BÖLÜM 4

C++ YAZILIM GELİŞTİRME ORTAMLARI

ANAHTAR KAVRAMLAR

Derleyici (Compiler), Hata Denetimi, Çalıştırılabilir (Executable) Dosyalar



ÖĞRENME HEDEFLE<mark>Rİ</mark>





BC++ genel amaçlı bir programlama dilidir. C++, C programlama dilinin bir çok

özelliğini içermekle birlikte, C programla dilinden farklı olarak kullanıcıya nesne yönelimli (object-oriented) programlama özellikleri sunar.

Bu bölümde C++ programlarının nasıl ve nerede yazılacağı anlatılmaktadır. Yazılan programın, kullanılan derleyici üzerinde derlenip, hata denetiminin yapıldıktan sonra nasıl çalıştırılacağı gösterilmektedir.





1. C++ YAZILIM GELİŞTİRME ORTAMLARI

C++ genel amaçlı bir programlama dilidir ve Windows, Linux, Unix, Mac vb. birçok platformda çalışır. C++ genel amaçlı programlama özelliklerinin yanısıra nesne yönelimli (object-oriented) programlama özelliklerine de sahiptir.

C++ ile programlamaya başlamadan önce, yazdığımız programları derlemek (compile) ve çalıştırmak (run) için, kullanmış olduğumuz bilgisayarın ve işletimin sisteminin özelliklerine uygun bir yazılım geliştirme platformunu (IDe) veya derleyicinin bilgisayarlarımıza kurulması gerekmektedir. Eğer yerel (bireysel) bilgisayarınıza bir C++ yazılım geliştirme platformu kurmak istemiyorsanız çevrimiçi olarak yazmış olduğunuz kodları derleyen ve çalıştırmanıza izin veren ortamlarda mevcuttur.

Bu bölümde bir C++ programını geliştirmek, derlemek ve çalıştırmak için ihtiyaç duyulan ortamlar tanıtılacak ve ilk C++ programı yazılarak izlenmesi gereken adımlar anlatılacaktır.

2. C++ YAZILIM GELİŞTİRME ORTAMININ KİŞİSEL BİLGİSAYARINIZA KURULMASI

Bu bölümde öncelikle Linux ortamında yazılım geliştirme ortamının düzenlenmesi gösterilecek ardından Windows üzerinde kullanılacak olan Cygwin derleyicisinin kurulumu gösterilecektir. Linux işletim sistemi için C++ programlarını geliştirebilecekleri ortamın oluşturulması aşağıdaki gibi yapılabilir.

2.1. Linux Ortamında GCC Kurulumu

Linux işletim sistemi kullanıcıları aşağıdaki komutları kullanarak GCC (GNU Compiler Collection) derleyicisini bilgisayarlarına kurabilirler.

Kurulum işlemi için öncelikle bilgisayarınızdan terminal ekranını açınız ve sırası ile aşağıdaki komutları giriniz.

sudo apt-get update

sudo apt-get install gcc

sudo apt-get install g++

Yukarıda verilen komutların sırayla çalıştırılması bilgisayarınıza GCC derleyicisini kurulmasını sağlayacaktır.

Sisteminizden kaynaklı olarak bazı durumlarda ek olarak aşağıdaki komutun kullanılması gerekebilir.

sudo apt-get install build-essential

Bu komut bir C++ programının derlenmesi ve çalıştırılması için gerekli kütüphanelerin yüklenmesini sağlayacaktır¹.

Yukarıdaki işlemlerin tamamlanmasının ardından bir metin (text) dosyasına C++ kodları yazılır ve aşağıdaki örnekte gösterildiği gibi bu dosyanın uzantısı cpp olarak belirlenir.

ilkprogram.cpp

Daha sonra yazılan programın derlenip çalıştırılması için aşağıdaki adımlar izlenir

• Bir Linux terminali açarak programını oluşturulduğu klasör içerisine gidilir.

