



BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

DERS 5
Temel Bileşenler
Haberleşme

Diğer cihazlarla haberleşme

- ▶ Bir bilgisayarın çekirdeğini **cpu** + **anabellek** oluşturur.
- ▶ Çekirdeğin haberleştiği çevre birimleri şunlardır:
 - ▶ Diskler
 - ▶ Yazıcı
 - ▶ Klavye
 - ▶ Fare
 - ▶ Ekran
 - ▶ Diğer bilgisayarlar gibi..



Bilgisayar donanımının temel bileşenleri

Donanım (Hardware)

- Bilgisayar donanımı, bir bilgisayarın fiziksel parçalarına karşılık gelmektedir.

Bilgisayar Donanımının Ana Bileşenleri

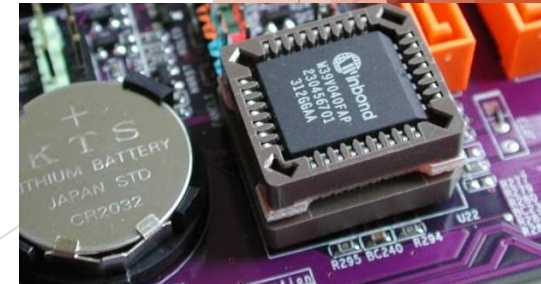
Masaüstü bilgisayarlar üzerinden...

- ▶ Güç kaynağı
- ▶ Anakart
- ▶ mikroişlemci
- ▶ Bellek
- ▶ Sabit disk sürücüsü
- ▶ CD / DVD sürücüsü
- ▶ Ses kartı
- ▶ Grafik kartı
- ▶ monitor
- ▶ Tuş takımı
- ▶ Fare
- ▶ ...





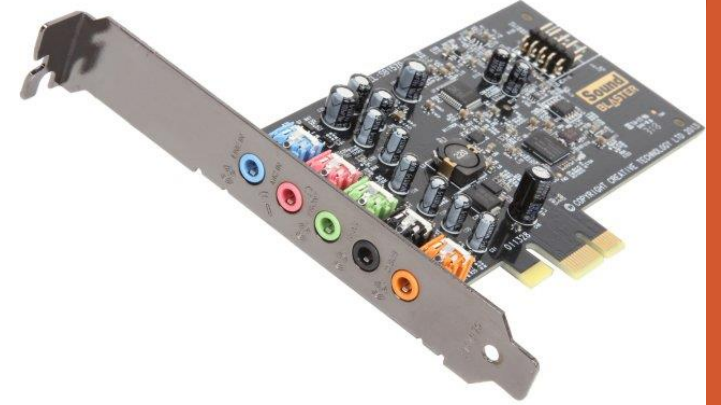
- **Güç kaynağı:** Bileşenler için güç kaynağı sunar.
- **Mikroişlemci, Merkezi İşlem Birimi (CPU):** Bir bilgisayarın beyni. yazılımlarının komutlarını işler.
- **Temel Giriş / Çıkış Sistemi (BIOS):** Bir ROM chipinde saklanan bir program. Bilgisayarınızı başlattığınızda aslında BIOS'u başlatır. Donanımların başlatılması ve işletim sisteminin yüklenmesinden sorumludur.





- ▶ **Rasgele Eriřim Belleęi (RAM):** İşlenecek verileri ve komutları saklar. Bilgisayarınızın gücü kapalıyken, birincil bellekteki veriler kaybolur.
- ▶ **Salt Okunur Bellek (ROM):** Okunabilir, ancak yazılabilir deęil.
- ▶ **Harddisk Sürücüsü (HDD):** Manyetik disk üzerinde, verileri kalıcı olarak saklar.
- ▶ **CD-ROM, DVD:** Verileri optik disk üzerinde, kalıcı olarak saklar.





- ▶ **Ses kartı:** Dijital sinyalleri analog ses sinyallerine dönüştürmemizi sağlayan bir genişleme kartıdır.
- ▶ **Grafik kartı:** Görsel veri üreten bir karttır.



