

1. VERİ YAPILARINA GİRİŞ

Giriş

Veri (Data), Enformasyon (Information) ve Bilgi aslında insanların tarih sahnesine çıktığı ilk günden beri insan hayatında önemli bir yer tutmuş ve özellikle bilgisayarların insan hayatına girmesi ile çok daha önemli hale gelmiştir. Günümüzde bilgisayar ve iletişim teknolojilerindeki gelişmeler her alandaki verinin çok hızlı şekilde toplanabilmesi, işlenebilmesi sonucunu doğurmuş, gelinen noktada bu değerler para ve benzeri diğer araçlardan daha önemli hale gelmiştir. Yirmi Birinci Yüzyılda bireyler, işletmeler hatta ülkeler arasındaki rekabetin en güçlü aracı doğru biçimde organize edilmiş, ihtiyaç duyulduğunda hızlı bir şekilde erişilebilen, enformasyona ve bilgiye dönüştürülebilen, elektronik ortamlarda saklanan güncel verinin miktarıdır. Dolayısıyla bu kıymetli değer (verinin) elektronik ortamda nasıl organize edildiği, ihtiyaç duyulduğunda hızlı bir şekilde erişilebilir olması gibi konular oldukça önemlidir.

Verinin elektronik ortamda nasıl organize edildiği, bellekte tutulma biçimi, sıralaması, hızlı bir şekilde erişilebilir olması vb. konular doğrudan **Veri Yapıları (Data Structure)** dersinin konusudur. Bu nedenle Veri Yapıları dersinde, kullanılan programlama dillerinin desteklediği önemli veri yapıları sırasıyla ele alınarak, dönem sonunda öğrencilerin, elektronik ortamlarda veri manipülasyonu yaparken doğru veri yapısını seçme düzeyine ulaşmaları hedeflenmiştir.

Bu bölümde, yukarıdaki parafıta sözü edilen veri yapılarına doğrudan giriş yapmak yerine önce, çoğu zaman hatalı şekilde kullandığımız Veri (Data), Enformasyon (Information) ve Bilgi (Knowledge) kavramları açıklanacak, sonra Veri Tipleri ele alınacak, Daha sonra Veri Yapıları Dersinde kullanılacak C programlama dili ve Entegre Geliştirme Ortamları (IDE) hakkında bilgiler aktarılacaktır.

Bu dönem Veri Yapıları dersi ANSI C programlama dili kullanılarak işlenecektir. ANSI C programlama dili C ailesinin yapısal programlama dili kategorisinde yer alır ve birçok özelliği (Nesneye Yönelik yaklaşımlar hariç) bakımından C++ ile aynıdır. Ayrıca, bu dersi alan öğrencilerin bir dönem önce Programlamaya Giriş ve Algoritmalar dersinde C++ öğrendikleri, veri yapıları dersinde orta düzey bir C kullanımına ihtiyaç duyulacağı ve konular işlenirken gereksiz ayrıntılarda kaçınılacağı göz önünde bulundurulursa öğrencilerin derse çalışırken zorlanmayacağı düşünülmektedir.

Dersin işlenişi sırasında C programlama dilinin seçilmiş olmasının temel amacı, C dilinin özellikle sistem programları, işletim sistemleri, derleyiciler gibi aşağı seviyeli sistem programlarının yazılımında hala yoğun olarak kullanılması ve bundan dolayı öğrencilerimizin orta seviyede de olsa bu aracı öğrenmesinin kariyerleri için önemli olacağı düşüncesidir.

Veri Yapıları dersinde derleyici ve editör olarak geçen yıl Algoritmalar ve Programlamaya Giriş Dersinde kullanılan DEV C++ kullanılacaktır. DEV C++, C ve C++ programlarının yazılması ve derlenmesi için kolay, yaygın olarak kullanılan bir Tümüleşik Geliştirme Ortamı (Integrated Development Environment) dır.

Bu bölümün son iki konusu olarak, DEV C++ ve C programlama ele alınacak ve Veri yapıları dersinde kullanılacak programlama ortamı ile programlama dili hakkında öğrenciler gerekli bilgiler verilecektir.

1.1. Veri (Data) Enformasyon (Information) ve Bilgi (Knowledge)

1.1.1. Veri

Veri, işlendiğinde veya düzenlendiğinde anlamlı çıktı oluşturan, en düşük soyut veya işlememiş ham girdiyi (**raw data**) ifade eder. Genel olarak veriler olguların, kavramların ve talimatların insan tarafında veya otomatik şekilde iletişim, yorumlama ve işleme amacına yönelik ifadesidir. Veriler gerçeklere, gözlemlere, algı sayılarına dayanır ve sayılar, karakterler, karakter dizileri, semboller, görüntülerle ifade edilir.

Veriler genellikle kayıtlar veya gözlemler sonucunda elde edilir. Örneğin, günlerin sıcaklığı veridir ve bu veriler toplanacağı zaman, bir sistem veya kişi günlük sıcaklıkları izler ve kaydeder. Bilgisayar bilimi açısından verilerin anlam ifade edebilmesi için elektronik ortamda işlenmesi gerekir. Son olarak, kısaca veri için yapılandırılmamış gerçekler ve rakamlardır diyebiliriz.

Üniversitelerin Öğrenci İşleri otomasyonunu veritabanındaki öğrencilerin adlarını, soyadlarını, doğum yerleri ve tarihlerini, öğrencilerin her birinin Veri Yapıları dersinden aldığı notları, öğrencilerin bölüme giriş puanlarını, mezun oldukları lise türlerini veriye örnek gösterebiliriz. Veriye ikinci bir örnek verecek olursak bir işletmenin veya işletmenin rakibinin satış rakamları verileri oluşturur.

1.1.2. Enformasyon

Enformasyon, verinin anlamlı bir konu etrafında işlenmiş halidir. Başka bir ifadeyle veriler işlendiğinde enformasyona dönüşür. Kısacası veriden enformasyona ulaşılabilmesi için verilerin ilişkilendirilmesi, kategorize edilmesi, özetlenmesi ve hesaplanması gerekir.

Veriler konusunda üniversite öğrenci işleri otomasyonundan örnek vermiştik. Aynı uygulamadan elde edilen Öğrenci Listesi, Öğrencilerin Veri Yapıları dersine ait harf notları, sınıfın ders ortalamalarını gösteren liste enformasyona örnek gösterilebilir. Veri için verilen ikinci örnekten devam edecek olursak, enformasyona örnek olarak işletmelerin satış raporlarını ve grafiklerini örnek gösterebiliriz.

