

# BSM101 Programlama Dilleri I

## Hafta 1 Programlamaya Giriş

Dr. Öğr. Üyesi Caner ÖZCAN

# Derse Giriş

► Ders Web Sitesi: [www.canerozcan.net](http://www.canerozcan.net)

► Ofis Saatleri: **Pazartesi 15:30-16:30**  
**Salı 13:00-14:30**

ya da email ile randevu alınız:

[canerozcan@karabuk.edu.tr](mailto:canerozcan@karabuk.edu.tr)

- Kaynak Kitaplar:

Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015

Harvey Deitel, “C How to Program”, Pearson Education

# Derse Giriş

- ▶ Bol bol pratik yapın!
- ▶ Notlandırma
  - Vize Sınavı: %40
  - Final Sınavı: %60
- ▶ Not için değil, öğrenmek için çalışın. Not nasılsa kazanılır.

# Dönem Boyunca Görülecek Konular

- Hafta 1- Temel Kavramlar ve Algoritma
- Hafta 2- Giriş/Çıkış İşlemleri, Değişken Kavramı ve Basit Matematiksel İşlemler
- Hafta 3- Kontrol Yapıları (if / if-else / switch-case)
- Hafta 4- Döngü Yapıları (while / do-while / for)
- Hafta 5- Algoritma Örnekleri ve Analizi-1
- Hafta 6- Algoritma Örnekleri ve Analizi-2
- Hafta 7- C Programlama Diline Giriş ve C Derleyicisi
- Hafta 8- C Dilinde Değişken Tipleri ve Temel Giriş/Çıkış İşlemleri (printf, scanf)
- Hafta 9- C Dilinde Kontrol ve Döngü Yapıları
- Hafta 10-Diziler
- Hafta 11-Çok Boyutlu Diziler
- Hafta 12-Karakter Tutan Diziler
- Hafta 13- Fonksiyonlar-1
- Hafta 14- Fonksiyonlar-2

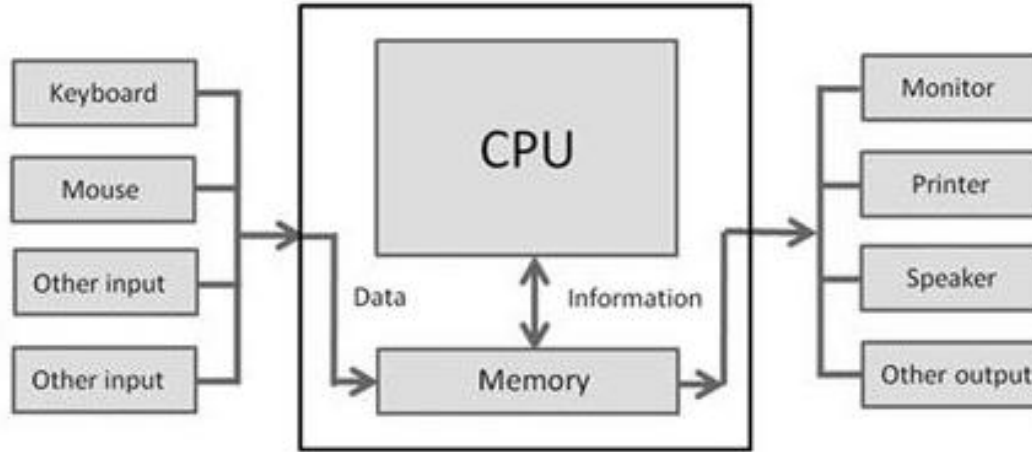
# Bilgisayar Nedir?

- Bilgi ile çalışmak için tasarlanmış bir elektronik cihazdır.
- ‘computare’ kelime kökünden gelmekte ve kelime olarak hesaplama yapmak veya programlanabilir makine anlamına gelir.
- **Bilgisayarlar program olmadan hiçbir şey yapamazlar.**

Bilgisayar,

- kullanıcıdan girdi olarak ham veri alan,
  - bu verileri komut seti (program adı verilen) kontrolü altında işleyen,
  - sonucu veren (çıktı),
  - ileride kullanmak üzere çıktıyı kaydeden gelişmiş bir elektronik cihazdır.
- Hem sayısal hem de sayısal olmayan (aritmetik ve mantıksal) hesaplamaları yapabilir.

# Bilgisayar Nedir?

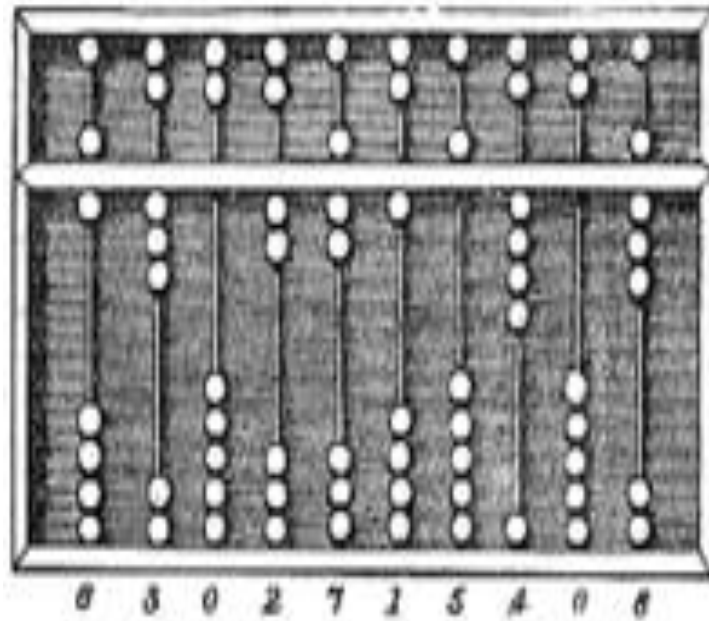


Modern bir dijital bilgisayarın temel bileşenleri şunlardır: Giriş Cihazı, Çıkış Cihazı, Merkezi İşlemci Birimi (CPU), yığın depolama cihazı ve bellek.