Bilgisayar kasası

- ▶ Bir bilgisayar kasası, bir masaüstü bilgisayarın sistem bileşenlerinin çoğunu içerir.
- ▶ Elektrik yalıtımı sağlar.
- ▶ Çevre birimlerin I/O bağlantı noktaları ve açma veya yeniden başlatma düğmeleri için bazı işlemleri sağlar.
- ▶ Dışarıya sıcak havayı transfer etmek için soğutma fanları içerir.
- ▶ Bir kasanın içinde genellikle aşağıdaki bileşenler vardır.
 - ▶ Güç kaynağı
 - ▶ Anakart
 - ▶ Sabit disk sürücüsü
 - ▶ CD / DVD sürücüleri

Different shapes;

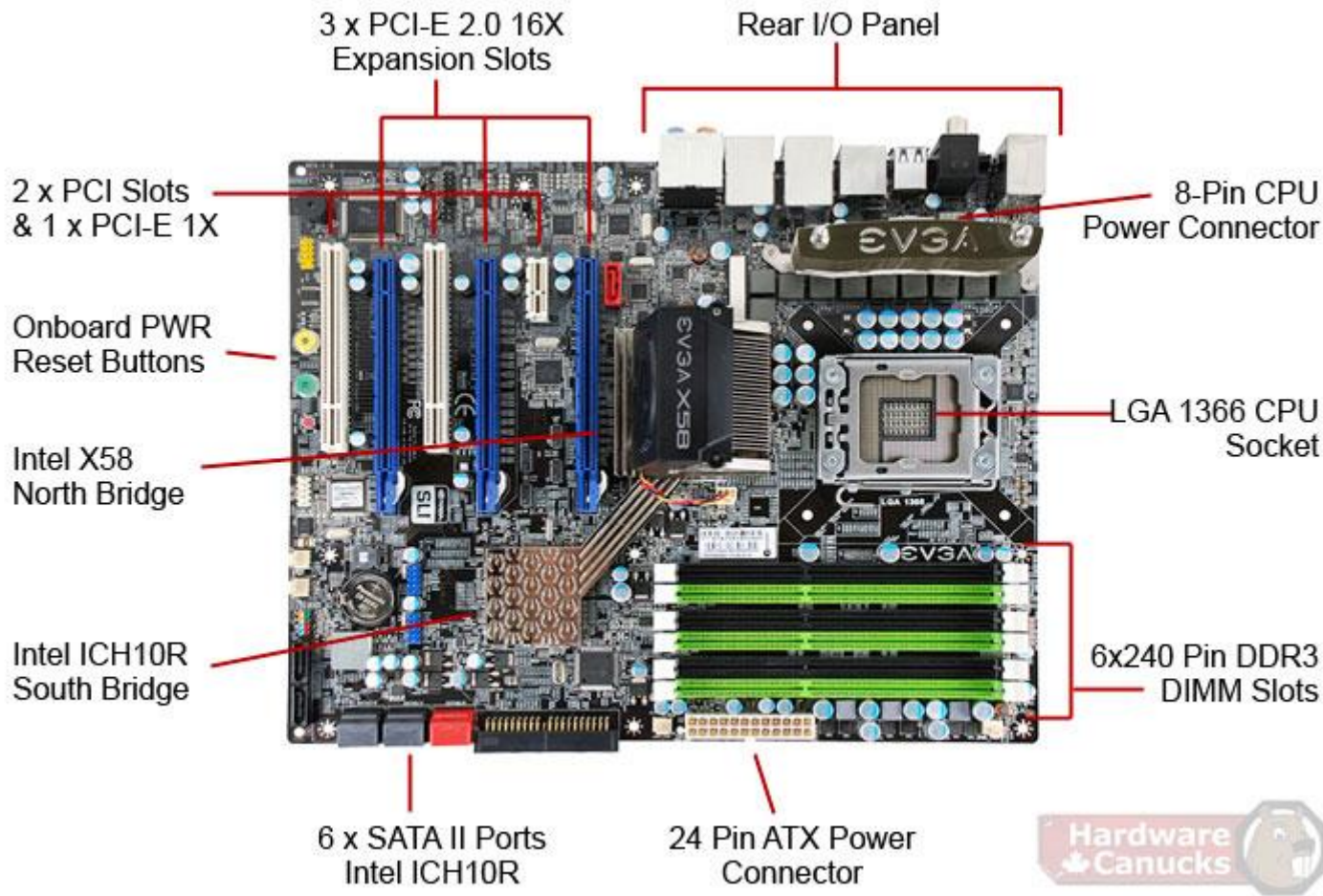
tower case (height>width)

desktop case (width>height)