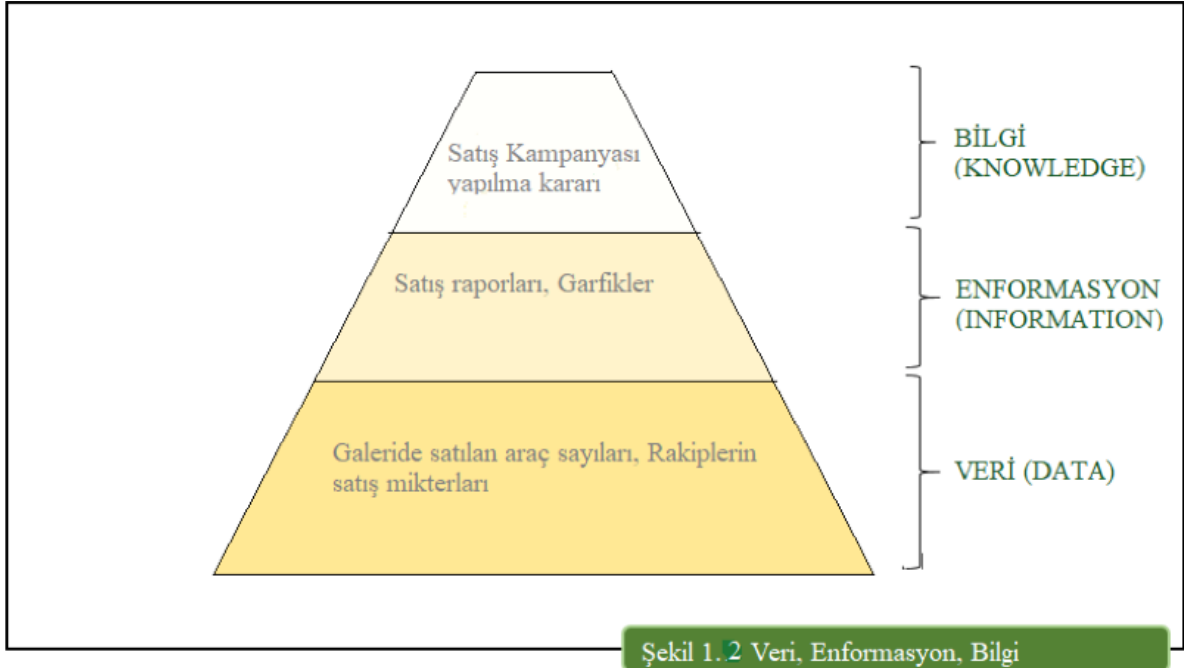
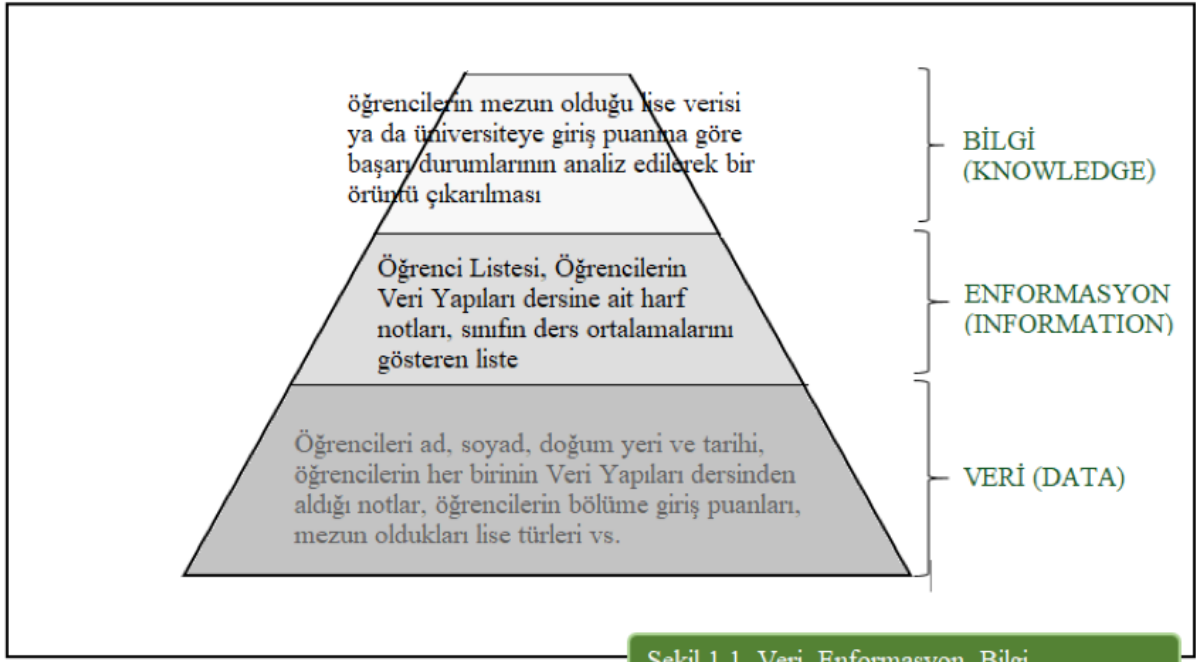
1.1.3. Bilgi (Knowledge)

Webster Sözlüğüne göre bilgi, "deneyim veya ilişkilendirme yoluyla kazanılan aşinalıkla bir şeyi bilmenin olgusu veya koşulu" dur. Bilginin değişik kaynaklarda birçok olası, eşit derecede makul tanımı vardır.

Bilginin sık kullanılan bir tanımı da, "kuruluşların hedef(ler) ine ulaşmak amacıyla etkili eylemde bulunmak için kullanabilecekleri fikirleri veya anlayışlarıdır. Bu bilgi, onu ortaya koyan varlığa özgüdür." şeklindedir.

Bir başka tanımla; bilgi, enformasyon parçaları ile bunlardan yararlanarak ne yapılabileceği arasındaki ilişkiyi anlama yeteneğidir.

Veriler ve enformasyon konusunda üniversitelerin öğrenci işleri otomasyonundan örnek vermiştik. Aynı şekilde öğrencilerin mezun olduğu lise verisi ya da üniversiteye giriş puanına göre başarı durumlarının analiz edilerek bir örüntü çıkarılmasını da bilgi (Knowledge) için örnek verebiliriz. Yukarıda veri ve enformasyon için verilen ikinci örnekten devam edecek olursak “Satış Kampanyası Yapılması Kararı” bilgidir. Aşağıda bu iki örnek şekil 1.1 ve Şekil 1.2’de gösterilmiştir.



