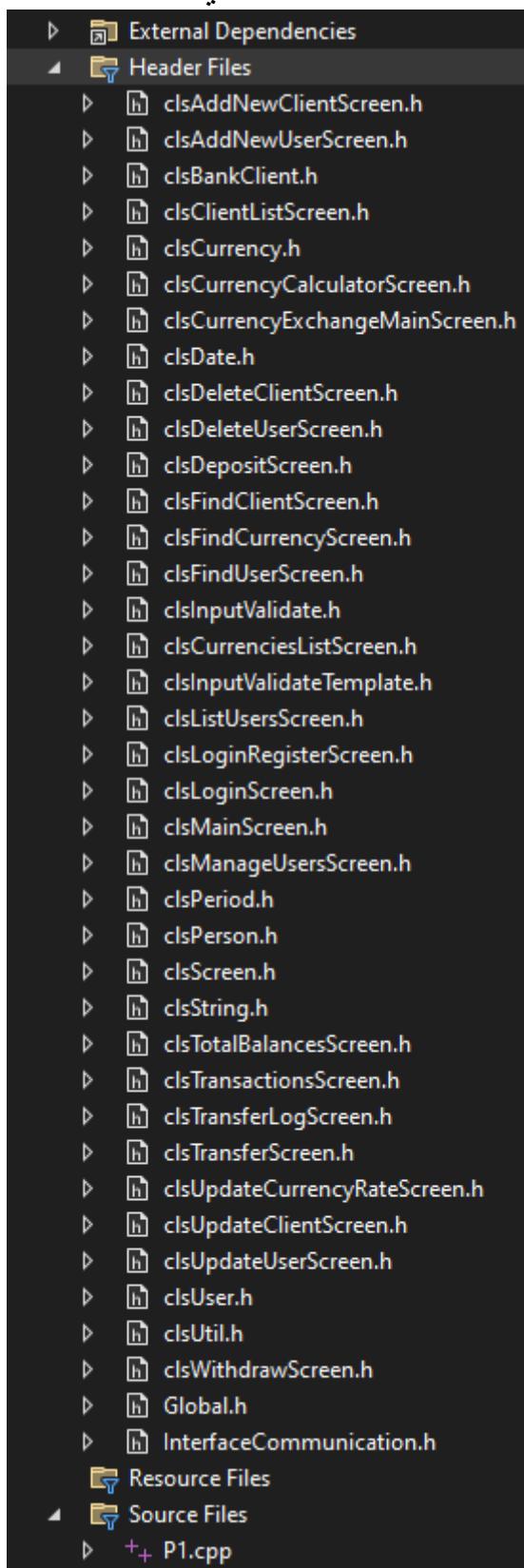
static string ReadString()
{
    string S1 = "";
    // Usage of std::ws will extract allthe whitespace character