Anakart

- ▶ Anakart bir baskılı devre kartıdır ve CPU, RAM,... gibi bir bilgisayarın temel bileşenleri için bir iletişim ortamı ve ev sahipliği sağlar.
- ▶ Anakartta ayrıca bazı bilgisayar parçaları arasında iletişim sağlamaktan sorumlu olan bazı bileşenler vardır:
 - ▶ **Slots**, dar ve uzun bağlantı yerleridir.
 - ▶ CPU'nun ana kart ile bağlantısı için bir CPU **Soketi** kullanılır.
 - ▶ **Bus**: Sistemin CPU ve RAM gibi bileşenler arasında iletişim sağlamak için kullanılan veri yollarıdır.
 - ▶ **Port**: Farklı cihazları bağlamak için kullanılan yerler.
 - ▶ **Seri portlar**: Bir seferde sadece bir bit gönderilebilir
 - ▶ **Paralel portlar**: Bir seferde birkaç bit gönderilebilir



PCI: Peripheral Component Interconnect

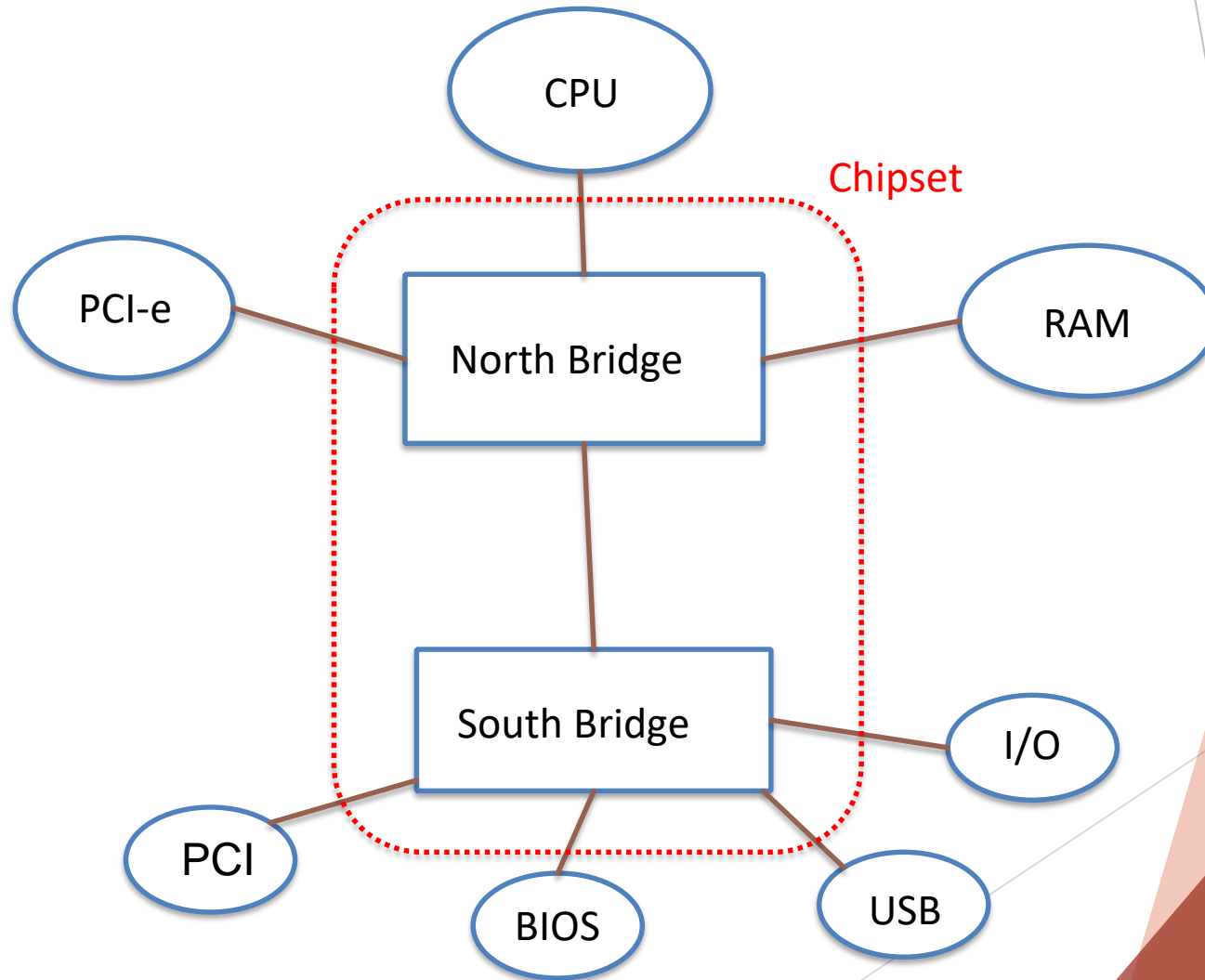
I/O: Input/Output

Chipset: Specific collection of chips. These chips generally deal with data flow between CPU, memory and peripherals.

