C++ HAFTA 1
DERS İÇERİĞİ ve KODLAR

Dr. Öğr. Üyesi İbrahim KÖK

C++ is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs.

Why to Learn C++

C++ is a MUST for students and working professionals to become a great Software Engineer. I will list down some of the key advantages of learning C++:

- C++ is very close to hardware, so you get a chance to work at a low level which gives you lot of control in terms of memory management, better performance and finally a robust software development.
- **C++ programming** gives you a clear understanding about Object Oriented Programming. You will understand low level implementation of polymorphism when you will implement virtual tables and virtual table pointers, or dynamic type identification.
- C++ is one of the every green programming languages and loved by millions of software developers. If you are a great C++ programmer then you will never sit without work and more importantly you will get highly paid for your work.
- C++ is the most widely used programming languages in application and system programming. So you can choose your area of interest of software development.
- C++ really teaches you the difference between compiler, linker and loader, different data types, storage classes, variable types their scopes etc.

Applications of C++ Programming

As mentioned before, C++ is one of the most widely used programming languages. It has it's presence in almost every area of software development. I'm going to list few of them here:

- **Application Software Development** - C++ programming has been used in developing almost all the major Operating Systems like Windows, Mac OSX and Linux. Apart from the operating systems, the core part of many browsers like Mozilla Firefox and Chrome have been written using C++. C++ also has been used in developing the most popular database system called MySQL.
- **Programming Languages Development** - C++ has been used extensively in developing new programming languages like C#, Java, JavaScript, Perl, UNIX's C Shell, PHP and Python, and Verilog etc.
- **Computation Programming** - C++ is the best friends of scientists because of fast speed and computational efficiencies.
- **Games Development** - C++ is extremely fast which allows programmers to do procedural programming for CPU intensive functions and provides greater control over hardware, because of which it has been widely used in development of gaming engines.
- **Embedded System** - C++ is being heavily used in developing Medical and Engineering Applications like softwares for MRI machines, high-end CAD/CAM systems etc.