• Terminal ekranında g++ ilkprogram.cpp yazılarak programın derlenmesi sağlanır. Bu komut .cpp uzantılı dosyanız ile aynı isimde .exe uzantılı çalıştırılabilir bir dosya oluştur. Çalıştırılabilir dosyanıza başka bir isim vermek isteniyorsa ko-

¹ Kurulum esnasında herhangi bir hata oluşmadığınızdan emin olunuz. Bunun için g++ --version komutunu kullanabilirsiniz.



mut **g++ ilkprogram.cpp -o** yeniisim şeklinde değiştirilebilir. Bu durumda **yeniisim.exe** isimli bir dosya oluşturulur.

• Bir önceki işlem adımından sonra herhangi bir hata alınmaz ise klasör içerisinde ilkprogram.exe ya da yeniisim.exe adında yeni bir dosya oluşturulur. Bu dosya çalıştırılabilir bir dosyadır. • Yazılan programı exe uzantılı dosyanın ismini kullanarak ./ilkprogram veya ./yeniisim şeklinde bir komut kullanarak çalıştırabilir ve sonucu yine terminal ekranında izlenebilir.

Not: Yukarıdaki yönergelerin detaylı anlatımı video kaydında bulunabilir.

2.2. Windows Ortamında C++ Derleyicisinin Kurulumu

Windows ortamında C++ programlarını yazmak, derlemek ve çalıştırmak için bir-çok farklı IDE veya derleyici kullanılması mümkündür. C++ programlama dilinin evrensel yapısı göz önünde bulundurularak bu kitapta Cygwin derleyicisi kullanılacaktır. Cygwin, Windows'ta Linux dağıtımına benzer işlevsellik sağlayan bir GNU ve açık kaynak araçlarının kullanımına olanak sağlayan bir ortam olarak ifade edilebilir (Cygwin, 2022)².



Şekil 4.1. Cygwin terminal ekranı ve klasör yapısı

Kurulum işlemi sonrasında terminal ekranını ve kurulum klasörü açıldığında Şekil.4.1'dekine benzer bir yapı görülür. Terminal ekranında klasörler arasında geçiş işlemleri için temel Linux/ DoS komutları kullanılacaktır. Terminal ekranı ilk açıldığında kurulum kök klasörünü gösteren Şekil.4.2'dekine benzer bir adres ekranda gözükecektir.



Şekil 4.2. Terminal ekranı ilk açılış görüntüsü

2 Cygwin kurulum dosyalarına ve yönergelerine https://www.cygwin.com/ adresinden ulaşılabilir. Kurulum ile ilgili önerilen videoya ise https://www.youtube.com/watch?v=hh-V6el8Oxk adresinden ulaşılabilir.



Terminal ekranına cd .. komutu yazılarak kurulum klasörünün içerisinde bulunan home klasörünün içerisine geçilebilir. Not: cd komutu klasörler arasında geçiş için kullanılır. İngilizce change directory kelimelerinin baş harflerinden oluşturulmuş bir komuttur.



Şekil 4.3. Cygwin terminali home klasörüne geçiş

Bu kitap örnekleri için home klasörünün içeresinde ders adında bir alt klasör daha oluşturulmuş ve kod dosyaları bu klasör içerisinde oluşturulmuştur. Terminal ekranından bu klasöre erişim için home klasörü içerisinde iken cd ders yazılmalıdır.



Şekil 4.4. Cygwin terminali üzerinden klasörler arası geçişler

Bu klasör içerisinde oluşturulacak C++ programlarını derlemek ve çalıştırmak için gerekli dosyalar tutulacaktır. Windows ekranından aşağıda gösterildiği gibi/home/ders/ klasörünün içerisine geçerek **ilk program** için ilkprogram adında bir metin dosyası oluşturun ve dosyanın uzantısını **cpp (c plus plus)** olarak belirleyin.