- * North bridge

- * South bridge

Chipset



- ▶ **Original Equipment Manufacturer (OEM): Orijinal Ekipman Üreticisi** Farklı şirketlerinin ürünlerini içeren bir bilgisayar sağlayan ve kendi markası altında toplam sistem satan üretici. HP, Asus, Dell, Sony, .. gibi.

Denetleyicilerin rolü

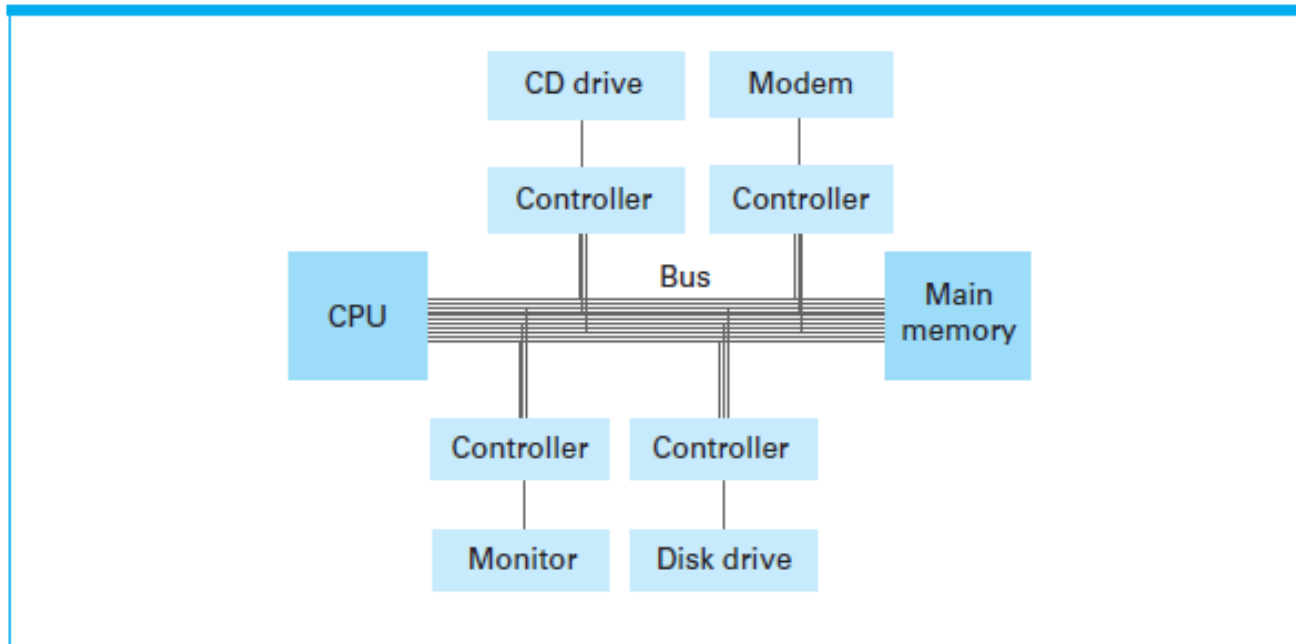
- ▶ Bilgisayarın diğer cihazlarla iletişimi denetleyiciler tarafından yönetilir
- ▶ Denetleyici **anakart üzerinde bir devre** veya **yuvalara takılan bir devre kartı** olabilir.
- ▶ **Çevresel birimlere direkt bağlanabilir** veya çevresel birimlerin bağlanabileceği bir port bulundurabilir.

Denetleyicilerin rolü

- ▶ Denetleyicilerin **kendine ait bellekleri** ve **basit işlemcileri** olabilir.
- ▶ Görevi; bilgisayar ve çevresel birimin iletişimde kullanılan **veri formatlarını** birbirine çevirmektir.
- ▶ **USB, FireWire** denetleyiciler için örnektirler.