    getline(cin >> ws, S1);
}

```

```
    return S1;  
}  
  
};
```

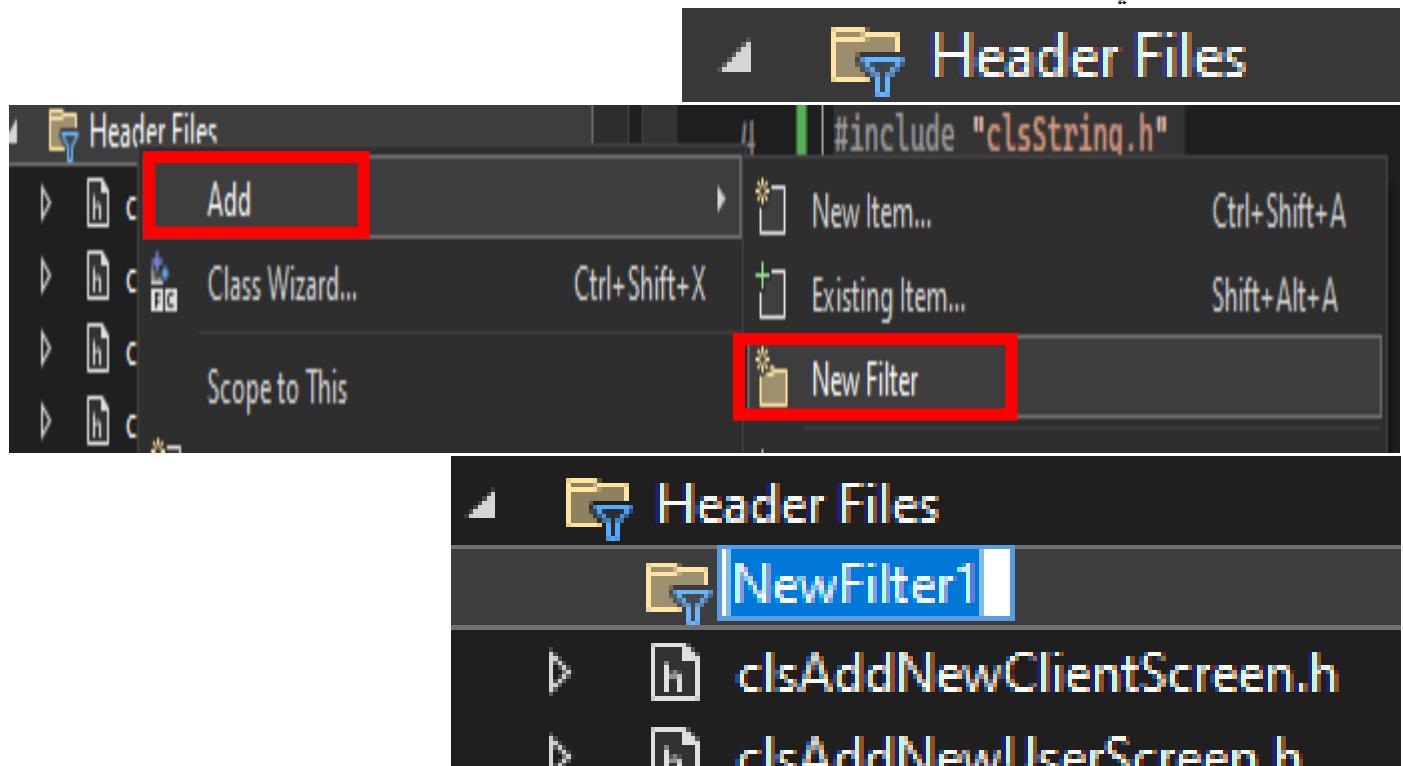
## Organize Your Classes

هذا ده شكل الملفات عندي