Şekil 4.5. İlk program için cpp uzantılı dosyanın oluşturulması

Oluşturulan dosya bir metin dosyasıdır. Dosyayı açıp Örnek.4.1 kodlarını dosya içerisine yazınız. Örnek.1. ilkprogram.cpp #include<iostream> using namespace std; int main() { cout<<"Ankara Üniversitesine Hoş Geldiniz"; return 0; }



Kodların Analizi:

• #include: Yazılan programa standart veya kullanıcı tanımlı bir dosya ekle-mek için kullanılan komuttur ve çoğunlukla bir C++ programının başında kullanılır.

• <iostream>: C++ programları için temel girdi-çıktı (input-output) servislerini aktif hale getirir.

#include<iostream> komutunun birlikte kullanılması, yazılım içerisindeki input-output servislerinin kullanılması için gerekli header (kütüphane) dosyasını programa dahil eder. input-output işlemleri dışında başka header dosyalarına ihtiyaç duyularsa bunlarında include komutu ile birlikte programın başında tanımlamasının yapılması gerekmektedir. Örneğin yazılan programda karekök alma, logaritma hesaplama veya yuvarlama işlemlerinin yapılması isteniyorsa bir matematik kütüphanesini prog-rama eklenmesi gerekmektedir.

Ör. #include<cmath>

• using namespace std: Bu komut ile std adında bir alan adının (namespace) programda kullanılacağı belirtilmektedir. Std standart kelimesinin kısaltılmasıdır. Eğer programda cout, cin gibi input-output işlemleri için gerekli fonksiyonların kullanılması gerekiyorsa, programın bu işlerin nasıl yapılacağını bilmesi gerekmektedir. Std alan adı bu fonksiyonların tanımlandığı bir alandır. Eğer başlangıçta bu tanımlama yapılmazsa, kodlar yazılırken std::cout, std::cin, std::endl gibi her bir fonksiyon öncesi std komutunun :: ile birlikte kullanılması gerekir.

• int main(): Derleyicinin ilk bağlantı kurduğu bölümdür. Her programın bir main fonksiyonu olmak zorundadır. Bu örnekte integer (tamsayı) türünde değer döndüren bir main fonksiyonu tanımlaması verilmiştir. Void gibi türlerin tanımlanması da mümkündür (void main). Main fonksiyon adından sonra açılıp kapatılan parantezler '(', ')' içerisinde fonksiyonun aldığı parametreler tanımlanır. Fonksiyonlar bölümünde bu konu daha detaylı olarak ele alınacaktır.

• { } : main fonksiyonundan hemen sonra açılan ve programın sonunda kapa-tılan küme parantezleri ise fonksiyonun etkili olduğu alanı (scope) tanımlar. Küme parantezleri içerisinde yazılan kodlar sadece ilgili fonksiyonun kullanımı içindir.

• cout<<: cout fonksiyonu ekrana bir şey yazdırmak için kullanılır. Yazdırılan ifade değişkenin içerisinde tutulan değer ya da bir string ifade olabilir. cout komutundan hemen sonra kullanılan <<operatörleri, aşırı yüklenmiş (over-loaded) stream (akış) operatörleri olarak ifade edilir. Programa girdi olarak bir değer alınacaksa cin>> şeklinde kullanımı vardır. Bu operatörler programlamada bitwise shift operatörleri olarak bilinir. leftshift (sola kaydır, <<) ve rigtshift (sağa kaydır, >>) olarak tanımlanmıştır. Ancak input-output işlemleri için overload edilmiş bu fonksiyonlar cout ve cin komutları ile birlikte kullanılırken std::ostrem (output akışı), std::istream (input akışı) işlemleri için ayrı bir görev ile yeniden tanımlanmışlardır (Bkz. Operatör overloa-ding).