accepts data	Input
processes data	Processing
produces output	Output
stores results	Storage

# Bilgisayar Nedir?

- En eski hesaplama aracı "Abaküs"

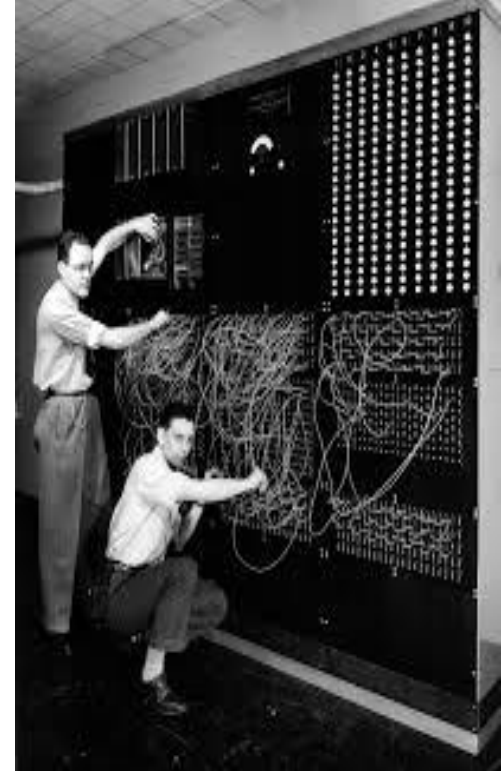


# Bilgisayar Nedir?

- Kronolojik olarak bilgisayarların gelişimi:
  - M.Ö. 500 Abaküs
  - 1642 Pascal'ın Mekanik Toplama Makinesi
  - 1827 Babbage'in çıkarma makinesi
  - 1941 İkili>İkili Mekanik Hesaplayıcı (Zuse)
  - 1944 Ondalık Elektromekanik Hesaplayıcı (Aiken)
  - 1945-54 İlk Kuşak Vakum Tüpler ve ışınlar
  - 1955-64 İkinci Kuşak Transistorlar ve Manyetik Bellekler
  - 1965-71 Üçüncü Kuşak Tümlleşik Devreler
  - 1971-90 Dördüncü Kuşak VLSI Devreler
  - 1982 IBM PC & MS-DOS
  - 1984 MAC
  - 1990'lar Paralel İşlemciler