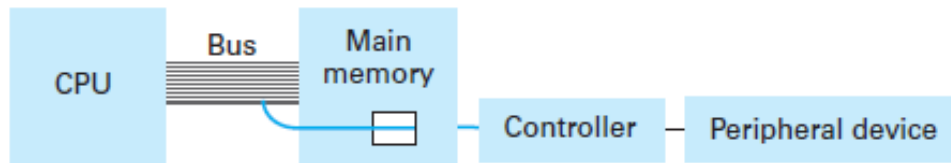
Denetleyicilerin rolü

- Denetleyiciler, veriyolu aracılığıyla bilgisayarla iletişim kurar



Denetleyicilerin rolü

- ▶ CPU ana bellekle haberleştiği şekilde denetleyicilerle de iki türde haberleşebilir.
- ▶ Giriş/çıkış komutları:
 - ▶ Denetleyiciye veri gönderirken **KAYDET benzeri**,
 - ▶ Denetleyiciden veri alırken **YÜKLE benzeri** komut kullanır
- ▶ Bellek haritalı giriş/çıkış komutları:
 - ▶ Her denetleyiciye **belli bir adres aralığı tahsis edilir** ve bu adresler ana bellek için kullanılmaz.
 - ▶ Bu yolla KAYDET ve YÜKLE komutları ana bellekteki gibi kullanılır.



Doğrudan bellek erişimi

- ▶ Denetleyicilerin **anabelleğe direkt olarak ulaşabilmelerine**; doğrudan bellek erişimi (Direct Memory Access - DMA) adı verilir.
- ▶ Örneğin işlemci sabit diskin bir bölümündeki veriyi anabelleğe aktarmak istiyorsa;
 - ▶ **İşlemci diskin denetleyicisine isteğini gönderir.**
 - ▶ **Disk denetleyicisi DMA sayesinde istenilen bilgiyi ana belleğe aktarır.**
- ▶ Böylece ikinci işlem için işlemci meşgul olmaz.

El sıkışma

- ▶ İki bilgisayar bileşeni arasındaki **haberleşme genellikle çift yönlüdür.**
- ▶ Örnek olarak **yazıcıdan bir çıktı alınmak istendiğinde;**
- ▶ Bilgisayar **yazıcının durumunu** öğrenmeden veri göndermez. Aksi durumda yazıcı veri aktarım hızına yetişemediği için veri kaybı yaşanacaktır.
- ▶ El sıkışma protokolleri ile bilgisayar ve çevresel birimler kendi **durum bilgilerini birbirleriyle paylaşırlar.**
- ▶ Paylaşılan durum bilgisine **durum sözcüğü** adı verilir.

Popüler haberleşme ortamları

- ▶ İki tür haberleşme vardır.
- ▶ **Paralel haberleşme:** her biri ayrı bir hat üzerinden aynı anda birden çok sinyal gönderir. (Örnek: bilgisayarın iç veri yolları)
- ▶ **Seri haberleşme:** sinyaller tek hat üzerinden arka arkaya gönderilir.
 - ▶ Birkaç metre için USB ve FireWire
 - ▶ Biraz daha uzun bağlantılar için Ethernet bağlantısı
 - ▶ Çok daha uzun bağlantılar için
 - ▶ Telefon hatları (Modem - DSL), Fiber optik kablolar.
 - ▶ Kablo modemler (Koaksiyel kablo ve Fiber optik kablolar)
 - ▶ Uydu hatları (Yüksek frekanslı radyo dalgaları)

Haberleşme Hızları

- ▶ bit/saniye cinsinden ölçülür.
- ▶ Kbps (1.000 bps), Mbps(1.000.000 bps)
- ▶ 1 KBps = 8 Kbps
- ▶ Geleneksel Telefon şebekeleri 57.6 Kbps bant genişliğine
- ▶ DSL 54 Mbps bant genişliğine sahiptir.