• return 0: Normalde bütün fonksiyonlar geriye bir değer döndürür. Program return 0 komut satırını gördüğünde program sorunsuz bir şekilde tamamlanmış demektir. C++ da return 0 komutunun kullanılması isteğe bağlı olmakla birlikte C programlama dilinde kullanımı bir zorunluluktur. Profesyonel yazılımcılar return tipi olarak farklı değerler kullanabilirler. Bu konu yine fonksiyonlar bölümünde detaylı olarak ele alınacaktır.

• ; (Noktalı Virgül) : Programda bazı satırların sonunda ; karakterinin kullanıldığına dikkat ediniz. C++' da komut satırları ; karakteri ile sonlandırılmak zorundadır. Aksi halde derleme zamanında hata alınır.



• " ": cout komutunu takip eden "Ankara Üniversitesine Hoş Geldiniz" ifadesinin "…" işaretleri içerisinde yazıldığına dikkat ediniz. Eğer bir metni (string ifade) ekrana yazdırmak istiyorsanız bu tırnak işaretlerini kullanmak zorundasınız. Şimdi yazdığımız programı derleyelim ve çalıştıralım. Kullanılan Cygwin derleyicisi bizlere Linux benzeri bir ortam sunduğu için Linux ortamı başlığı altında bahsedilen program derleme ve çalıştırma adımları bu bölüm içinde geçerlidir.

Öncelikle daha önce bahsedildiği gibi Cygwin terminal ekranını açınız ve programı yazdığınız klasörü işaret ediniz (Ör. .../cd).

E /home/ders	2 <u>111</u>	×
Lenovo@LAPTOP-HTJKC5LI ~ \$ cd		
Lenovo@LAPTOP-HTJKCSLI /home \$ cd ders		
Lenovo@LAPTOP-HTJKC5LI /home/ders \$ ls ilkprogram.cpp		
Lenovo@LAPTOP-HTJKC5LI /home/ders \$		

Şekil 4.6. Terminal ekranından klasör içerisine göz atma

Şekil 4.6'nın 3. satırında kullanılan ls (list files) komutu bulunduğunuz klasör altındaki dosyaları görüntülemek için kullanılır. Burada ders klasörünün altında metin dosyası içerisine yazılmış olan ilkprogram.cpp uzantılı dosya görüntülenmektedir. Bu dosyanın çalıştırılabilir (executable) bir dosya haline getirmek için derlenmesi gerekmektedir. Şimdi aşağıdaki adımları terminal ekranında sırası ile yazarak programınızı derleyiniz ve çalıştırınız.

• g++ ilkprogram.cpp //Bu komut yazdığınız programı derleyecek ve her-hangi bir hata alınmaması durumunda çalıştığınız klasör altında a.exe uzantılı yeni bir dosya oluşturacaktır. Dosyanın oluşup oluşmadığını ter-minal ekranından ls komutunu kullanarak veya Windows ekranlarından mevcut klasörün altına giderek kontrol edebilirsiniz.

• Eğer herhangi bir hata almadan exe uzantılı dosya oluşturuldu ise bu sefer terminal ekranına ./a.exe komutunu yazarak programınızı çalıştırabilirsiniz.

Derleme esnasında bir sözdizimi (syntax) hatası alınmadıysa ve program yazılırken mantıksal (logical) bir hata yapılmadıysa ./a.exe komutunun kullanımı ile yazılan program çalıştırılır ve cout komutu ile ekrana yazılması istenilen ifade ekranda görüntülenir.

Çalıştırılabilir dosyanın adı derleyici tarafından varsayılan (default) olarak a.exe olarak belirlenir. Aynı klasör içerisinde tutulan farklı programlar derlendiğinde a.exe uzantılı dosya en son derlenen programa ait olacaktır. Programları tekrar tekrar derlemek yerine varsayılan isim yerine özel bir isim kullanılarak programlar kendilerine verilen özel isimler ile oluşturulan exe uzantılı dosyalar üzerinden çalıştırılabilir. Derlenecek olan exe uzantılı dosya için özel bir isim kullanılmak istenirse derleme komutu:

• g++ ilkprogram.cpp -o yeniisim

şeklinde yazılabilir. Bu komut ilkprogram.cpp dosyasından yeniisim.exe adında çalıştırılabilir bir dosya oluşturur (Sizler istediğiniz herhangi bir ismi verebilirsiniz).