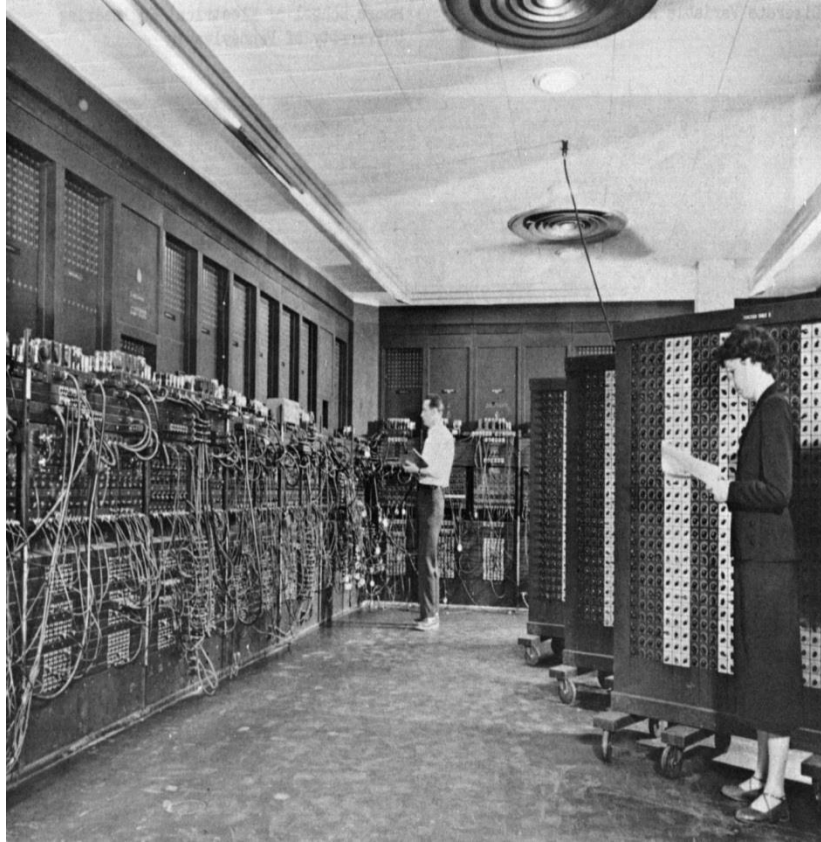


# Bilgisayar Nedir?



IBM MARK-I

# Bilgisayar Nedir?



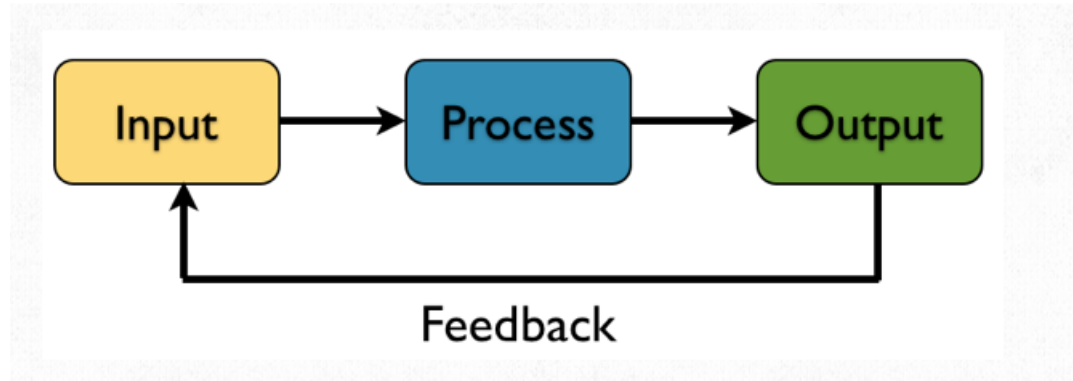
ENIAC- Electronic Numerical Integrator And Computer

# Mühendislik Nedir?

- Mühendislik, problemleri çözmek için fen ve matematiğin uygulanmasıdır.
- Mühendislik, problem çözmeye adanmış bir disiplindir.
- Gelişmiş protezler üzerinde çalışan, yeni malzemeler üreten, motor verimliliğini ve alternatif yakıtları inceleyen, köprüler inşa eden veya temiz su sistemleri geliştiren mühendislerdir.
- Uydulardan hücre zarlarına kadar mühendisler olağanüstü şeyler elde etmek ve dünyanın en karmaşık sorunlarının bazılarını çözümler bulmak için matematik ve bilimi kullanırlar.

# Bilgisayar Mühendisliği

- Bilgisayar mühendisleri, bilgisayar sistemleri ve diğer teknolojik aygıtları tasarlar ve geliştirir.



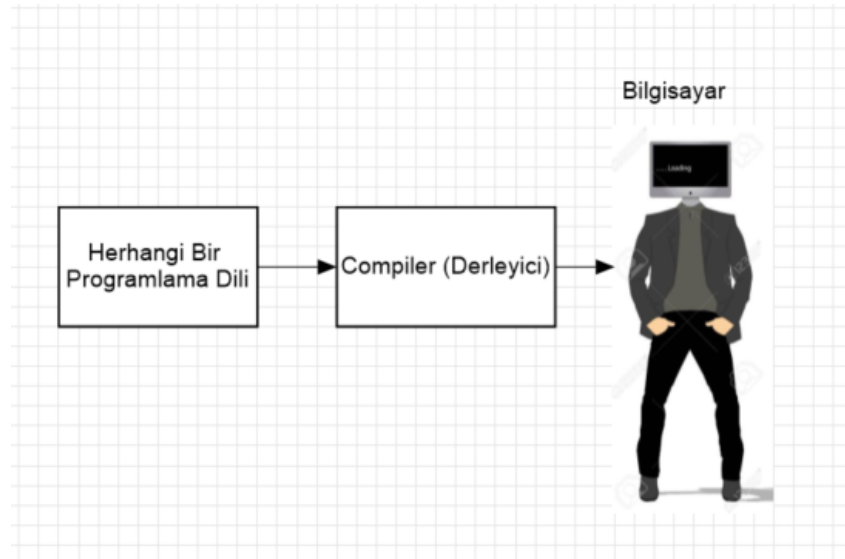
\* Genel bir sistemin yapısı

# Programlama Dili

- Bilgisayarın fonksiyonel bir beyni olmadığı için insanlar komutlar yazmalıdır.
- Bir problemin algoritmik çözümünün Bilgisayara anlatılmasını sağlayan, son derece sıkı-sıkıya kuralları bulunan kurallar dizisidir.
- Bu özel dile “**programlama dili**” denir.

# Programlama Dili

- Bilgisayar programlama dilleri, bilgisayara bilgisayarın anladığı bir dilde talimatlar vermemizi sağlar.
- Birçok insan tabanlı dil olduğu gibi, programcıların bir bilgisayarla iletişim kurmak için kullanabileceği bir dizi bilgisayar programlama dili vardır.
- C Dilinden Python'a kadar her dilin ortak özellikleri olmasına rağmen kendine özgü özellikleri de vardır.



# Programlama Dili

Pascal

C++



C  
Language

Microsoft®  
Visual C#®



# Veri Saklama Birimleri

- Bilgisayar sistemlerindeki bütün bilgiler ikilik sistemde **1** ve **0** ile temsil edilen elektrik sinyalleri ile saklanır.
- İkilik sistemdeki her bir basamağa **bit** denir.
- Bit nicelik ifade edebilmek için yeterli bir birim değildir. Temel hafıza birimi olarak **byte** kullanılır.
  - 1 bayt = 8 bit
- Bilgisayar sisteminde her bir karakter 8 bit'ten oluşur.
- Örneğin: **A** karakteri bilgisayar içinde 0100001 sayısıyla ifade edilir. İşte bu sayının her basamağına 1 Bit denir.



# Veri Saklama Birimleri

- Kapasite ölçüm birimleri küçükten büyüğe Bit, Byte (Bayt), KB (Kilo Byte), MB (Mega Byte), GB (Giga Byte), TB (Tera Byte).
- 1 Byte=8 Bit
- 1 KB (Kilo Bayt)=1024 Byte
- 1 MB (Mega Bayt)=1024 KB
- 1 GB (Giga Bayt)=1024 MB
- 1 TB (Tera Bayt)=1024 GB'dır.