Veri işlemeyi programlamak

- ▶ Programlama dillerinin en önemli özelliklerinden biri de **makine dilinin detaylarıyla ilgilenmekten** kurtarmasıdır.
- ▶ Yüksek seviyeli dillerin tek bir komutu, makine dilinde bir komuta karşı gelebileceği gibi yüzlerce komuta da karşılık gelebilir.
- ▶ Bu çevirme işlemi **yorumlayıcı** aracılığıyla olur.
- ▶ Python da toplama için kullanılan operatörler, makine dilinde TOPLA komutuna karşılık gelir.
- ▶ Değişkenlere değer atama ise YÜKLE, Kaydet ve TAŞI komutlarıyla gerçekleşir.

Kontrol yapıları

```
if (water_temp > 140):  
    print('Bath water too hot!')
```

```
while (n < 10):  
    print(n)  
    n = n + 1
```

Fonksiyonlar

- ▶ `print()`, `str()`, `bin()` gibi..
- ▶ Matematik fonksiyonu gibi
- ▶ Parantez içine **argüman**lar virgülle ayrılark yazılır.
- ▶ Fonksiyon olarak yazılmış kodun yürütülmesine **fonksiyon çağırılması** denir.

$$f(x) = x^2 + 3x + 4.$$

$$f(5) = 5^2 + 3*5 + 4 = 25 + 15 + 4 = 44$$

```
x = 1034
```

```
y = 1056
```

```
z = 2078
```

```
biggest = max(x, y, z)
```

```
print(biggest)
```

- Bir değer geri döndüren fonksiyonlara **verimli fonksiyonlar** (fruitful functions) (örnek: `bin()`, `max()`), dönürmeyen fonksiyonlara **void fonksiyonlar** (örnek: `print()`) adı verilir.

```
x = print('hello world!')      # x is assigned None
```

Kütüphaneler

- Kütüphaneler normalde lazım olmayacak ancak ihtiyaç duyulduğunda çağrılacak birçok fonksiyon barındırır.

```
# Calculates the hypotenuse of a right triangle
import math

sideA = 3.0
sideB = 4.0
# Calculate third side via Pythagorean Theorem
hypotenuse = math.sqrt(sideA**2 + sideB**2)

print(hypotenuse)
```

Giriş ve Çıkış

```
echo = input('Please enter a string to echo: ')\nprint(echo * 3)
```

- ▶ Girilen mesajı 3 kere ekrana yazdırır.

```
# Calculates the hypotenuse of a right triangle
import math

# Inputting the side lengths, first try
sideA = input('Length of side A? ')
sideB = input('Length of side B? ')
# Calculate third side via Pythagorean Theorem
hypotenuse = math.sqrt(sideA**2 + sideB**2)

print(hypotenuse)
```

```
hypotenuse = math.sqrt(sideA**2 + sideB**2)  
TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
```

```
# Calculates the hypotenuse of a right triangle  
import math
```

```
# Inputting the side lengths, with integer conversion  
sideA = int(input('Length of side A? '))  
sideB = int(input('Length of side B? '))  
# Calculate third side via Pythagorean Theorem  
hypotenuse = math.sqrt(sideA**2 + sideB**2)  
  
print(hypotenuse)
```


Maraton idman asistanı

Time Per Mile				Total Elapsed Time	
Minutes	Seconds	Miles	Speed (mph)	Minutes	Seconds
9	14	5	6.49819494584	46	10
8	0	3	7.5	24	0
7	45	6	7.74193548387	46	30
7	25	1	8.08988764044	7	25



```
# Marathon training assistant.
import math

# This function converts a number of minutes and seconds into just seconds.
def total_seconds(min, sec):
    return min * 60 + sec

# This function calculates a speed in miles per hour given
# a time (in seconds) to run a single mile.
def speed(time):
    return 3600 / time

# Prompt user for pace and mileage.
pace_minutes = int(input('Minutes per mile? '))
pace_seconds = int(input('Seconds per mile? '))
miles = int(input('Total miles? '))

# Calculate and print speed.
mph = speed(total_seconds(pace_minutes, pace_seconds))
print('Your speed is')
print(mph)

# Calculate elapsed time for planned workout.
total = miles * total_seconds(pace_minutes, pace_seconds)
elapsed_minutes = total // 60
elapsed_seconds = total % 60

print('Your total elapsed time is')
print(elapsed_minutes)
print(elapsed_seconds)
```



Ders bitti

Erciyes Üniversitesi
Selçuk Üniversitesi
Sakarya Üniversitesi
Kahramanmaraş Sütçü İmam Üniversitesi
ders notları kaynak ve içerik olarak kullanılmıştır.