

- 1- Ders içeriği, Kullanılacak ortam (MinGW C++, Notepad++)
- 2- Ödevler, Ödev içeriği, değerlendirme
- 3- Haftalık ders içeriği
- 4- Veri Yapıları ile ilgili genel bilgilendirme
- 5- MinGW detayı g++ exe detayı
- 6- C++ Kod yapısı, komut satırından derleme ve ifadeler
Kod G1
- 7- Komut satırı parametreleri
Kod G2
- 8- C++ Sınıf Yapısı

```
#include <iostream>
using namespace std;
class Ogrenci{
    private:
        int yas;
        double boy;
    public:
        Ogrenci(int ys,double by){
            yas=ys;
            boy=by;
        }
        Ogrenci(){
            yas=0;
            boy=30;
        }
        void Yaslarla(int ys){
            yas+=ys;
        }
        void BoyUza(double by){
            boy+=by;
        }
        int getYas(){
            return yas;
        }
        double getBoy(){
            return boy;
        }
};

int main(){
    Ogrenci o1,o2(5,135);
    cout<<"Ogrenci 1:"<<endl;
    cout<<"Yas :"<<o1.getYas()<<endl;
    cout<<"Boy :"<<o1.getBoy()<<endl;
    cout<<"Ogrenci 2:"<<endl;
    cout<<"Yas :"<<o2.getYas()<<endl;
    cout<<"Boy :"<<o2.getBoy()<<endl;
    o1.Yaslarla(15);
    o2.BoyUza(25);

    cout<<"-----"<<endl<<"Ogrenci 1:"<<endl;
    cout<<"Yas :"<<o1.getYas()<<endl;
    cout<<"Boy :"<<o1.getBoy()<<endl;
    cout<<"Ogrenci 2:"<<endl;
    cout<<"Yas :"<<o2.getYas()<<endl;
    cout<<"Boy :"<<o2.getBoy()<<endl;
    return 0;
}
```

9- Kurucu ve Yıkıcı Fonksiyonlar

```
...
    ~Ogrenci(){
        cout<<"Cagildi"<<endl;
    }
...
```

10- Başlık ve kaynak dosyaları

```
// Ogrenci.hpp
#ifndef OGRENCI_HPP
#define OGRENCI_HPP

#include <iostream>
using namespace std;
class Ogrenci{
    private:
        int yas;
        double boy;
    public:
        Ogrenci(int,double);
        Ogrenci();
        void Yasllerle(int);
        void BoyUza(double);
        int getYas();
        double getBoy();
        ~Ogrenci();
};

#endif

// Ogrenci.cpp
#include "Ogrenci.hpp"

Ogrenci::Ogrenci(int ys,double by){
    yas=ys;
    boy=by;
}
Ogrenci::Ogrenci(){
    yas=0;
    boy=30;
}
void Ogrenci::Yasllerle(int ys){
    yas+=ys;
}
void Ogrenci::BoyUza(double by){
    boy+=by;
}
int Ogrenci::getYas(){
    return yas;
}
double Ogrenci::getBoy(){
    return boy;
}
Ogrenci::~Ogrenci(){
    cout<<"Cagildi"<<endl;
}

// Deneme.cpp
#include "Ogrenci.hpp"
int main(){
    Ogrenci o1,o2(5,135);
    cout<<"Ogrenci 1:"<<endl;
```

```

        cout<<"Yas :"<<o1.getYas()<<endl;
        cout<<"Boy :"<<o1.getBoy()<<endl;
        cout<<"Ogrenci 2:"<<endl;
        cout<<"Yas :"<<o2.getYas()<<endl;
        cout<<"Boy :"<<o2.getBoy()<<endl;
        o1.YasIlerle(15);
        o2.BoyUza(25);

        cout<<"-----"<<endl<<"Ogrenci 1:"<<endl;
        cout<<"Yas :"<<o1.getYas()<<endl;
        cout<<"Boy :"<<o1.getBoy()<<endl;
        cout<<"Ogrenci 2:"<<endl;
        cout<<"Yas :"<<o2.getYas()<<endl;
        cout<<"Boy :"<<o2.getBoy()<<endl;
        return 0;
    }

```

11- Make dosyası

Önce klasör hiyerarşisinden bahset

```

hepsi: derle calistir

derle:
    g++ -I ./include/ -o ./lib/Ogrenci.o -c ./src/Ogrenci.cpp
    g++ -I ./include/ -o ./bin/deneme ./lib/Ogrenci.o ./src/deneme.cpp

calistir:
    ./bin/deneme

```

12- Şablon Sınıflar

```

//Sebze.hpp
class Sebze{
    private:
        double kalori;
    public:
        Sebze(double);
        double getKalori()const;
};

```

```

//Sebze.cpp
#ifndef SEBZE_HPP
#define SEBZE_HPP
class Sebze{
    private:
        double kalori;
    public:
        Sebze(double);
        double getKalori()const;
};
#endif

```

```

// Ogrenci.hpp Eklemler
....
#include "Sebze.hpp"
...

        double kilo;
Ogrenci(int,double,double);
...

        void YemekYe(Sebze);
        double getKilo();