# Programlama Nedir?

- Kullanım amaçları ve yerlerine göre birçok değişik program türü vardır:
  - Sistem programları: Her program, bir işletim sistemi üzerinde çalışır. İşletim sistemi, diğer programların çalışması için gerekli olan kaynakları ve ortamı sağlar.
  - Sürücüler: İşletim sistemi ile donanım aygıtları arasında iletişim sağlayan programlardır. Klavye ile yazılan yazıların algılanması için, klavyenin sürücü programı kullanılır.
  - Uygulamalar: İşletim sistemi üzerinde çalışan, kullanıcıların ihtiyaç duyduğu işlevleri sağlayan programlardır.

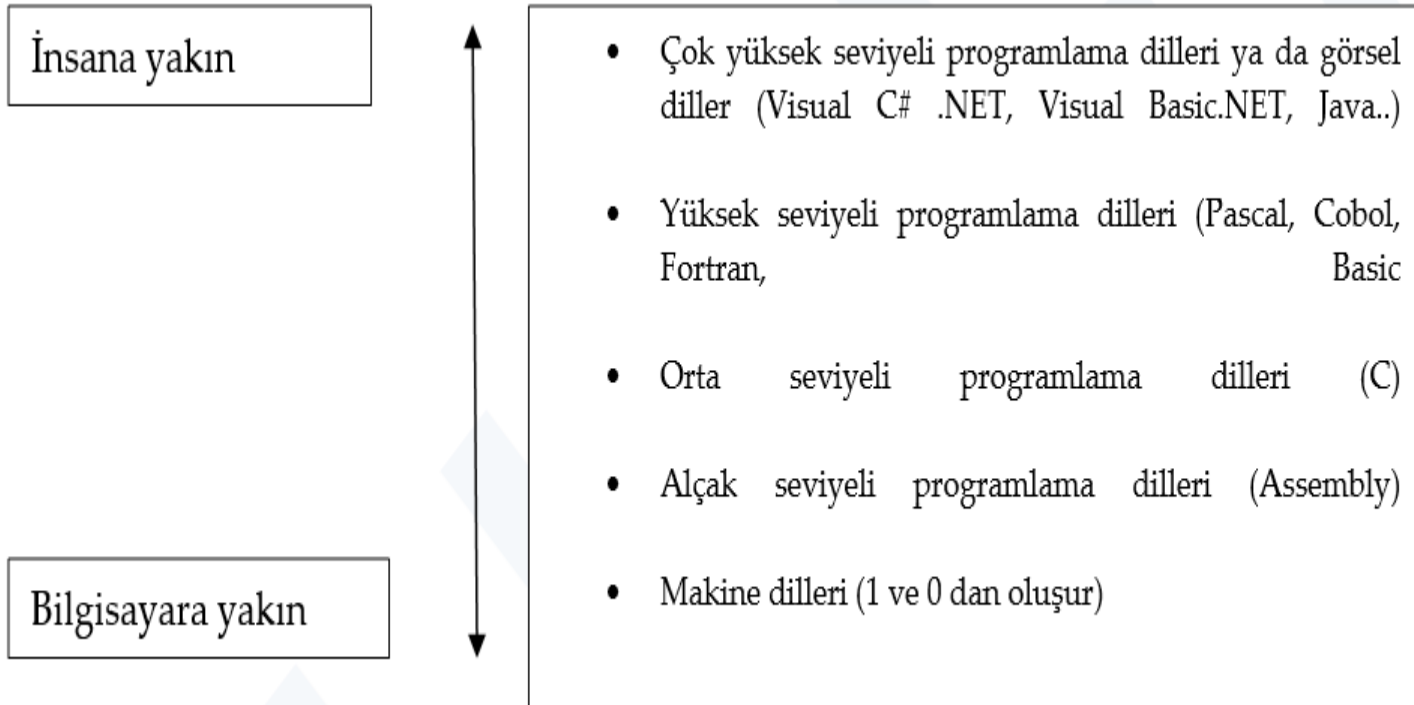
# Programcı Kimdir?

- Belirli işlevlere sahip programlar geliştiren kişilere programcı denir. Programcı, kullanılan teknolojiyi, platformu iyi tanınması gerekir.
- Programcılar 3 grupta incelenir:
  - **Mimar:** Programların yazılması için gerekli teknolojileri belirler.
  - **Geliştirici:** Programı yazan kişidir.
  - **Analist:** Programın geliştirilmesi aşamasında, hatanın kaynaklarını bulan ve geliştiricilere raporlayan programcıdır.