1.2. Veri Tipleri

Yukarıda veri başlığı (1.1.1 **Veri**) altında belirtildiği gibi, bilgisayar ortamında veriler sayılar, karakterler, karakter dizileri ve mantıksal olarak ifade edilir, bellekte kapladıkları alana ve sakladıkları verinin tipine göre sınıflandırılırlar. Aşağıda sizeof() fonksiyonu kullanılarak verilerin bellekte kapladıkları alanı ve bu alana yazılabilecek minimum ve maksimum değerleri ekrana yazdıran program (Program 1.1)

verilmiştir. Bu program veri yapıları dersinde kullanılacak olan C programlama dilinde yazılmıştır.

```
#include <stdio.h>
#include <math.h>
#include <locale.h>
int main() {
    setlocale(LC_ALL,"Turkish");
    printf("\n TİP ADI          UZUNLUK      ALT SINIR      ÜST SINIR\n");
    printf("-----\n");
    printf(" char                %d bit  -%3.0f      %3.0f \n",sizeof(char)*8 ,pow(2,7),pow(2,7)-1);
    printf(" unsigned char       %d bit  %3.0f      %3.0f\n",sizeof(unsigned char)*8,0,pow(2,8));
    printf(" int                 %d bit  -%7.0f      %7.0f\n",sizeof(int)*8,pow(2,31),pow(2,31)-1);
    printf(" unsigned int        %d bit  %3.0f      %7.0f\n",sizeof(unsigned int)*8,0,pow(2,32));
    printf(" long int            %d bit  -%7.0f      %7.0f \n",sizeof(long int)*8,pow(2,31),pow(2,31)-1);
    printf(" unsigned long int   %d bit  %3.0f      %7.0f\n",sizeof(unsigned long int)*8,0,pow(2,32));
    printf(" short int           %d bit  -%5.0f      %7.0f \n",sizeof(short int)*8,pow(2,15),pow(2,15)-1);
    printf(" unsigned short int  %d bit  %3.0f      %7.0f\n",sizeof(unsigned short int)*8,0,pow(2,16));
    printf(" float               %d bit  \n",sizeof(float)*8);
    printf(" double              %d bit  \n",sizeof(double)*8);
    printf("-----\n");
    return 0;
}
```

Program 1.1

TİP ADI	UZUNLUK	ALT SINIR	ÜST SINIR
char	8 bit	-128	127
unsigned char	8 bit	0	256
int	32 bit	-2147483648	2147483647
unsigned int	32 bit	0	4294967296
long int	32 bit	-2147483648	2147483647
unsigned long int	32 bit	0	4294967296
short int	16 bit	-32768	32767
unsigned short int	16 bit	0	65536
float	32 bit	3.4×10^{-38}	3.4×10^{38}
double	64 bit	1.7×10^{-308}	3.4×10^{308}

Program 1.1'in çıktısı

Not: float ve double tipindeki verilerin alt ve üst sınırları tabloya elle eklenmiştir.

Yukarıda verilen tabloda yer alan veri tipleri C programlama diline ait temel/ilkel (primitive) veri tipleridir. Yeri gelmişken burada belirtmeliyiz ki, C programlama dilinde sabit ve değişkenlerin tanımlanması ile ilgili kurallar C++ programlama dilindeki kurallarla uyumludur. C

programlama dilinde olduđu gibi diğerk programlama dillerinde de benzer şekilde veri tipleri vardır. Bütün programlama dillerinde program içerisinde verilerin klavyeden girişinin yapılabilmesi, belleğe yazılması, manipule edilmesi ve bellekten okunması gibi işlemler için tanımlanan sabitler ve değişkenler bu temel veri tiplerinden yararlanılarak tanımlanır.

Program 1.1'in çıktısında gösterilen temel veri tipleri, çođu zaman program geliştirirken ihtiyaçlara cevap vermez. Zaman zaman performansı iyileştirmek, zaman da verilerin erişilebilirlik düzeylerini arttırmak gibi nedenlerle programcılar, yine kullandıkları dilin desteklediğı değişik yaklaşımları kullanır.

Bu yaklaşımlardan ilki temel veri tiplerinden yararlanılarak oluşturulan Basit(Simple) veri yapılarıdır. Bunlar genellikle birbirleri ile ilişkili verileri manipule etmek amacıyla kullanılır. Cprogramlama dilinde, Basit (simle) veri yapılarına dizi (array), yapı (struct), Karakter katarı (string) ve topluluk (union) örnek gösterilebilir. Bir önceki dönemde aldığımız Algoritmalar ve Programlamaya Giriş dersinde C++ kullanarak yukarıda sayılan dizi, yapı, karakter katarı konuları ele alınmış ve her biri birden çok örnekte kullanılmıştır. Bu saydığımız basit veri yapılarının kullanılış biçimleri küçük bazı değişiklikler hariç C programlama dilinde de C++ ile aynıdır. Bu farklar yeri geldikçe bu derste açıklanacaktır.

Burada belirtmeliyiz ki verilerin bellekte tutulma biçimleri bir programdan beklenen başarımla doğrudan doğruya ilişkilidir. Ne yazık ki bazen hem temel veri yapıları hem de yukarıda sayılan basit veri yapıları bu ihtiyacı karşılamaz. Bu nedenle Birleşik (Compound) Veri Yapıları tanımlanmıştır ki Veri Yapıları dersinin asıl konusu daha önce karşılaşmadığımız bu Birleşik Veri Yapılarıdır. Bağlı Listeler (Linked List), Yığınlar (Stack), Kuyruklar (Queue), Ağaçlar (Tree), Graflar Birleşik Veri Yapılarına örnek olarak gösterilebilir. Şekil 1.3'de temel veri tipleri, basit veri tipleri ve Birleşik veri tipleri bir şema üzerinde gösterilmiştir.



1.3. DEV C++

Bu ve bundan sonraki bölümlerde, veri yapıları dersinde yazılacak örnekleri derlemek ve çalıştırmak için DEV C++ derleyicisini kullanabilirsiniz. Derleyiciler, sizin yazacağınız kaynak kodları çalıştırılabilir bir dosya haline getirir. Dev C++ herhangi bir düzenleme yapılmadan kurulduğunda standart derleyici olarak GNU Compiler Collection (GCC) kullanır. Veri yapıları dersinde yazılacak örnek C programlarının da bu derleyicide derlenmesini bekliyoruz.

C'de kaynak dosyalar, .c uzantılı metin dosyaları, çalıştırılabilir dosyalar ise, .exe uzantılıdır. Aynı zamanda Dev C++ size kaynak dosyalarınızı oluşturabilmeniz için bir editör, hata ayıklayıcı ve benzeri birçok özellik sunar. Program geliştiricilere bu şekilde birçok kolaylık sağlayan yazılımlara **Tümleşik Geliştirme Ortamı** (*Integrated Development Environment- IDE*) denir.

C programlarını geliştirmek için piyasada ücretli veya ücretsiz birçok tümleşik geliştirme ortamı vardır. Örneğin Code Blocks, Turbo C bunlardan birkaçıdır. Biz bu derste örnek uygulamalarımızı Dev C++

ortamında geliştireceğiz. Bu IDE 'yi seçmemizin nedeni, bu geliştirme ortamının sadeliği ve kolay olmasıdır.