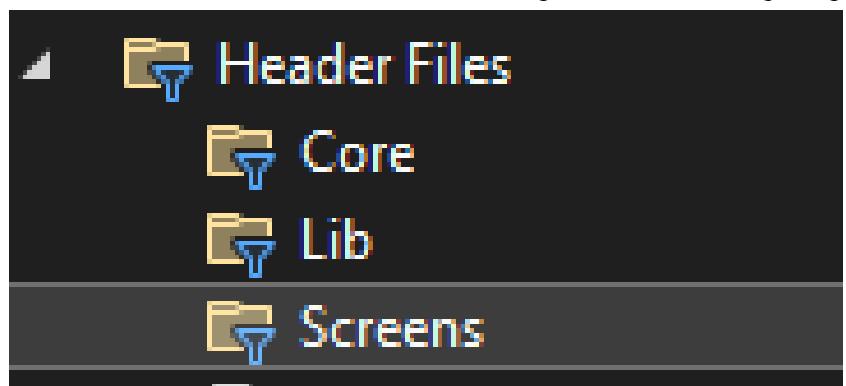


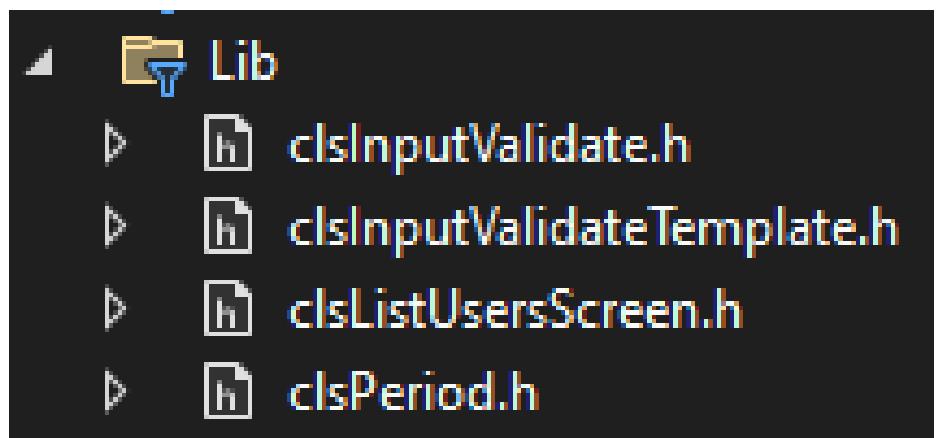
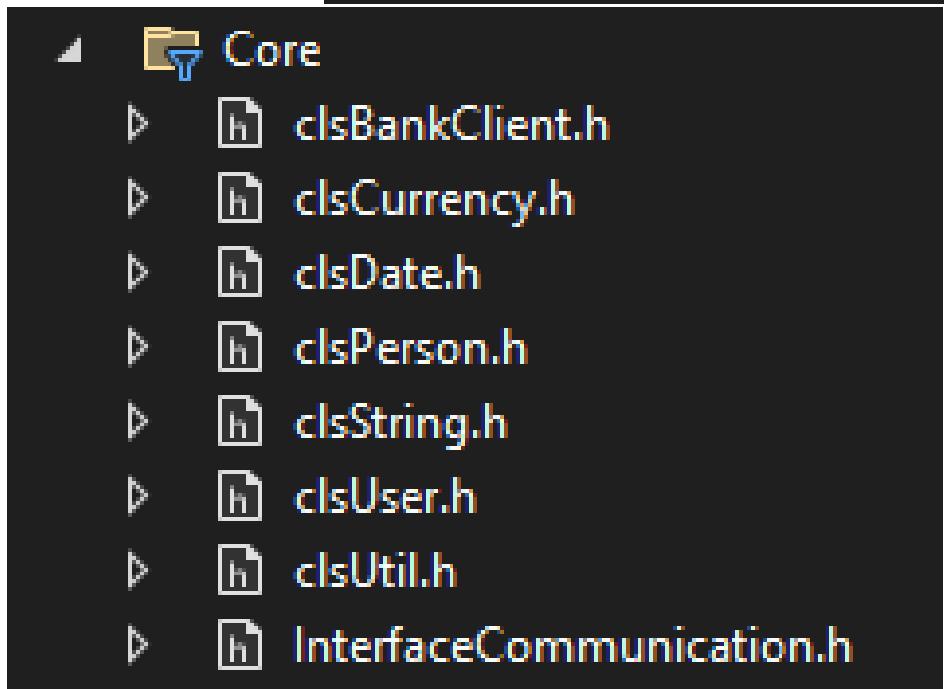
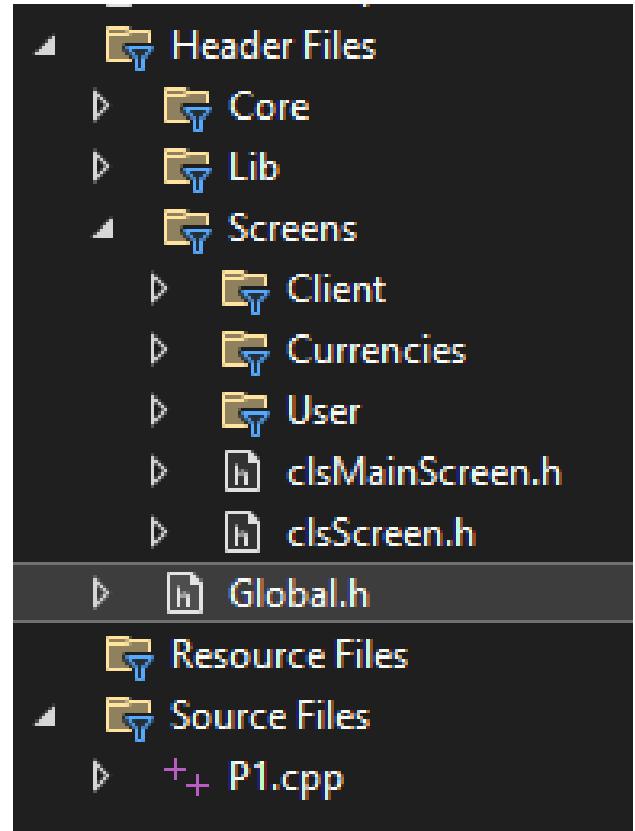
