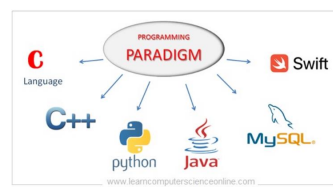
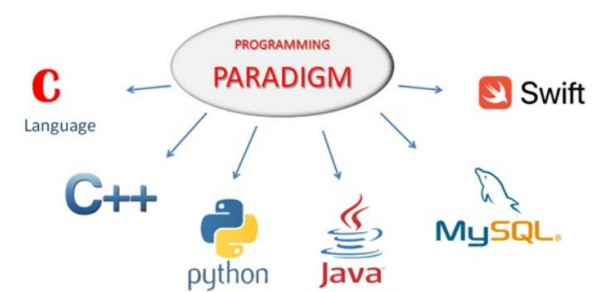
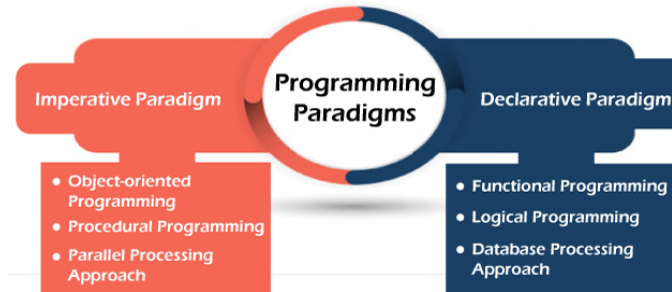
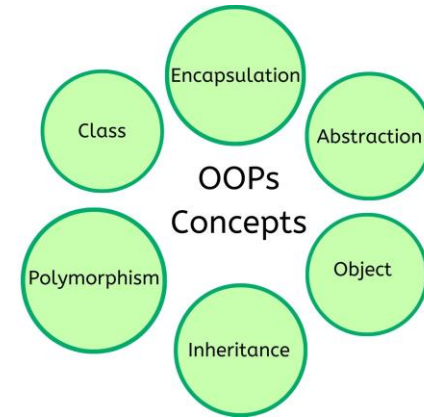
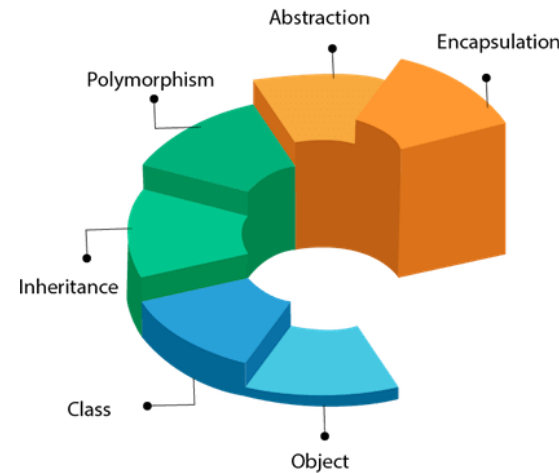




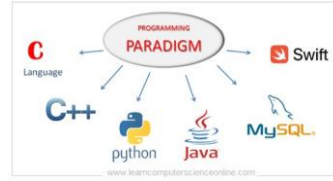
# Programlama Paradigmaları



*Yaklaşımınızı yansıtan en iyi alıntı...*  
*"Bir insan için küçük bir adım, insanlık için dev bir sıçrama."*