1.3.1. DEV C++ 'ın Kurulumu

İlk geliştirilmesi Bloodshed firması tarafından yapılan DEV C++, 2011 yılından sonra Orwell firmasına geçmiş ve bundan sonraki geliştirmeler bu firma tarafından yapılmıştır. İnternet üzerinden indirip bilgisayarımıza kurabileceğiniz DEV C++'ı <https://sourceforge.net/projects/orwelldvcpp/> adresinden indirebilirsiniz.

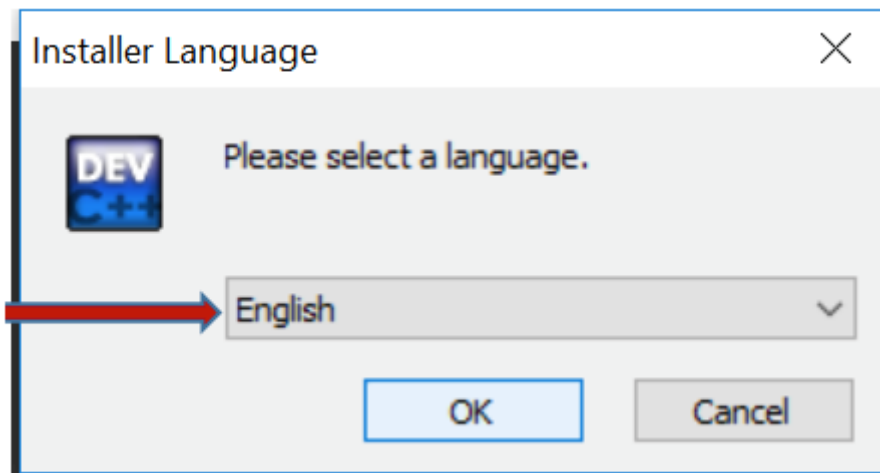
Dev C++ 'ı bilgisayarınıza kurmak için aşağıdaki adımları takip ediniz:

1. Önce DEV

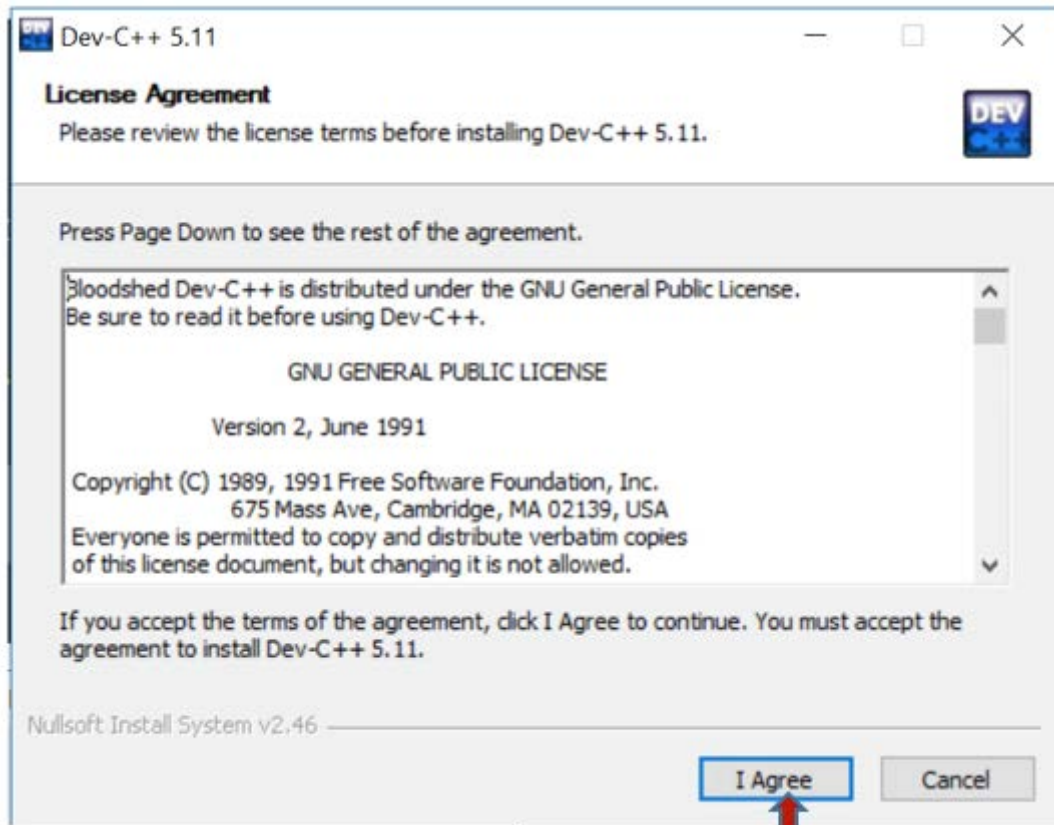
C++'ı <https://sourceforge.net/projects/orwelldvcpp/> adresinden bilgisayarınıza indiriniz.



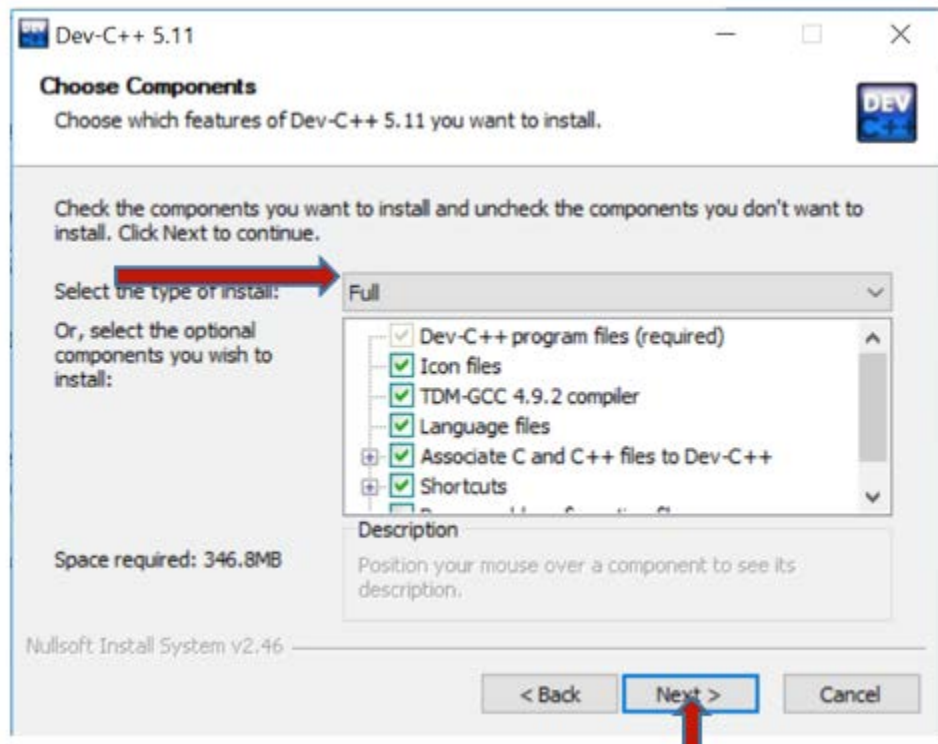
2. İndirilen dosyayı çalıştırın ve aşağıdaki pencereden dil seçimini yapınız.