Yukarıda bahsedilen tüm işlem adımları aşağıdaki Şekil.4.7'de verilmiştir.





Şekil 4.7. Terminal ekranından bir programın derlenmesi ve çalıştırılması

3. SÖZDİZİMİ HATA DENETİMİ

Yazılan programın derlenmesi esnasında karşılaşılabilecek sözdizimi (syntax) hataları bu bölümde ele alınmıştır. Program geliştirme esnasında sözdizimi hatalarının yanısıra mantıksal hataların yapılması da olasıdır. Bir örnek oluşturması açısından problem analizinde ya da algoritma geliştirirken toplama işlemi yapılması gereken bir yerde çıkarma işleminin yapılması mantıksal bir hata meydana getirecektir.



Şekil 4.8. Program geliştirme adımları

Programda yapılacak mantıksal hatalar, hatanın yapıldığı bölüme bağlı olarak problem analizi kısmına kadar yapılan işleri gözden geçirmeyi gerektirebilir. Sözdizimi hataları ise kod yazımı esnasında yapılan hataları ifade eder. Bu hatalar kodlama ekranında düzeltilip program yeniden derlenip çalıştırılabilir.

Bir sözdizimi hatası için Örnek.4.2.a'daki kodları inceleyin: Örnek.4.2.a. program2.cpp #include<iostream> using namespace std; int main() { cout<<"Ankara Üniversitesine Hoş Geldiniz" return 0;



Programın 4. satırında cout ile verilen ifadenin sonunda ";" özellikle kaldırılmıştır. Program bu şekilde kaydedilip derlendiğinde aşağıdaki ekran görüntüsündeki gibi bir hata ile karşılaşılır.



Şekil 4.9. Sözdizimi (syntax) hatası

Şekil 4.9'da görüldüğü gibi program2.cpp dosyası program2.exe dosyasını oluşturacak şekilde derlenmeye çalışılmıştır. Ancak derleme esnasında programın 4. satırında return komutundan hemen önce olası bir ";" ifadesinin kullanımının unutulduğu ile ilgili bir uyarı ekranda görüntülenmiştir. Aynı zamanda bu bölüm bir ok işareti ile gösterilmiştir. Sözdizimi hataları, derleyici tarafından kolaylıkla tespit edilebilir ve programcıyı sözdizimi hataları konusunda uyarır. Şimdi Örnek.4.2.b'de cout kullanımında "<<" (stream out) operatörünü ">>" (stream in) operatörü ile değiştirin.

Örnek.4.2.b. program2.cpp #include<iostream> using namespace std; int main() { cout>>"Ankara Üniversitesine Hoş Geldiniz"; return 0;



Şekil 4.10. Derleme zamanı hatası

Programın derlenmesi sırasında, terminal ekranında Şekil 4.10'dan görüldüğü gibi oldukça uzun bir hata mesajı ile karşılaşılır. Daha önce belirtildiği gibi, "<<" ve ">>" operatörleri aşırı yüklenmiş operatörlerdir. Bu operatörler cout ya da cin gibi komutlar ile kullanılırken kendi görevleri dışında (bitshift) stream akışını (out ya da in) kontrol etmek üzere programlanmışlardır (aşırı yüklenmişlerdir, overload edilmişlerdir). Hata ekranında iki bölümde gösterildiği gibi (normal ekranda daha fazlası gözükecektir) std::ostream (stream out) sınıfından ">>" operatörünün türetilemediğine dair bir hata mesajı çıkmaktadır. Buradaki hata, bir operatörün görevi dışında başka bir amaçla kullanılmasından kaynaklanmaktadır.