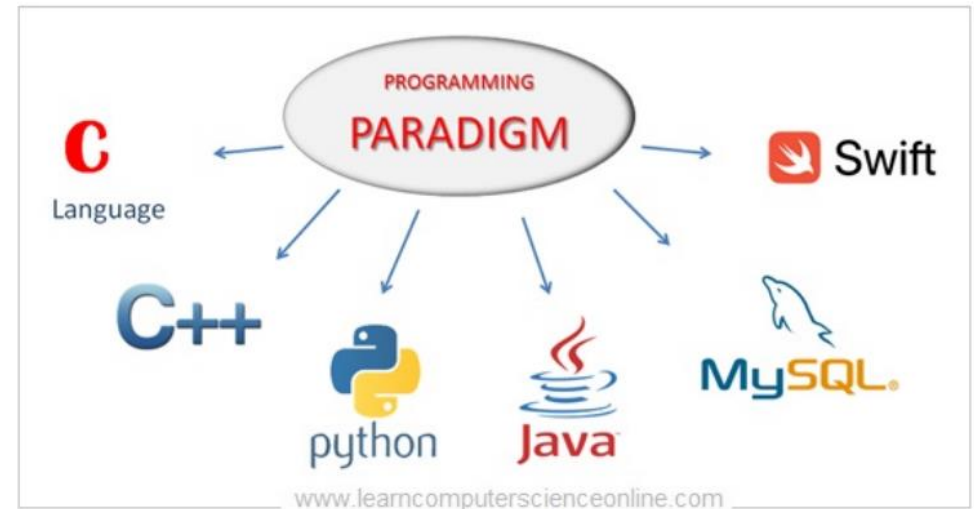
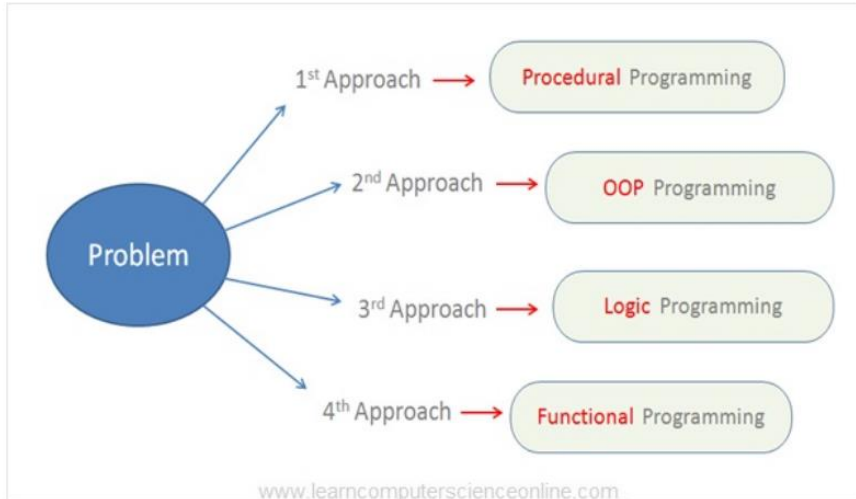
- NEİL ARMSTRONG

# Programlama Paradigmaları



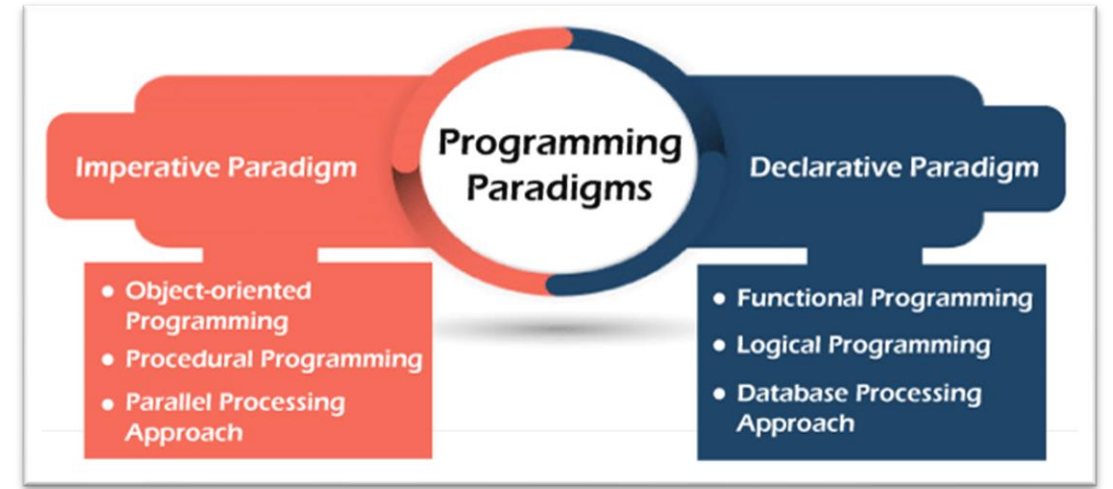
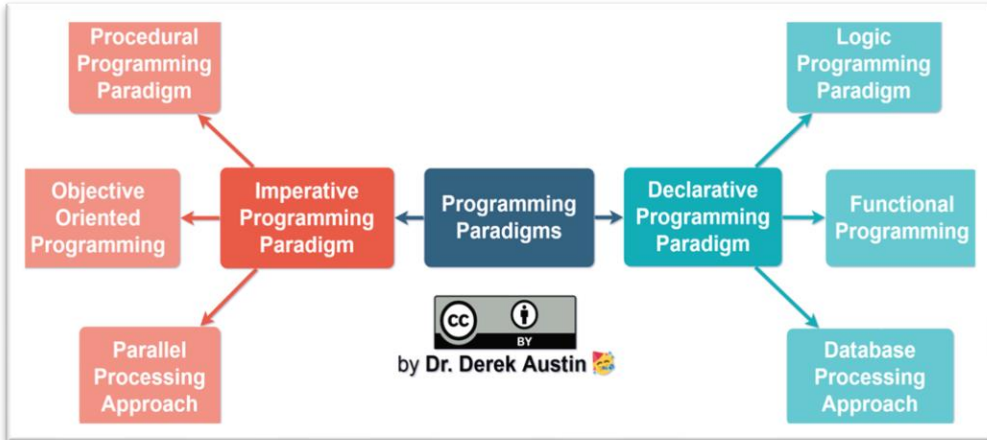
**Paradigma** , bazı problemleri çözme veya bazı görevleri yerine getirme yöntemi olarak da adlandırılabilir.