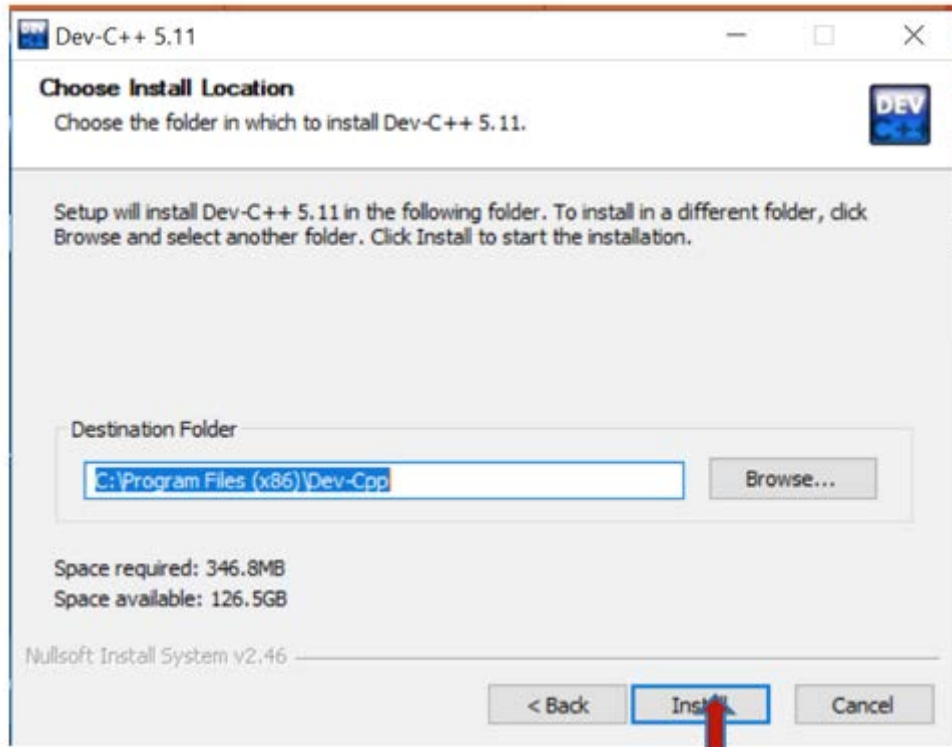
3. Sözleşmeyi okuyup onaylayınız.



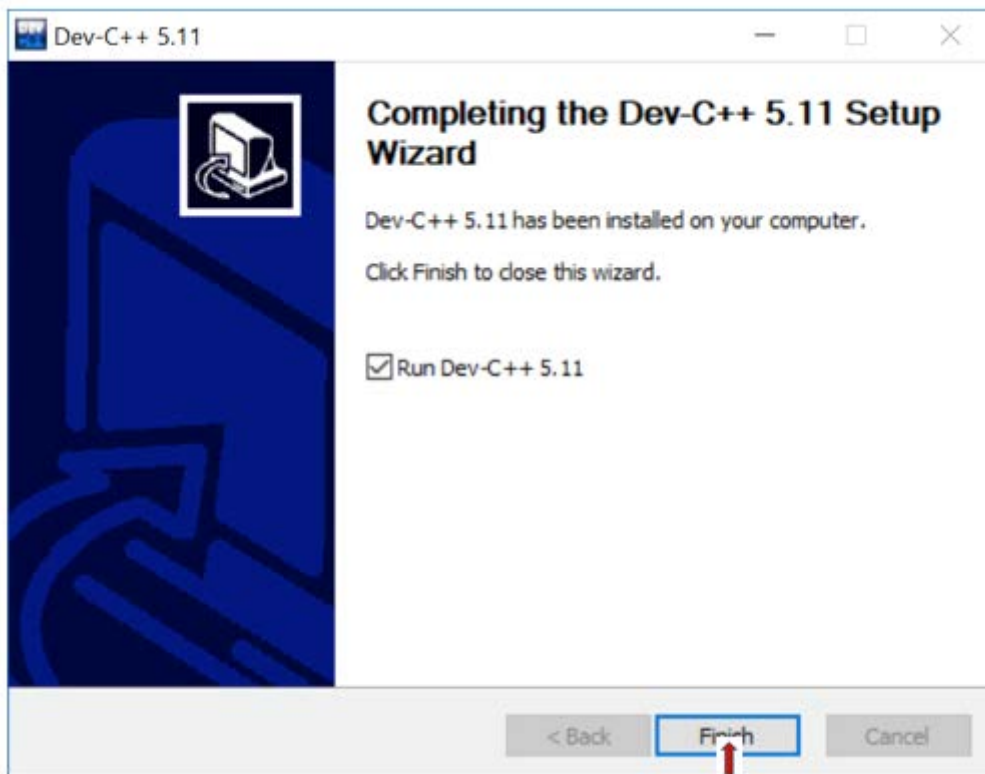
4. Kurulum tipini **Full** seçiniz ve **ileri** butonunu tıklayınız.