Şimdi, iostream header dosyasının (kütüphane) programa dahil edilmediği Örnek.4.2.c'yi inceleyiniz. Örnekte iostream dışında kalan bölümler düzgün olarak verilmiştir.

Örnek.4.2.c. program2.cpp using namespace std; int main() { cout<<"Ankara Üniversitesine Hoş Geldiniz"; return 0; }



Şekil 4.11. Gerekli header dosyası yüklenmediğinde karşılaşılan hata

Terminal ekranından görüleceği üzere, gerekli header dosyası programa eklenmediğinde yine derleme zamanı hatası ile karşılaşılır. Hata mesajında main fonksiyonu içerisinde kullanılan cout ifadesinin bildirilmediği ve cout komutu-

nun iostream header dosyası içerisinde olduğu ifade edilmektedir. Sonuç olarak iostream header dosyasının programa dahil edilmediği için bir derleme zamanı hatası oluştuğu belirtilmektedir.

4. TERMİNAL EKRANINDAN VERİ GİRİŞİ

Daha önceki örneklerde cout komutunu kullanarak ekrana string bir ifadenin nasıl yazdırılacağı gösterildi. Programa terminal ekranından kullanıcının veri girişini nasıl yapacağı bu bölümde gösterilecektir. İlgili klasör içerisinde oluşturulacak yeni program içerisine aşağıdaki kodları yazınız.

Örnek.4.3. program3.cpp //Bu satır açıklama satırıdır derleyici tarafından derlenmez #include<iostream>

```
using namespace std;
```

```
int main() {
int a;
int b, c;
cout<<"İlk Sayıyı Giriniz:";
cin>>a;
cout << "İkinci Sayıyı Giriniz:\n";
cin>>b;
c = a + b;
cout<<"Toplam="<<c<endl;
return 0;
```

```
}
```



Kodların Analizi:

• Programın 1. Satırında verilen "//" ifadesi program yazılırken açıklama satırları eklenmek istenirse kullanılır. Tek satırlık açıklamalar için kullanılır. Eğer birden fazla satırı açıklama satırı haline getirmek istiyorsanız "/**/" şeklinde kullanılır. Program tarafından işlenmez.

• Programın 5. Satırında integer (tamsayı) türünde a adında bir değişken tanımlaması yapılmıştır. Değişken türleri ile ilgili daha detaylı bilgi için 5. bölümü inceleyiniz.

• Programın 6. satırında yine integer türünde b ve c adında iki değişken tanımlaması yapılmıştır. 5. Satırdan farklı olarak burada bir satırda 2 değişken birlikte tanımlanmıştır. 5. Satırda olduğu gibi bir değişken tanımlamasından sonra ";" kullanımı ile o satır sonlandırılabileceği gibi, değişkenler arasına "," işareti konularak birden fazla değişken tanımlanması mümkündür.

• 7. satırda programı kullanan kişiye bir sayı girmesi gerektiğini söyleyen bir mesaj oluşturulmuştur. 9. satırda yazılan mesajın sonunda "\n" ifadesinin kullanıldığına dikkat ediniz. "\n" ifadesi programlamada kullanılan escape (kaçış) ifadelerinden bir tanesidir ve yeni satır (new line) anlamına gelir. Bu karakter kullanıldığında mesaj ekrana yazıldıktan sonra sanki enter tuşuna basılmış gibi terminal ekranındaki imleç yeni ekranda görüntülenir. 7 ve 9. Satırlarda kullanılan iki kullanımda doğrudur ve sadece kullanım farklılıklarını göstermek için farklı şekillerde gösterilmiştir.