Standard Libraries

Standard C++ consists of three important parts –

- The core language giving all the building blocks including variables, data types and literals, etc.
- The C++ Standard Library giving a rich set of functions manipulating files, strings, etc.
- The Standard Template Library (STL) giving a rich set of methods manipulating data structures, etc.

```
// tek satır yorum
/*
Çok satırlı yorum
*/

 #include <iostream>
//https://www.cplusplus.com/reference/iostream/
int main() {
    std::cout << "Merhaba dünya!" << std::endl;
    return 0; // main metodunu sonlandırır
    //sonrasına bir şey ekleyelim
}


//namespaceler; sınıflar, nesneler ve fonksiyonlar gibi varlıkları
//bir ad altında gruplamaya izin verir.
#include <iostream>
using namespace std;
//using std::cout;

int main() {
    cout << "Hello PAU Welcome!";
    return 0;
}
```

```
// DATA TYPES
#include <iostream>
using namespace std;

int main() {
// Basic Types in variable in C++
    cout << "Size of bool : " << sizeof(bool) << endl;
    cout << "Size of int : " << sizeof(int) << endl;
  // cout << "Size of short int : " << sizeof(short int) << endl;
  // cout << "Size of long int : " << sizeof(long int) << endl;
    cout << "Size of float : " << sizeof(float) << endl;
    cout << "Size of double : " << sizeof(double) << endl;
```

```cpp
    cout << "Size of char : " << sizeof(char) << endl;
    cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;

   //Variable Definition in C++
   //type variable_list;

    bool bv= false;
    char cv='A';
    int iv=4;
    float fv=3.12;
    double dv=4.123;
    wchar_t wv  = L'A';

    int a=3, b=5, c=7;

    cout<<"Bool variable value: "<< bv <<endl;
    cout<<"Int variable value: "<< iv <<endl;
    cout<<"Float variable value: "<< fv <<endl;
    cout<<"Double variable value: "<< dv <<endl;
    cout<<"Char variable value: "<< cv <<endl;
    cout<<"WChar variable value: "<< wv <<endl;
    cout<<"a:"<<a<<" b:"<< b <<" c:"<< c <<endl;
   return 0;
}
```

```cpp
//Aritmetik operatörler
#include <iostream>

using namespace std;

int main() {

    int v1=46;
    int v2=10;
    cout << "v1 + v2=" << v1+v2 << endl;
    cout << "v1 - v2=" << v1-v2 << endl;
    cout << "v1 * v2 =" << v1*v2 << endl;
    cout << "v1 / v2 =" << v1/v2 << endl; // float(v1) or float(v2) */

     //artırma azaltma işlemleri/operatörleri
    int v=5;
    cout<<"v'nin degeri:"<< v <<endl;
    //v=v+2;
    //v+=2;
    v--;  //v--;
    cout<<"v'nin yeni degeri:"<< v <<endl;

    return 0;
}
```

```cpp
//girdi ve çıktı işlemleri cin ve cout ile yapılır.
#include <iostream>

using namespace std;
int main() {
    int s1,s2,s3;

    cout<<" sayi 1 giriniz: ";
    cin >>s1;
    cout<<" sayi 2 giriniz: ";
    cin >>s2;
    cout<<" sayi 3 giriniz: ";
    cin >>s3;

    // alternatif olarak girdi al
    //cout<<" sayiları giriniz: ";
    //cin >> s1 >> s2 >>s3;

    cout<<" sayilar toplamı: "<< s1+s2+s3 <<endl;


    return 0;
}


int main() {
    string input1, input2;
    cout<<"input 1 giriniz";
    cin>>input1;
    cout<<"input 2 giriniz";
    cin>>input2;


    cout<<" sayi 1 giriniz: ";
    cin >>s1;
    cout<<" sayi 2 giriniz: ";
    cin >>s2;
    cout<<" sayi 3 giriniz: ";
    cin >>s3;

    // alternatif olarak girdi al
    //cout<<" sayiları giriniz: ";
    //cin >> s1 >> s2 >>s3;

    cout<<" sayilar toplamı: "<< s1+s2+s3 <<endl;

    return 0;
}
```

```
// Döngüler -- Loops
#include <iostream>
using namespace std;

int main () {
   // Local variable declaration:
   int a = 10;
   // while loop execution
   while( a < 15 ) {
      cout << "** While-- value of a: " << a << endl;
      a++;
   }

   // for loop execution
   for ( init; condition; increment ) {
    statement(s);
    }

   for( int b = 10; b < 15; b = b + 1 ) {
      cout << "--For-- value of b: " << b << endl;
   }

    // do-while loop execution
    do {
      statement(s);
    }
    while( condition ); */