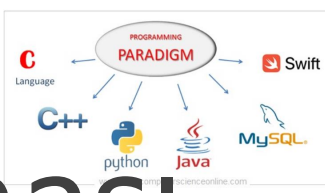
**Programlama paradigması**, bazı programlama dillerini kullanarak problemi çözmeye yönelik bir yaklaşımdır veya aynı zamanda, bir yaklaşımı takip ederek bizim için mevcut olan araç ve teknikleri kullanarak bir problemi çözme yöntemidir diyebiliriz.



# Programlama Paradigmaları

Programlama dili için bilinen çok şey var ama hepsinin uygulanması sırasında bir strateji izlemesi gerekiyor ve bu metodoloji/strateji paradigmalardır. Programlama dili çeşitlerinin yanı sıra, her talebi yerine getirmek için birçok paradigma vardır.



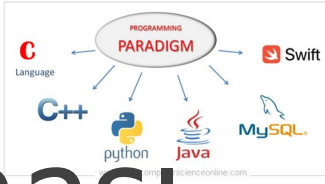


# 1. Zorunlu Programlama Paradigması

## 1. Zorunlu Programlama Paradigması(Imperative Programming Paradigm):

En eski programlama paradigmlarından biridir. Makine mimarisiyle yakın ilişki içindedir. Von Neumann mimarisine dayanmaktadır. Atama ifadeleri aracılığıyla program durumunu değiştirerek çalışır. Durum değiştirerek adım adım görevi gerçekleştirir. Ana odak, hedefe nasıl ulaşılacağıdır. Paradigma birkaç ifadeden oluşur ve tüm sonuçların yürütülmesinden sonra saklanır.





# 1. Zorunlu Programlama Paradigması

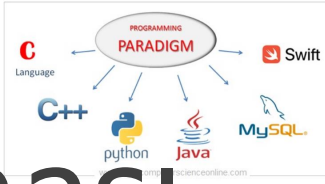
---

## Avantajları:

- Uygulaması çok basit
- Döngüler, değişkenler vb. içerir.

## Dezavantajları:

- Karmaşık sorun çözülemez
- Daha az verimli ve daha az üretken
- Paralel programlama mümkün değil



# 1. Zorunlu Programlama Paradigması

---

Zorunlu programlama üç geniş kategoriye ayrılır:

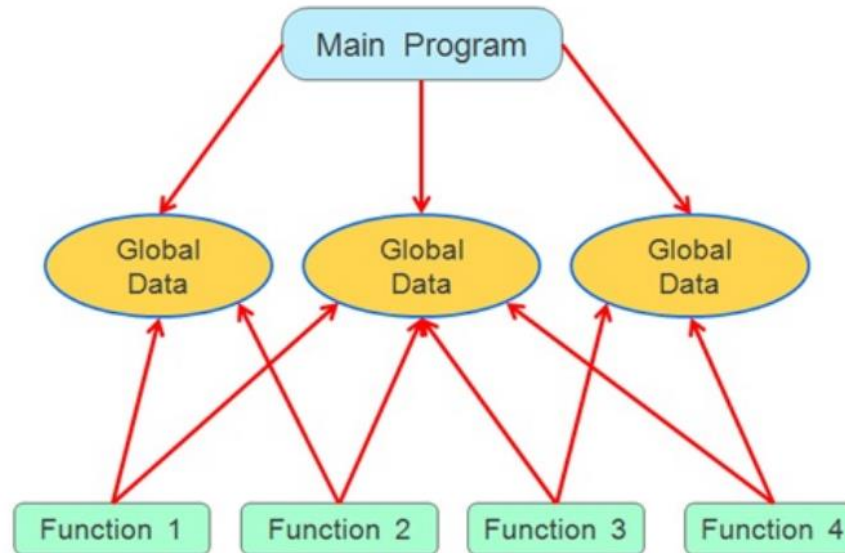
- a. Prosedürel Programlama Paradigması (Procedural Programming Paradigm)
- b. Nesne Yönelimli Programlama (Object Oriented Programming)
- c. Paralel İşleme Yaklaşımı (Parallel Processing Approach)