```

```
//Ogrenci.cpp eklemeler
Ogrenci::Ogrenci(int ys,double by,double kl){
    yas=ys;
    boy=by;
    kilo=kl;
}

...
double Ogrenci::getKilo(){
    return kilo;
}

void Ogrenci::YemekYe(Sebze yemek){
    kilo += yemek.getKalori()/1000;
}
}
```

```
// Deneme.cpp
#include "Ogrenci.hpp"

int main(){
    Ogrenci o1(5,135,52);
    cout<<"Ogrenci :"<<endl;
    cout<<"Yas :"<<o1.getYas()<<endl;
    cout<<"Boy :"<<o1.getBoy()<<endl;
    cout<<"Kilo :"<<o1.getKilo()<<endl;
    Sebze fasulye(1200);
    o1.YemekYe(fasulye);
    cout<<"Ogrenci :"<<endl;
    cout<<"Yas :"<<o1.getYas()<<endl;
    cout<<"Boy :"<<o1.getBoy()<<endl;
    cout<<"Kilo :"<<o1.getKilo()<<endl;
    return 0;
}
}
```

Şimdi Ayrı bir Meyve Sınıfı ekleyelim

```
// Meyve.hpp
#ifndef MEYVE_HPP
#define MEYVE_HPP
class Meyve{
    private:
        double kalori;
    public:
        Meyve(double);
        double getKalori()const;
};
#endif
```

```
//Meyve.cpp
#include "Meyve.hpp"

Meyve::Meyve(double kl){
    kalori = kl;
}

double Meyve::getKalori()const{
    return kalori;
}
}
```

```
// Ogrenci.hpp eklemeler
#include "Meyve.hpp"
void YemekYe(Meyve);
```

```
// Ogrenci.cpp
void Ogrenci::YemekYe(Meyve yemek){
    kilo += yemek.getKalori()/1000;
}
}
```

Şablon Metot dönüştürme

```
// Oğrenci.hpp ye tüm öğrenci kodunu kopyala şablonda başlık kaynak ayırımı olmaz.
template <typename YIYECEK>
void YemekYe(YIYECEK yemek){
    kilo += yemek.getKalori()/1000;
}
```

```
#include "Oğrenci.hpp"
#include "Sebze.hpp"
#include "Meyve.hpp"
int main(){
    Oğrenci o1(5,135,52);

    cout<<"Oğrenci :"<<endl;
    cout<<"Yaş :"<<o1.getYas()<<endl;
    cout<<"Boy :"<<o1.getBoy()<<endl;
    cout<<"Kilo :"<<o1.getKilo()<<endl;
    Sebze fasulye(1200);
    o1.YemekYe<Sebze>(fasulye);
    Meyve kavun(2000);
    o1.YemekYe<Meyve>(kavun);
    cout<<"Oğrenci :"<<endl;
    cout<<"Yaş :"<<o1.getYas()<<endl;
    cout<<"Boy :"<<o1.getBoy()<<endl;
    cout<<"Kilo :"<<o1.getKilo()<<endl;
    return 0;
}
```

hepsi: derle çalıştır

derle:

```
g++ -I ./include/ -o ./lib/Sebze.o -c ./src/Sebze.cpp
g++ -I ./include/ -o ./lib/Meyve.o -c ./src/Meyve.cpp
g++ -I ./include/ -o ./bin/deneme ./lib/Sebze.o ./lib/Meyve.o ./src/deneme.cpp
```

çalıştır:

```
./bin/deneme
```

13- << operatörünün aşırı yüklenmesi

```
// Öğrencinin ekrana yazdırılması // Oğrenci.hpp
friend ostream& operator<<(ostream& ekran,Oğrenci& sag){
    ekran<<"Oğrenci :"<<endl;
    ekran<<"Yaş :"<<sag.yas<<endl;
    ekran<<"Boy :"<<sag.boy<<endl;
    ekran<<"Kilo :"<<sag.kilo<<endl;
    return ekran;
}
```

```
//deneme.cpp
#include "Oğrenci.hpp"
#include "Sebze.hpp"
#include "Meyve.hpp"
int main(){
    Oğrenci o1(5,135,52);
    cout<<o1;
    Sebze fasulye(1200);
    o1.YemekYe<Sebze>(fasulye);
    Meyve kavun(2000);
    o1.YemekYe<Meyve>(kavun);
    cout<<o1;
    return 0;
}
```

Karşılaşılan Hatalar ve Öneriler

Genelde ödev ve uygulamalarda karşılaşılan hatalar.

- C dili yordamı kullanmak. (C++ kullanılmalıdır.) printf, scanf terimleri, string yerine char* gibi

| | |
|---|--|
| <pre>#include <iostream> #include <cstdlib> #include "stdio.h" using namespace std; int StringToInt(const char* str){ return atoi(str); } int main(){ const char* sayi1 = "102"; const char* sayi2 = "105"; int a = StringToInt(sayi1); int b = StringToInt(sayi2); printf("sayi1 + sayi2 = %d",a+b); return 0; }</pre> | <pre>// Daha doğru kullanım #include <iostream> #include <sstream> using namespace std; int StringToInt(string str){ stringstream ss(str); int num; ss>>num; return num; } int main(){ string sayi1 = "102"; string sayi2 = "105"; int a = StringToInt(sayi1); int b = StringToInt(sayi2); cout<<"sayi1 + sayi2 = "<<a+b; return 0; }</pre> |
|---|--|

- Problemi belli bir örneğe göre çözmek, genele göre çözmek gerekir. Örneğin sistemde sadece 100 öğrenci olacak şekilde kod yazmaz. 1000 öğrenci olunca hata veriyor.
- Farklı sistemlerin anlayabildiği makefile dosyası kullanmak.
- Ödevleri son güne veya günlere bırakmak.
- Nesne yönelimli yaklaşım yerine C dili kod yazmak.