# Programlama Dillerinin Sınıflandırılması

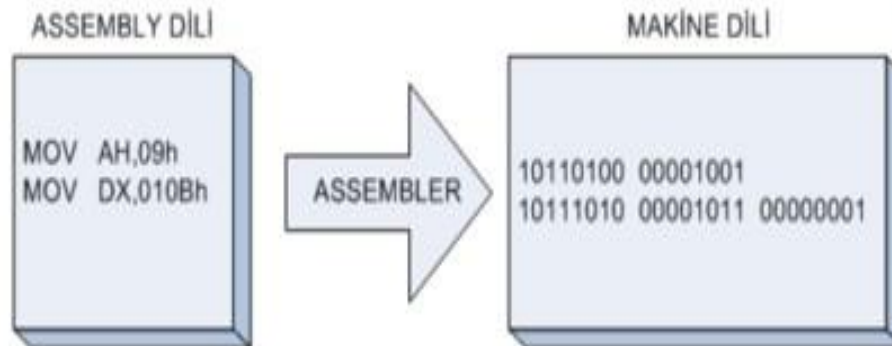
- Programlama dilleri sıklıkla seviyelerine göre sınıflandırılır.
- Seviye, bir programlama dilinin insan algılamasına olan yakınlığının bir ölçüsüdür.
- Yüksek seviyeli diller insan algılayışına daha yakın, alçak seviyeli diller de bilgisayarın doğal çalışmasına daha yakın olan dillerdir.
- Dilin seviyesi yükseldikçe programcının işi de kolaylaşır.
- Çok yüksek seviyeli dillerde bir işin nasıl yapılacağına ilişkin değil, ne yapılacağına ilişkin komutlar bulunur.
- Seviyenin yükselmesi programcının işini kolaylaştırırken genel olarak verimliliği ve esnekliği azaltır.

# Programlama Dillerinin Sınıflandırılması



# Programlama Dillerinin Sınıflandırılması

- **Makine Dilleri**
  - Bilgisayarın kendi doğal dili
  - Sayı dizilerinden oluşur (çoğunlukla 1 ve 0'a indirgenmiştir.)
  - Makine bağımlıdır
- **Alçak Seviyeli Diller (Assembly Dili)**
  - Temel komutları temsil etmek için İngilizce benzeri kısaltmalar.
  - Assembler, Assembly dilini makine diline çevirmek için kullanılır.



# Programlama Dillerinin Sınıflandırılması

- Makine dili programlaması çok yavaş ve hata yapmaya çok elverişlidir.
- Bilgisayarların doğrudan anlayabildiği sayı dizileri ile uğraşmak yerine programcılar temel işlemleri temsil etmek için İngilizce kısaltmalar kullanmaya başladılar.
- Bu kısaltmalar **assembly dilinin** temelini oluşturmuştur.
- **Assembler**'lar assembly dilinde yazılan bir programı makine diline çevirmek için tasarlanmışlardır.

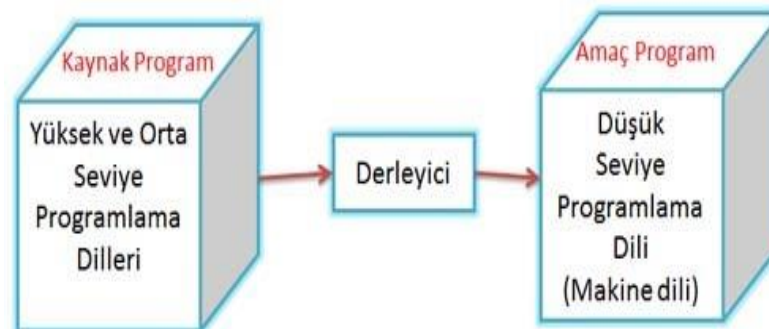
# Programlama Dillerinin Sınıflandırılması

- **Orta seviyeli diller** hem kullanıcıya hem de bilgisayara yakın olan yapılar içerirler.
- Orta seviyeli diller, yüksek seviyeli dillerin kolaylıkları ile aşağı seviyeli dillerin esnekliğini ve doğallığını kullanırlar.
- C tipik bir orta seviyeli dildir.
- Orta seviyeli diller özellikle sistem programlarının yazımında kullanılırlar.



# Programlama Dillerinin Sınıflandırılması

- **Yüksek seviyeli programlama dilleri** daha algoritmik dillerdir.
- Bu dillerde önce işlerin nasıl yapılacağına ilişkin algoritmalar tasarlanır. Daha sonra bu algoritmalar program koduna çevrilir.
- Basic, Pascal, Fortran gibi diller bu grup dillerdir.



# Programlama Dillerinin Sınıflandırılması

- **Çok yüksek seviyeli diller**, program kodunun kısmen ya da tamamen görsel biçimde çeşitli araçlar tarafından üretildiği dillerdir.
- Windows sistemlerinde .NET çatısı altında C# ve Visual Basic gibi diller yoğun olarak kullanılmaktadır.
- Çok yüksek seviyeli dillere deklaratif diller de denir.
- Veri tabanlarının yönetimlerinde kullandığımız dilleri bu gruba sokabiliriz.

# Derleyici Nedir?

- Derleyici (Compiler), bir bilgisayar dilinde yazılmış olan kodu, bilgisayarın (yada elektronik cihazın) donanımına uygun makine diline çeviren bilgisayar programıdır.
- Derleyici öncelikle yazılan program kodunun doğru yazılıp yazılmadığını kontrol eder, eğer hatalar varsa bunları programcıya bildirir.
- Eğer kod doğru ise derleme yapılan sisteme uygun olan 0 ve 1'lerden oluşan makine kodunu üretir (EXE dosyası).

# Yorumlayıcı Nedir?

- Yorumlayıcı (Interpreter), bir programın kaynak kodunu derlemek yerine doğrudan satır satır yürüten bir programdır.
- BASIC gibi bazı diller hem derleyici hem yorumlayıcı programlar ile kullanılabilir.
- JAVA dili, yorumlayıcı kullanması nedeniyle taşınabilir (platform bağımsız) bir yapıya sahip olmuştur.