# 1. Zorunlu Programlama Paradigması

Bu paradigmlar şu şekildedir:

**1.a Prosedürel Programlama Paradigması** :Bu paradigma, altta yatan makine modeli açısından prosedürü vurgular. Prosedürel ve zorunlu yaklaşım arasında hiçbir fark yoktur. Kodu yeniden kullanma yeteneğine sahiptir.

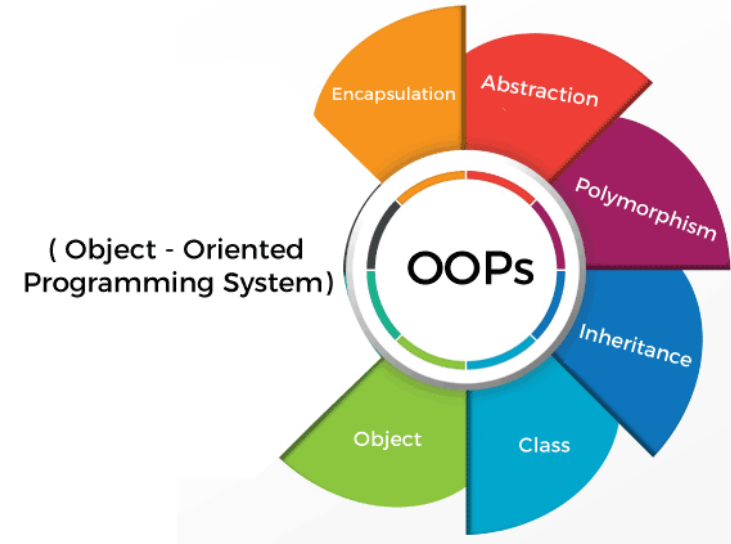
Prosedürel paradigmaya örnek; C, C++, Java, ColdFusion, Pascal





# 1. Zorunlu Programlama Paradigması

**1.b Nesne Yönelimli Programlama :** Program, iletişim amaçlı sınıflar ve nesneler topluluğu olarak yazılmıştır. En küçük ve temel varlık nesnedir ve her türlü hesaplama yalnızca nesneler üzerinde yapılır. Prosedürden ziyade veriye daha fazla vurgu yapılır. Bugün senaryoda olan hemen hemen her türlü gerçek hayat probleminin üstesinden gelebilir.