عشان ادور علي ملف فيهم بتعب  
قالك انك ممكن تقسمهم وترتبهم عادي

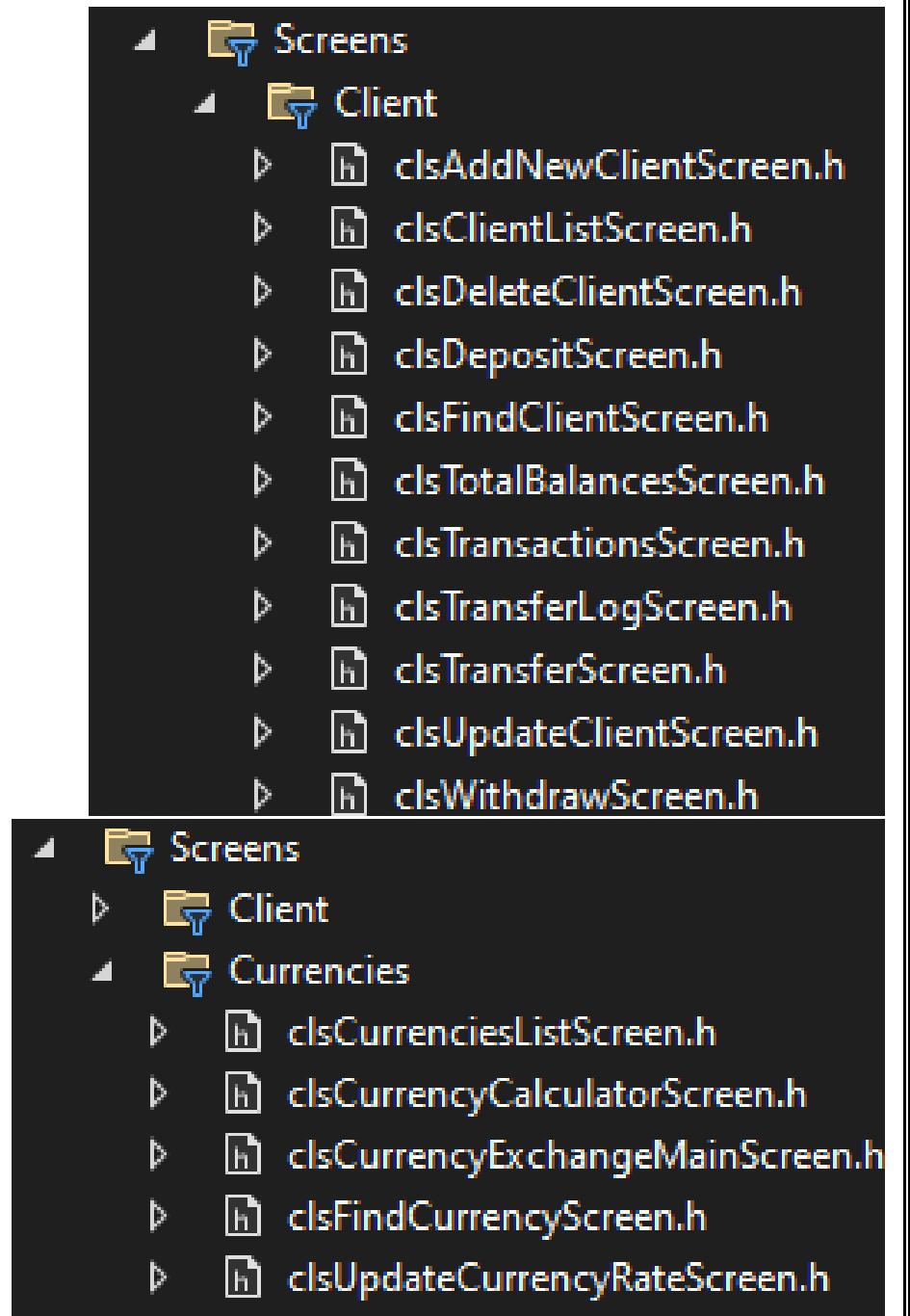
بتعمل كليك يمين علي ال HEADER FILES وبختار new filter