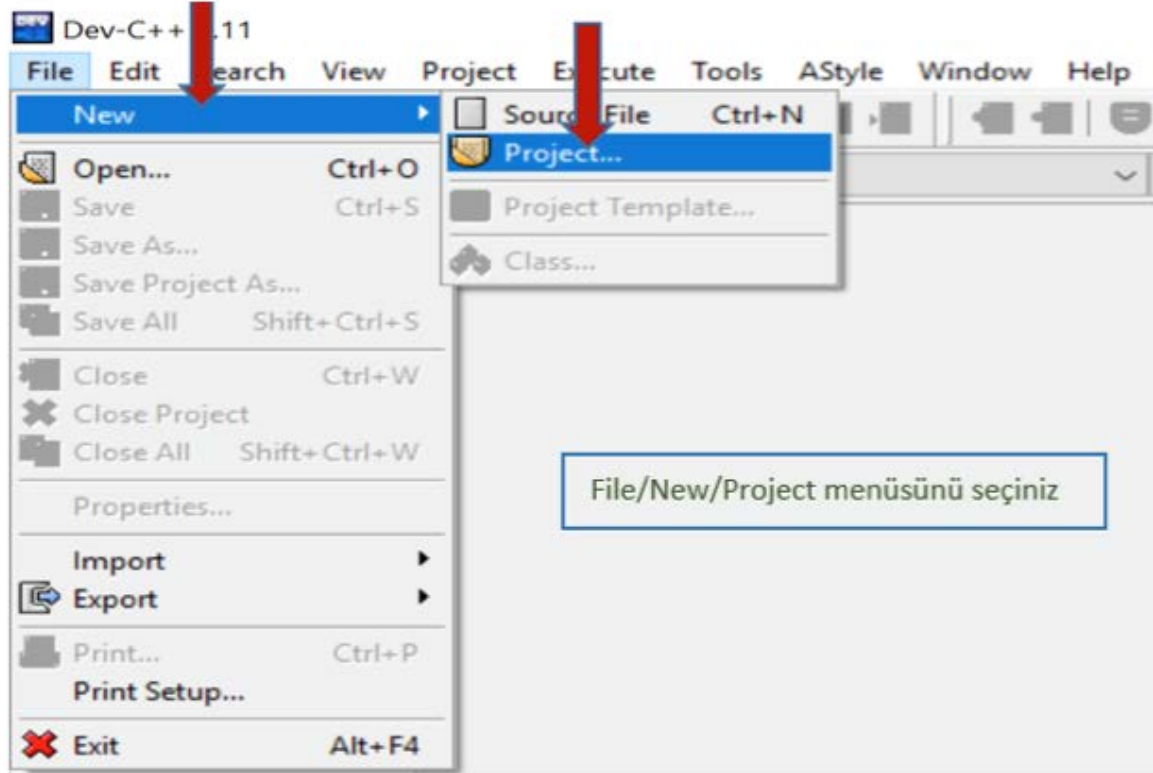
5. Hedef klasör seçiminde bir deęişiklik yapmadan **Install** butonuna tıklayınız.

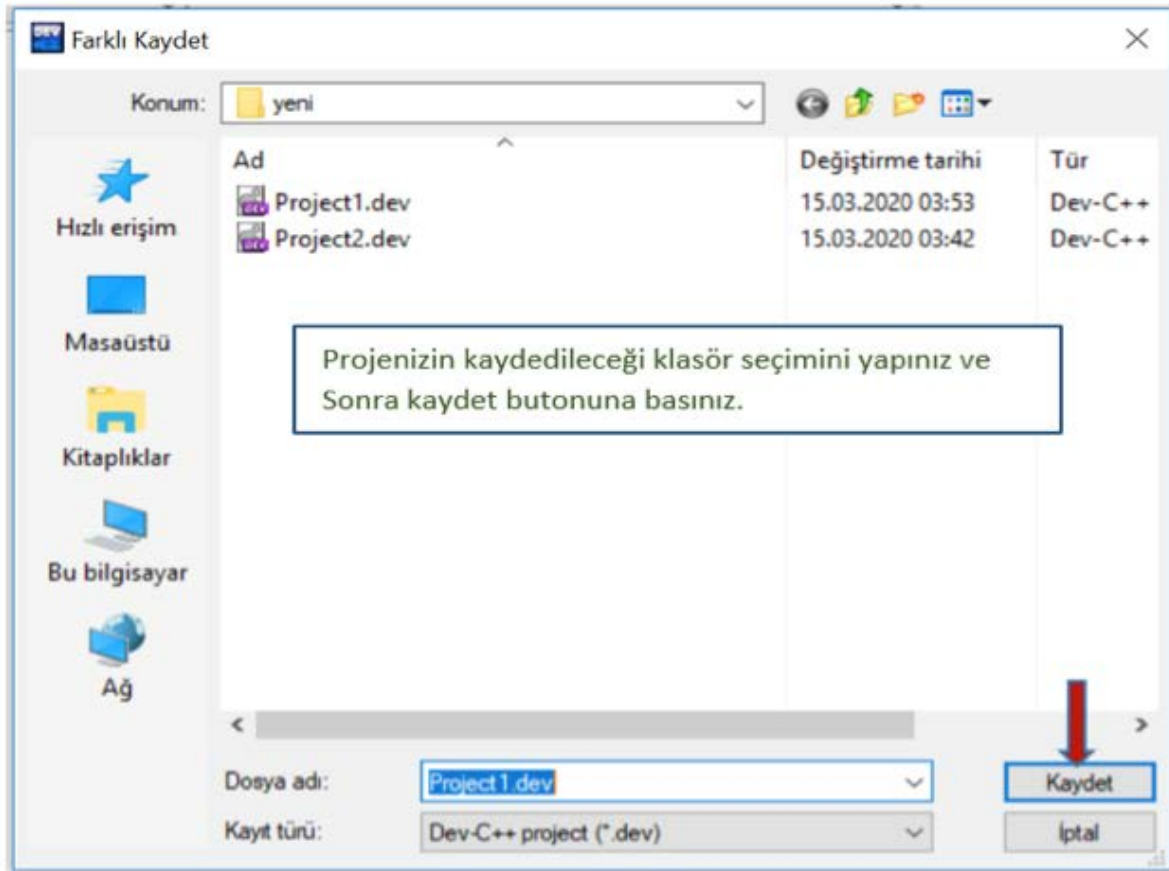
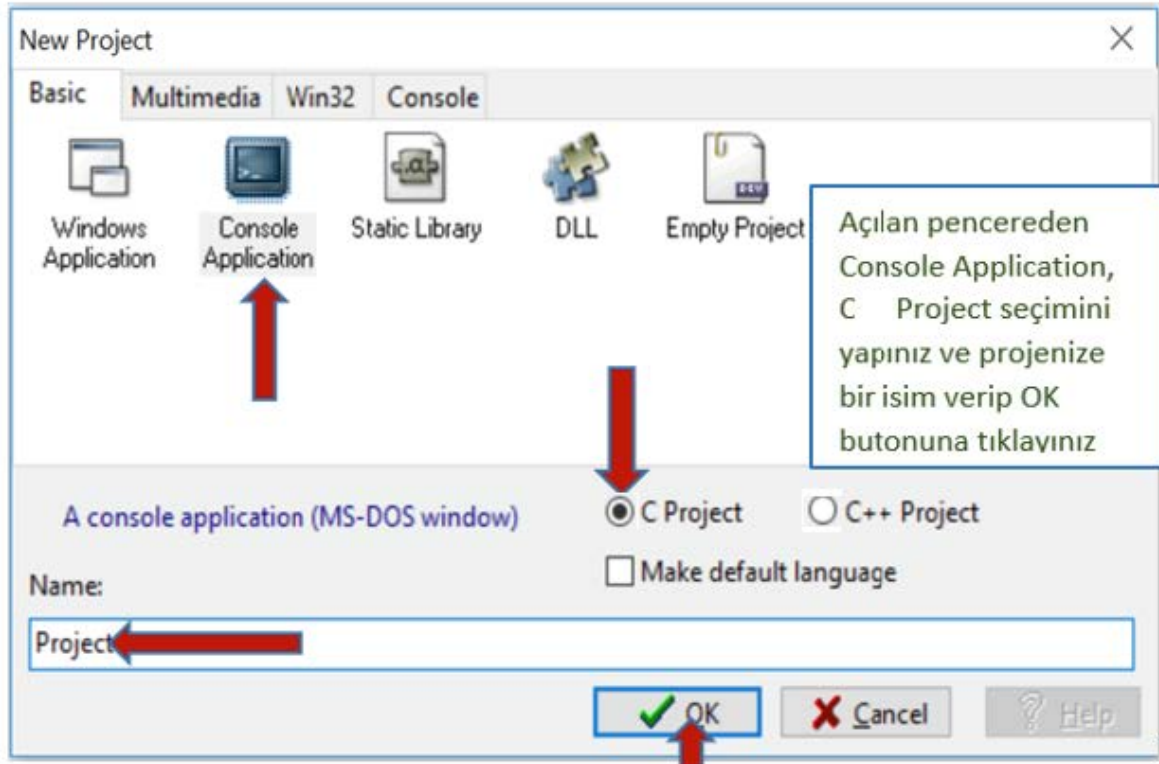