# Derleyiciler ile Yorumlayıcılar Arasındaki Farklar

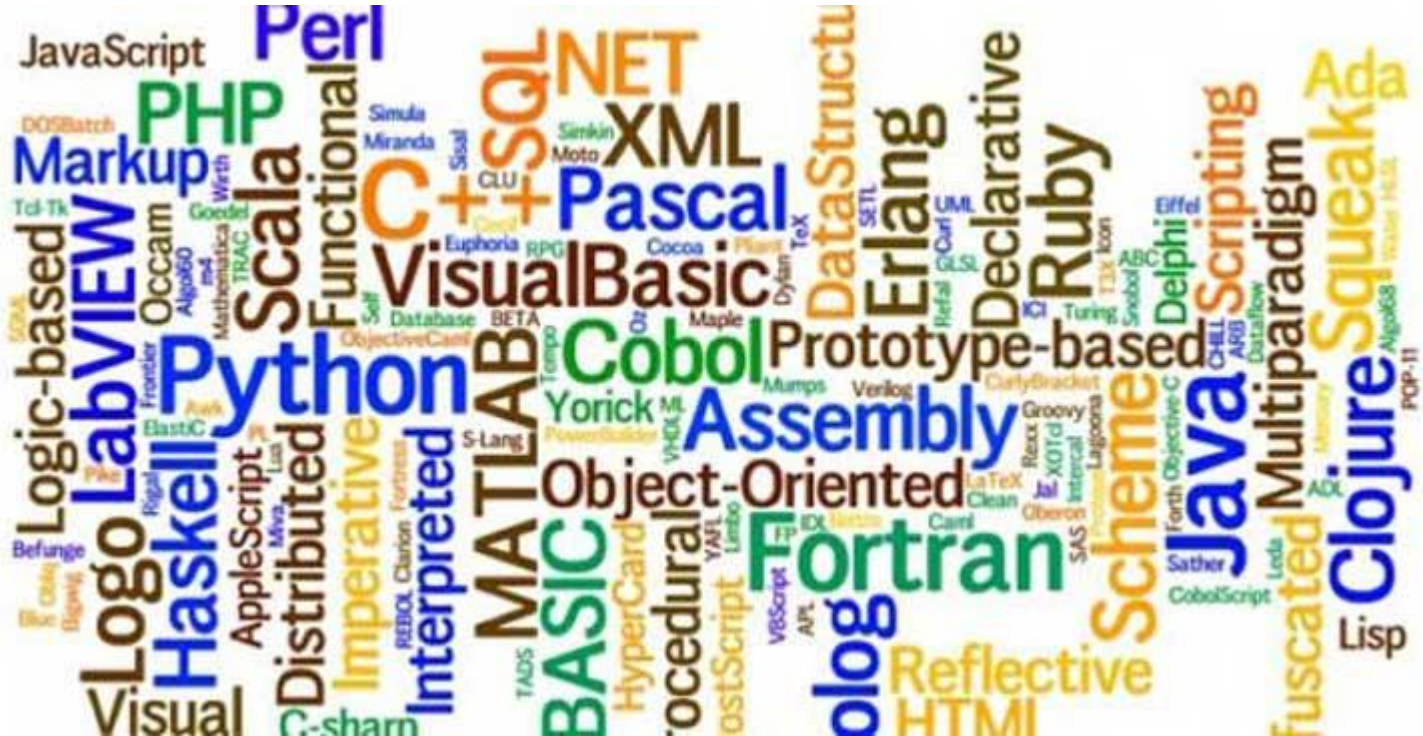
- Derleyiciler, basitçe bir kaynak kodu hedef koda çevirdikten sonra çalıştırır ve dolayısıyla koddaki hataları yakalama işlemini ve kodun iyileştirilmesini daha kod çalıştırmadan yapar.
- Yorumlayıcılar ise kodu satır satır veya bloklar halinde çalıştırıp sırası gelmeyen satırları hiç çalıştırmaz ve bu satırlardaki hataları hiçbir zaman göremez. Kodun bütününe ait iyileştirmeleri yapamaz.

# Bilgisayarlar





# Neden Programlama



# Problem ve Problem Çözümü

- Bir işlemin, otomasyonun yada bilimsel hesaplamanın bilgisayarla çözülmesi fikrinin ortaya çıkmasına **problem** denir.
- Problemi çözebilmek için öncelikle sorunun çok net olarak programcı tarafından anlaşılmış olması gerekir.
- Tüm ihtiyaçlar ve istekler belirlenmelidir.
- Bir sorunun tabii ki birden fazla çözümü olabilir. Bu durumda bilgisayar ile en uygun çözüm seçilmelidir.



# Algoritma

- Bir sorunu çözebilmek için gerekli olan sıralı mantıksal adımların tümüne **algoritma** denir.
  - Her adım son derece belirleyici olmalıdır. Hiç bir şey şansa bağlı olmamalıdır.
  - Belirli bir sayıda adım sonunda algoritma sonlanmalıdır.
  - Algoritmalar karşılaşılabilecek tüm ihtimalleri ele alabilecek kadar genel olmalıdır.
- Algoritmalar **pseudo kod(sahte kod)** veya **akış diyagramları** ile ifade edilirler.

# Algoritma



# Pseudo Kod

- **Pseudo kod** algoritma geliřtirmeye yardımcı olmak için kullanılan yapay ve resmi olmayan bir dildir.
- Pseudo code günlük konuşma dili gibidir;
  - Kolay
  - Kullanıcı dostu
  - Gerçek olmayan bir programlama dili.
- Bilgisayarlar tarafından çalıştırılmaz.
- Bir programı yazmaya başlamadan önce onun hakkında düşünmenize yardımcı olur.
- Sadece eylem ifadelerinden oluşur. Tanımlamalar çalıştırılabilir ifadeler değildir ve herhangi bir eyleme sebep olmaz. Bu sebeple pseudo kod içinde yer almazlar.