• 8 ve 10. satırda verilen cin komutu kullanıcının terminal ekranından programa bir değer girmesine olanak tanır. istream (input stream) komutu için ">>" aşırı yüklenmiş operatörleri ile birlikte kullanılır. Program çalıştırılırken sıra bu satıra gelince denetim terminal ekranına geçer ve kullanıcının bir değer girmesi beklenir. Sonrasında klavyeden enter tuşuna basılarak denetim yeniden derleyiciye bırakılır, bu esnada kullanıcı tarafından girilen değerler a ve b değişkenleri içerisine atanır.

• 11. satırda kullanıcı tarafından a ve b değişkenleri için girilen iki sayı toplanır ve sonuç c değişkenine atanır.

• 12. satırda öncelikle ekrana "Toplam=" ifadesi yazdırılır daha sonra bu ifadeye birleşik olacak şekilde c değişkeni içerisindeki değer ekrana yazdırılır ve son olarak endl (end line) komutu yazma işlemi sonrası programın yeni bir satıra geçmesi sağlanır. endl komutu "\n" komutu ile aynı işi yapar. Bu satırda, cout komutunun "<<" operatörlerinin tekrarlı kullanımı ile birden fazla işlemi aynı anda gerçekleştirildiğine dikkat ediniz.



Şekil 4 12. Terminal ekranından veri giriş örneği



BÖLÜM ÖZETİ

Bu bölümde temel veri türleri, değişkenler, atama ve aritmetik işleçlerden bahsedilmiştir. Bir bilgisayar programı hafızada sakladığı çeşitli verileri işleyerek birtakım sonuçlar üretir. Birçok farklı alanda programlara ihtiyacımız olduğu düşünülünce çok farklı veri türlerine gereksinim duyulmaktadır. C++ programlama dilinde tanımlı üç farklı temel veri türü vardır. Bu türler sayısal, boolean ve karakter veri türleridir. Sayısal veri türleri kendi içinde tam ve kesirli sayılar olmak üzere ikiye ayrılmaktadır. Bir bilgisayar programı programcının tanımladığı değişkenleri işleyerek istenilen sonuca ulaşır. Değişkenlere uygun isim vermek yazılan kodun okunabilirliğini artırır. Çünkü yazılım yaşam döngüsünde bulunan süreçlerin gereği olarak bir programcının yazmış olduğu kodu başka programcılarında kullanması veya test etmesi gerekmektedir. Ayrıca C++ programlama dilinde bir değişkene isim vermek için bazı kurallar vardır. Değişkenlere bu kurallara uygun isimler vermek derleme hatalarının önüne geçecektir. Tanımlı değişkenler aritmetik, ilişkisel veya mantıksal işleçler içerisinde kullanılarak daha karmaşık ifadelerin yazılması mümkün hale gelmektedir. İşleçler konusu oldukça geniş bir kavramdır. Bu bölümde atama ve aritmetik işleçlerden bahsedildi. Atama işleci bir değişkene bir değer atamak için kullanılır. Buna ek olarak toplama, çıkarma, bölme, çarpma ve kalan gibi aritmetik işleçler anlatılmıştır. Bir ifade içerisinde birden fazla işleç kullanarak daha karmaşık ifadelerin elde edilmesi mümkündür. Bu durumda hangi işlecin önce hesaplanması gerektiği işlem öncelik kurallarına göre belirlenir. Bu kuralları bilmek program içerisinde yapılabilecek hataları en aza indirgeyecektir.