6. Kurulum tamamlandıktan sonra gelen pencereden **Finish** butonuna tıklayınız.



1.3.2. DEV C++ 'ın Kullanımı





1.4. C Programlama Dili

C programlama Dili 1972 yılında Dennis Ritchie tarafından Bell laboratuvarlarında Unix işletim sistemi ile kullanılmak üzere tasarlanmıştır. Genel amaçlı orta seviye bir dildir. Özellikle UNIX işletim sisteminin kullanımının yaygınlaşması, dilin sistem programlamaya uygun olması gibi nedenlerle kısa zamanda popüler olmuştur. Günümüzde hala sistem programlama konusu gündeme geldiğinde C programlama dili en önemli, hatta vazgeçilmez bir seçenektir. Bugün, C Programla Dili için American National Standards Institute (ANSI) kurumunun Mart 2000'de belirlediği C99: ISO/IEC 9899:1999 standardı Standart C olarak kabul edilmektedir.

C Programlama Dili, standart olması, taşınabilir olması, yapısal bir dil özelliğine sahip olması, kendisinden sonra çıkan birçok dilin C'den etkilenmiş olması, Özel komut ve veri tipi tanımlanmasına izin vermesi, işletim sistemi, derleyici, kelime işlemci yazılmasında kullanılması gibi nedenlerle günümüzde hala güçlü bir diller arasında yer almaktadır. Bu nedenle bilgisayar teknolojileri, özellikle yazılım konusu ile ilgili olanların C konusunda bilgi sahibi olmasında önemli faydalar vardır.

1.4.1. Temel Giriş – Çıkış Fonksiyonları

1.4.1.1. printf() Fonksiyonu

```
1  /* Bu program C'de yazdığım
2     ilk programdır */
3
4  #include <stdio.h>
5
6  int main() {
7      printf("ilk C programı\n");
8      return 0;
9  }
```

Program 1.2

Program 1.2 derlenip çalıştırıldığında ekran “ilk C programı” yazar. Programda birinci ve ikinci satır arasında;


```
/* Bu program C'de yazdığım  
ilk programdır */
```

yazılmıştır. Bu satırlar derleyici tarafından değerlendirilmeyen açıklama satırlarıdır. C'de, C++ benzer şekilde tek satırlık açıklamalar, açıklama olarak değerlendirilecek satırın başına // yazılarak yapılır. Eğer açıklamanın birden fazla satırda yapılması isteniyorsa o zaman yapılacak açıklama yukarıda verilen örnekte olduğu gibi /* */ arasında yapılmalıdır. C programlarında açıklama satırı yazmak için kullanılan bu işaretlere **açıklama operatörü** denir.

Program 1.2'de dördüncü satırda **#include <stdio.h>** ifadesi yazılmıştır. Bu ifadedeki **#include** deyimini programa eklenecek başlık dosyasını işaret eder. Programda işaret edilen başlık dosyası (header file) **stdio.h** 'tır.

C++ programlarında olduğu gibi C programları da fonksiyonlar halinde yazılır ve her C programında main() fonksiyonu yer alır. Programın çalışmaya bu fonksiyondan başlar.

Programda yedinci satırda **printf()** fonksiyonu yazılmıştır. **printf()** fonksiyonu C programa dilinde ekrana formatlı çıktı yazdırmak (Bu işlem için C++ programlarında **cout** kullanılmaktaydı) için kullanılır. C programları içerisinde printf() fonksiyonunu kullanabilmek için başta programa stdio.h başlık dosyasının eklenmiş olması gerekmektedir.

C programlarında printf() fonksiyonu sadece ekrana açıklama yazdırmak için değil değişkenlerin değerlerini yazdırmak için de kullanılır.

C programlarında printf() fonksiyonunun kullanım şekilleri aşağıda verilmiştir:

1. Ekrana düz metin, etiket yazdırmak

printf("**düz metin**\n"); ifadesi yazılır.

Metnin sonunda yer alan \n kontrol karakteri (escape squence) ‘dir. Tablo 1.1’de C’de kullanılan kontrol karakterleri verilmiştir.

Kontrol Karakteri	Anlamı
\n	Yeni satır
\t	Yatay TAB
\v	Dikey TAB
\r	Satır Başı
\f	Sayfa atla
\b	İmleci bir karakter sola kaydır (backspace)
\a	Ses üretir
\"	Çift tırnak karakterini ekrana yaz
\'	Tek tırnak karakterini ekrana yaz
\\	\ karakterini ekrana yaz
\%	% karakterini ekrana yaz

Tablo 1.1 C’de kullanılan Kontrol Karakterleri

2. Tip belirleyici (conversion specifier): % işareti ile başlar ve bir veya iki karakterden oluşur (%d gibi). Ekrana yazdırılmak istenen değişkenin tipi, % işaretinden sonra belirtilir. C programlama dilinde kullanılan tip belirleyiciler Tablo 1.2’de verilmiştir.

