

YAPILAN İŞ : C++ komutları ve yazımına aşina olmak ve nesneye yönelik programlama konusuna yatkınlığı artırmak için doküman okuma ve örnek projeler ile çalışma.

```
#include <iostream>

using namespace std;

int sum_(int x, int y) {
    return x + y;
}

float sum_(float x, float y) {
    return x + y;
}

int main()
{
    int x, y;
    float a, b;
    cout << "iki sayi giriniz: ";
    cin >> x;
    cin >> y;
    cout << sum_(x, y) << endl;
    cout << "iki sayi giriniz: ";
    cin >> a;
    cin >> b;
    cout << sum_(a, b) << endl;
}
```

Görsel-1: Function Overloading

Not: Aynı isimli, farklı veri tiplerini işleyen iki fonksiyon kullanılarak overloading kavramı incelendi.

```
iki sayi giriniz: 4
5
9
iki sayi giriniz: 2.7
4.6
7.3
```

Görsel-2:

```

#include <iostream>
#include <vector>

using namespace std;

int main()
{
    int satir;
    cout << "Kac satir gosterilsin:" << endl;
    cin >> satir;
    cout << "*****" << endl << endl;
    vector<int> vect1;
    vector<int> vect2;
    vect1.assign(1, 1);
    vect2.assign(2, 1);
    cout << vect1[0] << endl;
    cout << vect2[0] << " " << vect2[1] << endl;
    int j = 3;
    while (j <= satir) {
        vect1.swap(vect2);
        vect2.clear();
        //cout << vect1[0] << vect1[1] << endl;
        for (int i = 0; i <= vect1.size(); i++) {
            auto itPos = vect2.begin() + i;
            if (i == 0 || i == vect1.size()) {
                vect2.insert(itPos, 1, 1);
            }
            else {
                int x = vect1[i] + vect1[i - 1];
                vect2.insert(itPos, x);
            }
        }
        for (int i = 0; i <= vect2.size() - 1; i++) {
            cout << vect2[i] << " ";
        }
        cout << endl;
        j++;
    }
}

```

Not: Vector objeleri oluřturulduktan sonra assign ile ilk iki satır atanmakta. Ardından vect2'nin içerięi vect1'e geęirilip vect2'nin içerięi silinmekte. Böylece sonraki for dōngüsü içinde vect2 üstüne bir sonraki satır, vect1'deki veriler üzerinden (binom üçgeninde bir satır üstündeki satırın toplamlarından oluşur.) yazılıyor. Bu işlem istenilen satıra gelinceye kadar devam ediyor.



Görsel-4: Binom Üçgeni

Görsel-3: Vector ile istenilen satır kadar binom üçgeninin yazdırılması

```

Kac satir gosterilsin:
10
*****

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

```

Görsel-5: Binom Üçgeni Uygulamasının Çıktısı

```

#include <iostream>
#include <vector>
#include <string>
#include <typeinfo>

using namespace std;

class Vehicle {
public:
    const char* name;
    string brand;
    int speed;
    string color;
    string objectName;
};

class Car : public Vehicle {
public:
    //int speed;
    //string objectName;
    //const char* name;
    //string color;
    Car(int speed_, string color_, string brand_, string name_) {
        speed = speed_;
        color = color_;
        brand = brand_;
        objectName = name_;
    }
};

class Motor : public Vehicle {
public:
    //int speed;
    //string objectName;
    //const char* name;
    //string color;
    Motor(int speed_, string color_, string brand_, string name_) {
        speed = speed_;
        color = color_;
        brand = brand_;
        objectName = name_;
    }
};

int main()
{
    Motor motor1(145, "black", "Suzuki", "Motor");
    Car car1(100, "red", "Fiat", "Car");
    Car car2(60, "blue", "Renault", "Car");
    vector<class Car> cars;
    vector<class Vehicle> vehicles;
    vehicles.push_back(motor1);
    vehicles.push_back(car1);
    vehicles.push_back(car2);
    cars.push_back(car1);
    cars.push_back(car2);
    for (int i = 0; i < cars.size(); i++) {
        //cout << "Your " << cars[i].name << " is a " << cars[i].color << " " << cars[i].brand << " and is going " << cars[i].speed << " kms per hour" << endl;
    }
    for (int i = 0; i < vehicles.size(); i++) {
        cout << "Your " << vehicles[i].objectName << " is a " << vehicles[i].color << " " << vehicles[i].brand << " and is going " << vehicles[i].speed << " kms per hour" << endl;
    }
}

```

Görsel-6: Vector ve Class Mantığı Hakkında Uygulamalar

Not: Vector içerisine bir sınıftan türetilmiş nesneler atılarak daha dinamik kodlar yazılabilir.

```

#include <iostream>

using namespace std;

void numSwap(int x, int y) {
    int z = x;
    x = y;
    y = z;
}

void numSwapWithRef(int &x, int &y) {
    int z = x;
    x = y;
    y = z;
}

int main()
{
    int a, b;
    cout << "İki Sayı Giriniz: \n";
    cin >> a;
    cin >> b;
    cout << "Önce: " << a << " " << b << endl;
    numSwap(a, b); // değişim fonksiyon dışına çıkmıyor
    cout << "Sonra: " << a << " " << b << endl;
    numSwapWithRef(a, b); // değişim fonksiyon dışına çıkıyor (a ve b)
    cout << "Sonra (REF): " << a << " " << b << endl;
}

```

Görsel-7: Reference Kullanımı ile Fonksiyon Dışındaki Değişkenlerin Değiştirilmesi

Not: Reference kullanımı olmadığı takdirde fonksiyona verilen değişkenler fonksiyon dışına çıkıldığında swap edilmeden kalmaktadır.

```

#include <iostream>
#include <string>

using namespace std;

class Box {
private:
    int length;
public:
    void setLength(int len) {
        length = len;
    }
    friend class ClassTwo;
    friend void printLength(Box box);
};

class ClassTwo {
public:
    void greeting(Box box) {
        cout << "Hello your box has a length of " << box.length << " cm" << endl; //private verilere de ulaşabiliyor
    }
};

void printLength(Box box) {
    cout << "Uzunluk: " << box.length << " cm" << endl;
}

int main()
{
    Box kutu;
    ClassTwo obj;
    kutu.setLength(8);
    printLength(kutu); //private verilere de ulaşabiliyor
    obj.greeting(kutu);
}

```

Görsel-8: Friend Functions

Not: Friend functionlar (ve classlar) normalde dışarıdan erişimi mümkün olmayan private değişkenlere erişme konusunda yararlıdır.

```

Uzunluk: 8 cm
Hello your box has a length of 8 cm

```

Görsel-9: Friend Functions Çıktısı

Yukarıda örnek olarak gösterilen kodlar haricinde for döngüleri, if/elif/else statements, switch/case yapısı, temel düzeyde exception handling, nesneye yönelik programlama açısından inheritance, encapsulation, abstract class gibi yapılar ve mantıklar örnek proje denemeleriyle irdelenmiş yukarıdaki programların bazıları içinde de kullanılmıştır. Bu konular için;

<https://www.w3schools.com/cpp/default.asp>

<https://www.tutorialspoint.com/index.htm>

temel izlenen kaynak olarak kullanılıp gerektiğinde daha derin bilgi ve öğrenme için farklı kaynaklara da yönelinmiştir.