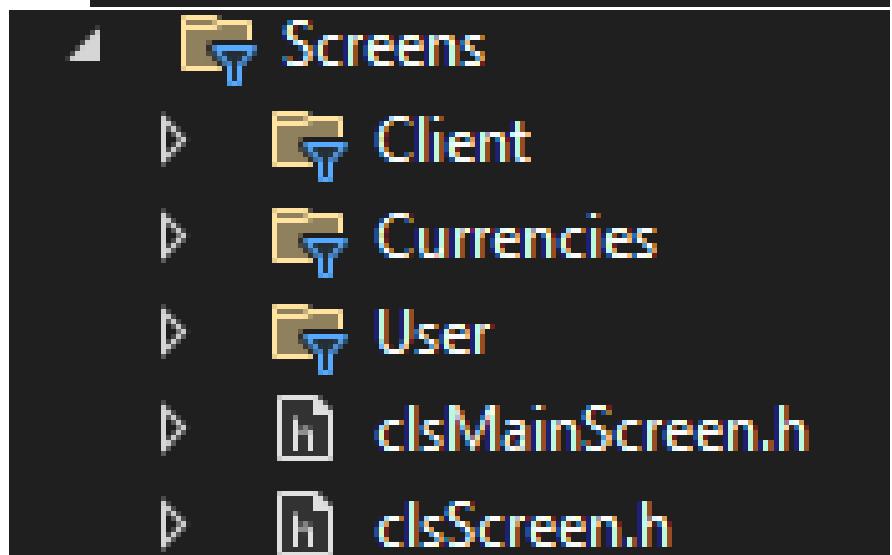
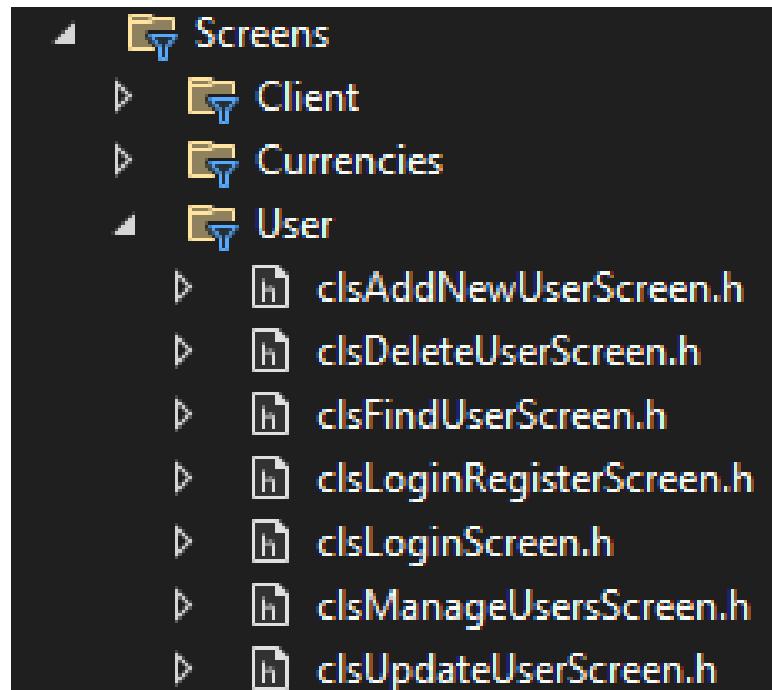


بعد كده بتسميه و بتسحب الكلاس و بترميه فيه بالماوس  
وبتقدر تعمل filter جوه ال









**END**