    int c=10;
    do {
        cout << "* Do While- value of c: " << c << endl;
        c ++;
    } while( c < 15 );

   return 0;
}
```

```
// IF yapıları- Decision making
#include <iostream>
using namespace std;

int main () {
//****** if statement ************
/*  if(boolean_expression) {
           statement(s) will execute if the boolean expression is true
    }
*/
   int a = 20;  // local variable declaration:
   if( a < 15 ) {  // check the boolean condition
```

```
      cout << "a is less than 15;" << endl; // if condition is true then print this
line/statement
   }
   cout << "value of a is : " << a << endl;

//****** if else statement ***************
/* if(boolean_expression) {
        statement(s) will execute if the boolean expression is true
    } else {
        statement(s) will execute if the boolean expression is false
    }
*/

   int b = 100;  // local variable declaration:
   if( b < 20 ) {   // check the boolean condition
      cout << "b is less than 20;" << endl;  // if condition is true then print
this line
   } else {
      cout << "b is NOT less than 20;" << endl; // if condition is false then print
the line
   }
      cout << "value of b is : " << b << endl;



//****** if else if else  statement ***************
/*
if(boolean_expression 1) {
     Executes when the boolean expression 1 is true
} else if( boolean_expression 2) {
     Executes when the boolean expression 2 is true
} else if( boolean_expression 3) {
     Executes when the boolean expression 3 is true
} else {
     executes when the none of the above condition is true.
}
 */

   // local variable declaration:
   int c = 10;

   // check the boolean condition
   if( c == 10 ) {
      // if condition is true then print the following
      cout << "Value of c is 10" << endl;
   } else if( c == 20 ) {
      // if else if condition is true
      cout << "Value of c is 20" << endl;
   } else if( c == 30 ) {
      // if else if condition is true
      cout << "Value of c is 30" << endl;
   } else {
      // if none of the conditions is true
      cout << "Value of c is not matching" << endl;
      cout << "Exact value of c is : " << c << endl;
   }




// Taş kagıt makas -giriş
   string input1, input2;
   //cout<<"input 1 giriniz: ";
   input1="kagıt";
   //cout<<"input 2 giriniz: ";
   input2="tas";
   if( input1 == "kagıt" && input2 == "makas") {
      // if condition is true then print the following
      cout << "ikinci kazandı" << endl;
   } else if( input1 == "kagıt" && input2 == "tas" ) {
      // if else if condition is true
      cout << "birinci kazandı" << endl;
   } else if( input1 == "kagıt" && input2 == "kagıt" ) {
      // if else if condition is true
      cout << "berabere" << endl;
   //...........
   }
   return 0;
}
```

```
/*  Fonksiyonlar

return_type function_name( parameter list ) {
   body of the function
}
Return Type
-- fonksiyon değer döndürecekse tipi (int/float/double gibi)
-- değer döndürmeyecekse void olmalı
*/

#include <iostream>
using namespace std;

// function declaration
int max(int num1, int num2);
void faktoriyel(int sayi);

int main () {
   int a,b;     // local variable declaration:
```

```
   //int a = 100;
   //int b = 200;
   cout<<"iki sayi giriniz: ";
   cin >> a >> b;

   //int ret;
   int ret = max(a, b);   // calling a function to get max value.
   cout << "Max value is : " << ret << endl;
   faktoriyel(ret);

   //int f=faktoriyel(ret);
   //cout << "Max value is : " << f << endl;
   return 0;
}

int max(int num1, int num2) {
   // local variable declaration
   int maximum;

   if (num1 > num2)
      maximum = num1;
   else
      maximum = num2;

   return maximum;
   //cout << "Beni ekranda görebiliyormusun ☺ "<< endl;
}


void faktoriyel (int sayi){
    int faktoriyel= 1;
    for (int i = 2 ; i <= sayi; i++)
    {
        faktoriyel=faktoriyel*i;
        //faktoriyel*=i;
    }