Tip Karakteri	Anlamı	Yazdırılacak Veri Tipi
%d	İşaretli tamsayı	İnt, short
%ld	İşaretli uzun tamsayı	long
%f	Tek duyarlılıklı gerçel sayı	float
%lf	Çift duyarlılıklı gerçel sayı	double
%u	İşaretsiz tam sayı	unsigned int, unsigned short
%lu	İşaretsiz uzun tam sayı	unsigned long
%c	Karakter	char
%s	Karakter Dizisi (string)	char

Tablo 1.2 Tip belirleyiciler

1.4.1.2. scanf() Fonksiyonu

```

1  #include <stdio.h>
2  #include<locale.h>
3  int main() {
4      setlocale(LC_ALL,"Turkish");
5      int sayil; float sayi2;
6      printf("Birinci Sayıyı Giriniz : \n");
7      scanf("%d",&sayil);
8      printf("İkinci Sayıyı Giriniz : \n");
9      scanf("%f",&sayi2);
10     printf(" \nGirilen Birinci sayı : %d İkinci Sayı : %3.2f\n",sayil,sayi2);
11     return 0;
12 }
```

Program 1.3

Program 1.3 ‘te scanf() fonksiyonunun kullanımı gösterilmiştir. scanf() fonksiyonu klavyeden veri okumak için kullanılan fonksiyondur. printf()

fonksiyonu gibi scanf() fonksiyonu da Tablo 1.2 'de verilen karakterleri kullanır. Aşağıda scanf() fonksiyonunun kullanımına iki örnek verilmiştir.

```
scanf ("%d", &sayi1) ;
```

```
scanf ("%f", &sayi2) ;
```

Burada **&** işareti adres operatörü olarak adlandırılır. Klavyeden iki farklı sayı okunmak istendiğinde scanf() fonksiyonu şöyle kullanılabilir:

```
scanf ("%d %f", &sayi1, &sayi2) ;
```

1.4.1.3. puts() ve gets() Fonksiyonları

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  int main() {
5      setlocale(LC_ALL, "Turkish");
6      char s1[10];
7      char s2[10];
8
9      //Ad Soyad Giriliyor
10     puts("Öğrencinin Adı ");
11     gets(s1);
12     puts("Öğrencinin Soyadı");
13     gets(s2);
14
15     //değişkenlerin değerleri ekrana yazdırılıyor.
16     puts(s1);
17     puts(s2);
18
19     return 0;
20 }
```

Program 1.4 puts() ve gets()

Program 1.4'te C programlarında puts() ve gets() fonksiyonlarının nasıl kullanılacağı gösterilmiştir.

C programlarında ekrana yazdırılacak ifade bir karakter topluluğu ise, printf()'e alternatif olarak puts() fonksiyonu kullanılabilir. Ancak puts(), ekrana bu karakter topluluğu yazdıktan sonra, imleci alt satıra geçirir. Buna göre: printf("Adınızı Giriniz\n"); ile puts("Adınızı Giriniz"); kullanımları eşdeğerdir. puts() fonksiyonu Tablo 1.1 de verilen kontrol karakterleri ile kullanılabilir.

C programlarında gets() fonksiyonu klavyeden bir karakter topluluğu okumak için kullanılır. Okuma işlemi yeni satır karakteriyle(\n) karşılaşıncaya kadar sürer. puts() - gets() arasındaki ilişki, printf() - scanf() arasındaki ilişki gibidir. Yani,

scanf("%s",s1);
İle

gets(s1);

Aynı anlamdadır.

1.4.1.4. getche() Fonksiyonu

C programlama dilinde getche() fonksiyonu ile klavyeden (standart girişten) tek karakter okunur. En çok kullanılma amacı, getche() fonksiyonun programın çalışmasının durdurulması gereken noktaya yazılır ve programın çalışması durdurulur. Daha sonra klavyeden basılacak tuşa göre programın çalışması yönlendirilir. Program 1.5'te getche fonksiyonun kullanışı gösterilmiştir.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  int main() {
5      setlocale(LC_ALL,"Turkish");
6      char s1[10];
7      char s2[10];
8      char secim;
9      //Ad Soyad Giriliyor
10     do{
11         printf("Öğrencinin Adı   : ");
12         scanf("%s",s1);
13         printf("Öğrencinin Soyadı : ");
14         scanf("%s",s2);
15         printf("devam edecek misiniz : ");
16         secim=getche();
17         printf("\n");
18     }while(secim!='h');
19
20     return 0;
21 }

```

Program 1.5

Bölüm Özeti

Veri yapısı kavramının tanımını yapabilmek

Veri Yapısı, verinin bilgisayar belleğinde nasıl organize edildiğini, bellekte tutulma biçimini ve sırasını gösterir. Günümüzde veri, enformasyon ve nihayet bilgi hem bireyler hem işletmeler hem de ülkeler için önemli bir rekabet aracı haline gelmiştir. Dolayısı ile gün geçtikçe verinin toplanması, saklanması ve işlenip karar süreçlerinde kullanılmasının sağlanması daha da önemli olacaktır. Bunun için veri yapısı araçlarını, algoritmalarını ileri düzeyde öğrenmemiz gerekmektedir.

Veri (Data), Enformasyon (Information), Bilgi (Knowledge) kavramlarını açıklayabilmek

Veri, işlendiğinde veya düzenlendiğinde anlamlı çıktı oluşturan, en düşük soyut veya işlememiş ham girdiyi (**raw data**) ifade eder.

Enformasyon, verinin anlamlı bir konu etrafında işlenmiş halidir. Başka bir ifadeyle veriler işlendiğinde enformasyona dönüşür. Kısacası veriden enformasyona ulaşılabilmesi için verilerin ilişkilendirilmesi, kategorize edilmesi, özetlenmesi ve hesaplanması gerekir.

Bilgi, kuruluşların hedef(ler) ine ulaşmak amacıyla etkili eylemde bulunmak için kullanabilecekleri fikirleri veya anlayışlarıdır şeklinde tanımlanabilir.

Veri yapılarını Temel Veri Yapıları, Basit Veri Yapıları ve Birleşik Veri Yapıları şeklinde sınıflandırabilmek,

Her programlama dilinin desteklediği, kullanıma izin verdiği temel veri tipleri vardır. Ancak çoğu zaman bu temel yapılar verilerin bellekte organize edilmesi için yeterli olmaz. Bu nedenle birçok dil temel veri tiplerinden yararlanılarak basit yapılar oluşturulmasına, yani kullanıcının kendisinin veri tipini tanımlamasına izin verir. Ancak bu da çoğu kere yeterli olmaz bu nedenle birleşik veri yapıları oluşturulmuştur. Veri yapıları dersinin konusu da bu birleşik veri yapılarıdır.

Veri Yapıları dersi araçlarından derleyici, editör ve C programlama dilini kullanabilecek

Veri yapıları dersini işlerken her bir veri yapısı hakkında teorik bilgiler verilecektir. Ancak konuların anlaşılması için örnekler de çözmemiz gerekecektir. Bu nedenle bir editör ve derleyiciye ihtiyacımız olacak ve bir de örnekleri çözeceğimiz programlama diline ihtiyaç duyacağız. Bu derste editör ve derleyici için DEVC++ seçilmiştir. Programlama dili olarak da ANSIC programlama Dili kullanılacaktır.