# Pseudo Kod

- Bazı temel sahte kod komutları şunlardır:
- **Başla** : Programın başladığını ifade eder.
- **Bitir** : Programın bittiğini ifade eder.
- **Oku** : Kullanıcı girişi için yazılır.
- **Yaz** : Kullanıcıya bilgi veya sonuç göstermek için yazılır.
- **Eğer ... İse ...** : Şartlara göre akışın değişmesinde kullanılır.
- **Eğer .... Değilse...** : Şartlara göre akışın değişmesinde kullanılır.

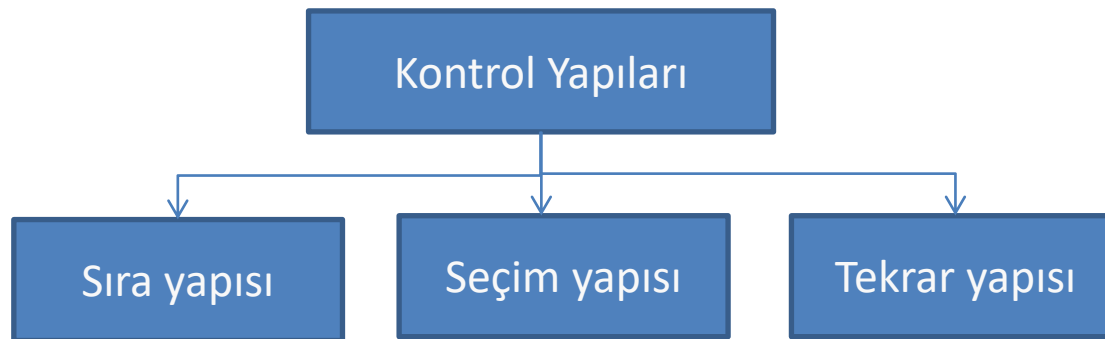
# Akış Diyagramı

- Çeşitli anlamlar ifade eden ve birbirine oklarla bağlanan şekillerle görsel olarak algoritmanın adımlarını ifade etmektir.
- Akış şemaları Dikdörtgen, Baklava, Elips, Daire gibi özel amaçlı bazı sembollerin çizilmesi ile oluşturulurlar.



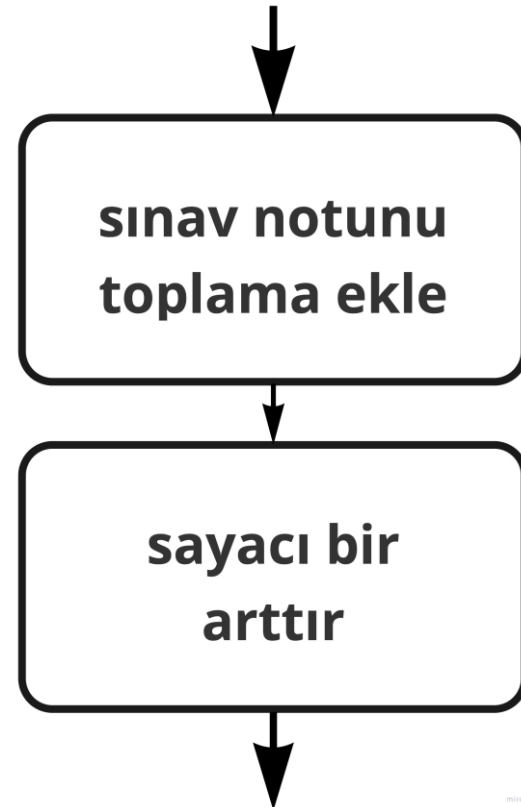
# Kontrol Yapıları

- Araştırmacılar tüm programların sadece üç kontrol yapısı ile yazılabileceğini gösterdi.
- Bu yapılar, sıra yapısı, seçim yapısı ve tekrarlama yapısı.



# Kontrol Yapıları (Sıra Yapısı)

- Bir programdaki ifadeler bir biri ardına yazıldıkları sırada çalıştırılırlar.
- Aksi belirtilmediği sürece C ifadeleri bir biri ardına yazıldıkları sıra ile çalıştırılırlar.



# Kontrol Yapıları (Sıra Yapısı)

- Örneğin; eğer bir uçak bir hedefe füze göndermek isterseniz adımlarınız şu şekilde olabilir.
- Adım sırası takip edilmez ise işlem başarısız olacaktır.
  - 1. Hedefin koordinatlarını al
  - 2. Füzenin şu anki koordinatını al
  - 3. Hedefi vurması için gerekeni hesapla
  - 4. Füzeyi ateşle



# Kontrol Yapıları (Sıra Yapısı)

## 1. adımın daha da gelişmiş hâli şöyle olabilir:

- a. Füze teknisyeninden hedef koordinatları edin
- b. Hedef koordinatların geçerli olduğundan emin ol
- c. Hedef koordinatları hafızaya kaydet
- a. maddesini daha alt basamaklara bölebiliriz.
- i. Koordinattan emin ol
- ii. Hedef füzenin atış uzaklığı içinde mi, kontrol et
- iii. Hedef dost bölge mi, kontrol et

# Algoritma

- Temelde algoritmamızı üç ana bölüme ayırabiliriz:
- **Giriş:** Bilgisayarın üzerinde çalışacağı veri kullanıcı tarafından girilir.
- **İşlem:** Bilgisayar girilen bilgiyi işler.
- **Çıkış:** Bilgisayar insanların anlayacağı şekilde ekrana sonucu gösterir.