    cout<<"girilen sayinin faktoriyeli:" << faktoriyel<<endl;
}
```

```
// Pointers
#include <iostream>
using namespace std;

// pointerlar adres tutar * işareti ile başlar, adres ataması için & kullanılır
int main () {
    int a=5;   // int 4 byte   s 5000 5001 5002 5003
    int *ptr;
    ptr=&a;

    cout<<"a değişkeninin adresi- a: " << a <<endl;
    cout<<"a değişkeninin adresi- &a: " << &a  <<endl;
    cout<<"a değişkeninin adresi ptr: " << ptr <<endl;
    cout<<"a değişkeninin ptr üzerinden değeri *ptr: " << *ptr <<endl;

    *ptr=10;
    cout<<"a değişkeninin YENİ DEĞERİ " << a <<endl;
   return 0;
}
#include <iostream>
using namespace std;
int main () {

    //type arrayName [ arraySize ];     -- array tanımlama
    int arr [4] ={5,6,7,8};

    cout<< arr[1] <<endl;
    cout<< arr[3] <<endl;
    cout<< arr <<endl;   //başlangıç adresi taşıyan adress değeri verir
    cout<< "array adresi "<< arr +1 <<endl;
 //   cout<< sizeof(arr)/sizeof(arr[0]) <<endl;   //sizeof byte cinsinden değer
verir

   int *ptr =arr;         // 2. indexin adresini ptr ye atama istersek &arr[2] olur
   ptr =ptr + 1;
   cout<< "pointerin yeni adresi"<<ptr <<endl;
   return 0;
}
```

```
/* STL Standart Template Library- standart şablon kütüphaneleri

C Standart Kitaplığı, bir dizi C yerleşik işlev, sabit ve <stdio.h>, <stdlib.h>,
<math.h> vb. Başlık dosyalarıdır.
Bu kitaplık, C programcıları için bir başvuru kılavuzu olarak çalışacaktır. .

1  Containers
Containers are used to manage collections of objects of a certain kind.
Belirli türden nesne koleksiyonlarını yönetmek için kullanılır.
There are several different types of containers like deque, list, vector, map etc.

2  Iterators
Iterators are used to step through the elements of collections of objects.
İterators, nesne koleksiyonlarının öğeleri arasında gezinmek için kullanılır.

3  Algorithms
Algoritmalar container üzerinde işlem yapmamızı sağlar.
Bilgileri değiştirme, yerleştirme, sıralama, arama ve dönüştürme gibi veri
işlemlerini gerçekleştirirler.*/
```

```
#include <iostream>
using namespace std;

template <typename T>
T kare (T sayi){
return sayi * sayi;
}

//aşağıdaki fonksiyonları tanımlamak yerine template tanımla
/* int kare (int sayi){
return sayi * sayi;
}

double kare (double sayi){
return sayi * sayi;
}

float kare (float sayi){
return sayi * sayi;
}
 */
int main () {
   cout<< kare(7.688)<<endl; //call kare with int, float or double value
   return 0;
}
```

```
// vector=dinamik olarak artırılıp azaltılan dinamik kontainer
#include <iostream>
#include <vector>
using namespace std;

void yazdir(vector<int> v) {
 // use iterator to access the values
    vector<int>::iterator iter = v.begin();
    while (iter != v.end())
    {
        cout << *iter++<<endl;
    }

}

void yazdir1(vector<int> v) {
 // use iterator to access the values
    for (auto iter = v.begin(); iter != v.end(); ++iter)
    {
    cout << *iter << " ";
    }
    cout << endl;
}
```

```
void yazdir2(vector<int> v) {
    for (int i = 0; i < v.size(); ++i)
    {
    cout << v[i] << " ";
    }
    cout << endl;
}


int main () {

    // constructing vectors
    vector <int> sayilar;
    //vector <int> sayilar1 (5);
   // vector <int> sayilar2 (5,10);
   // vector <int> sayilar3 (sayilar1.begin(),sayilar1.end());


    sayilar.push_back(3);
    sayilar.push_back(5);
    sayilar.push_back(7);
    sayilar.push_back(9);
    //sayilar.insert(2,200);
    sayilar.insert(sayilar.begin()+2,15);
    sayilar.erase(sayilar.begin(), sayilar.begin()+2);
    //yazdir2(sayilar);

   // sayilar.insert(sayilar.begin()+2, 13);
    yazdir2(sayilar);
  //  cout<< "vektör ilk elemanı: " <<sayilar.front()<<endl;
  //  cout<< "vektör son elemanı: " <<sayilar.back()<<endl;
  //  cout<< "0. index elemanı:" <<sayilar.at(0)<<endl;
    cout<< "vektör boyutu: " <<sayilar.size()<<endl;
  //  yazdir2(sayilar1);
  //  yazdir2(sayilar2);
  //  yazdir2(sayilar3);
   return 0;
}
```