GÖZDEN GEÇİRELİM

- Aşağıdakilerden hangisi C++ programları için dosya uzantısı olarak kullanılır?
 a) c
 - b) cpp
 - c) c++
 - d) ls
 - e) cd

- 4. Ornek.exe uzantılı dosyayı çalıştırmak için aşağıdaki komutlardan hangisi kullanılmalıdır?
 - a) ./Ornek.cpp
 - b) Ornek.cpp
 - c) Ornek.exe
 - d) ~Ornek.exe
 - e) ./Ornek.exe

2. Bir program yazılırken komut satırının sonuna ";" eklenmesinin unutulması nasıl bir hata oluşmasına neden olur?

- a) Sözdizim (Snytax) Hatası
- b) Mantıksal (Logical) Hata
- c) Kütüphane hatası
- d) Alan adı hatası
- e) Hata oluşmaz

- 5. Bir C++ programına iostream header'ı eklenmemişse aşağıdaki sorunlardan hangisi ile karşılaşılır?
 - a) Değişken tanımlaması yapılamaz
 - b) Main fonksiyonu hata verir
 - c) Aşırı yükleme hatası alınır
 - d) input-output işlemlerinde hata alınır
 - e) Mantık hatası verir

3. Ornek.cpp isimli bir dosyanın Ornek. exe şeklinde derlenmesi için gerekli komut aşağıdakilerden hangisidir?

- a) g++ Ornek.cpp -o
- b) g++ Ornek.cpp
- c) g++ Ornek.cpp -o Ornek
- d) Ornek.exe -o g++ Ornek.cpp
- e) g++ Ornek.exe

- 6. Ornek.cpp isimli dosya g++ Ornek.cpp yazarak derlenirse oluşturulacak çalıştırılabilir dosyanın adı ve uzantısı aşağıdakilerden hangisi olur?
 - a) Ornek.cpp
 - b) Ornek.exe
 - c) a.exe
 - d) a.cpp
 - e) Hata verir

 7. cout komutu ile ilgili aşağıdakilerden hangisi(leri) söylenebilir? I. Çıkış akışını (output-stream) kontrol eder II. Giriş akışını (input-stream) kontrol eder III. Sağa kaydır operatörüdür a) Yalnız I 	 9. ls (List Files) komutu ne için kullanılır? A) Dosya isimlerini değiştirmek için B) Dosya oluşturmak için C) Klasör isimlerini değiştirmek için D) Klasörleri silmek için E) Bulunulan klasör içerisindeki dosyaları ve diğer klasörleri listelemek için
 b) Yalnız II c) Yalnız III d) II ve II e) Hiçbiri 	 10. endl komutu ile ilgili olarak aşağıdaki- lerden hangisi(leri) söylenebilir? I. Program bu komutu görünce bir alt sa- tırdan devam eder II. /n komutu ile benzer göreyi gercek-
 8. cd (Change Directory) komutu ne için kullanılır? a) Klasör ismini değiştirmek için 	leştirir III. Programı sonlandırır IV. Hata denetimi için kullanılır
 b) Klasörler arası geçiş için c) Klasör silmek için d) Klasörleri yeniden isimlendirmek için e) Dosya oluşturmak için 	 a) Yalnız I b) Yalnız II c) Yalnız III d) I ve II e) III ve IV

Yanıt Anahtarı: 1-B, 2-A, 3-C, 4-E, 5-D, 6-C, 7-A, 8-B, 9-E, 10-D

Sözlük

Header file

Program başında kullanılması gereken, program içerisindeki komutların bulunduğu alan adı.

Compiler (Derleyici):

Yazılan programların derlenip denetiminin yapıldığı ortam

Exe (Executable)

Çalıştırılabilir dosyaların uzantısı

cpp: C++ programlarının yazıldığı dosyanın uzantısı

syntax (Sözdizimi)

Program geliştirilirken kullanılan dile ait özellikler (Örneğin, ;, küme parantezleri vb)

overload (aşırı yükleme)

Bir operatörün normal kullanım amacı dışında farklı görevler içinde yüklenmesi



cd (change directory)

Terminal ekranında klasörler arası geçiş komutu

ls (list-file)

Terminal ekranından bir klasörün içerisinde bulunan alt klasör ve dosyaları listeleme komutu

g++

Program derleme komutu

Kaynakça

(2022, Ocak 22). Cygwin: https://www.cygwin.com/ adresinden alındı