# Algoritma

- **Giriş:**
- a. Ürünün adını al.
- b. Ürünün miktarını al.
- c. Bir dosyadan ürünün fiyatını oku.
- **İşlem:**
- a. Ürünün toplam fiyatını hesapla.
- b. Gerekli indirimini yap.
- **Çıkış:**
- a. Toplam satış değerini yaz.

- **Örnekteki adımları algoritma olarak alt alta toplarsak:**
- 1. Başla.
- 2. Ürünün adını al.
- 3. Ürünün miktarını al.
- 4. Bir dosyadan ürünün fiyatını oku.
- 5. Ürünün toplam fiyatını hesapla.
- 6. Gerekli indirimini yap.
- 7. Toplam satış değerini yaz.
- 8. Bitir.

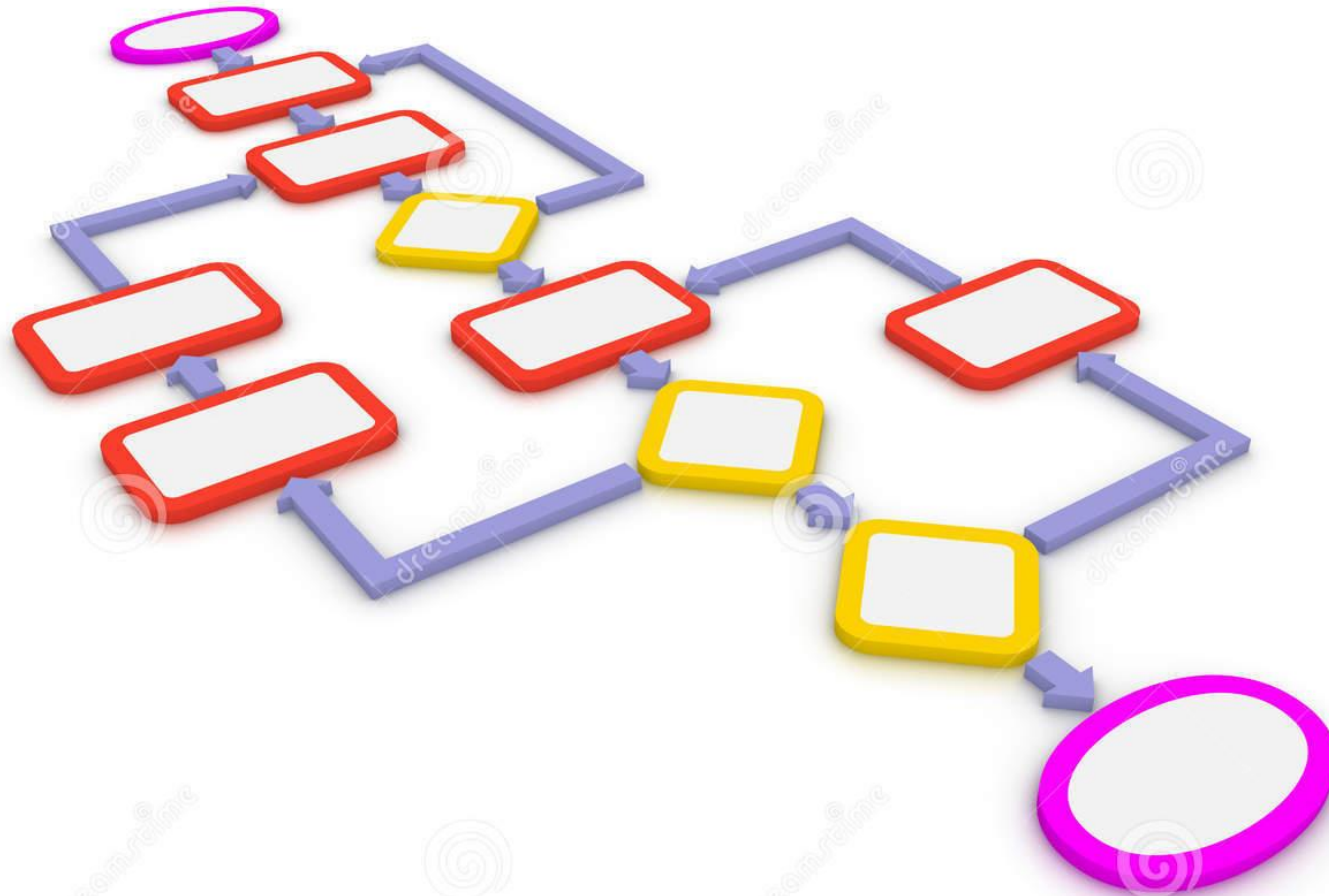
# Flow chart



# Pseudo Code

1. Get up from the bed. Switch off the alarm. Get out of your room.
2. Take the toothbrush. Take the toothpaste. ***Start brushing your teeth.***
3. Go to the bathroom. Take off your clothes. Choose hot/cold shower.  
***Shower.***
4. Go to the dressing room. Get your dress for that day. ***Dress-up.***
5. Go to the parking lot. Take your car/bike out. ***Get into it.***
6. Start the car/bike. Check out your way. ***Drive to the work/College.***

# Algoritma



# Kaynaklar

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ Bayram AKGÜL, C Programlama Ders notları