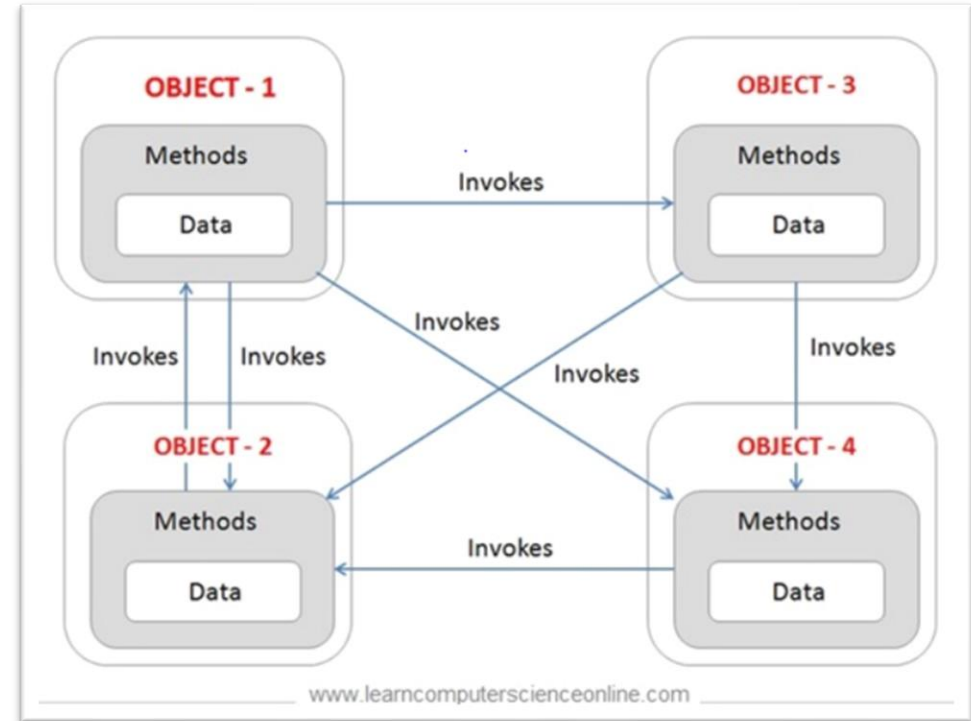


# 1. Zorunlu Programlama Paradigması

**1.b Nesne Yönelimli Programlama :** OOP örnek: Simula, Java, C++, C#, Python, Ruby, Smalltalk, Objective-C, VB.Net

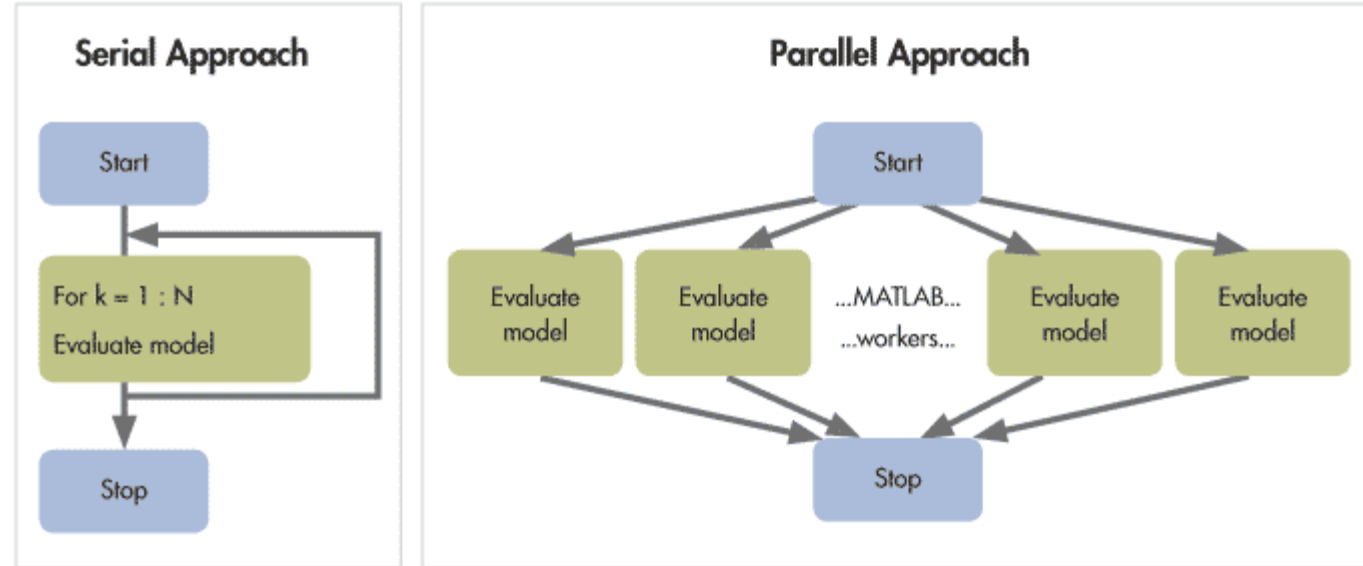
## Avantajlar:

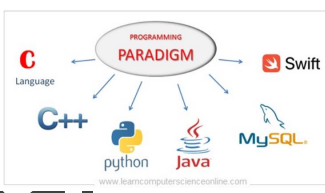
- Veri güvenliği
- Miras
- Kod yeniden kullanılabilirliği
- Esneklik ve soyutlama da mevcuttur



# 1. Zorunlu Programlama Paradigması

**1.c Paralel İşleme Yaklaşımı:** Paralel işleme, program talimatlarının birden çok işlemci arasında bölünerek işlenmesidir. Paralel bir işlem sistemi, bir programı bölerek daha kısa sürede çalıştırmak amacıyla çok sayıda işlemciye sahiptir. Bu yaklaşım böl ve yönet gibi görünüyor. Örnekler, MATLAB , NESL'dir (en eskilerinden biri) ve C/C++ da bazı kitaplık işlevleri nedeniyle destekler.





# 2. Bildirimsel Programlama Paradigması

## 2. Bildirimsel Programlama Paradigması (Declarative Programming Paradigm):

- a .Mantıksal,
- b. İşlevsel,
- c. Veritabanı olmak üzere üçe ayrılır.

Bilgisayar biliminde bildirimsel programlama , kontrol akışından bahsetmeden hesaplama mantığını ifade eden bir program oluşturma tarzıdır.

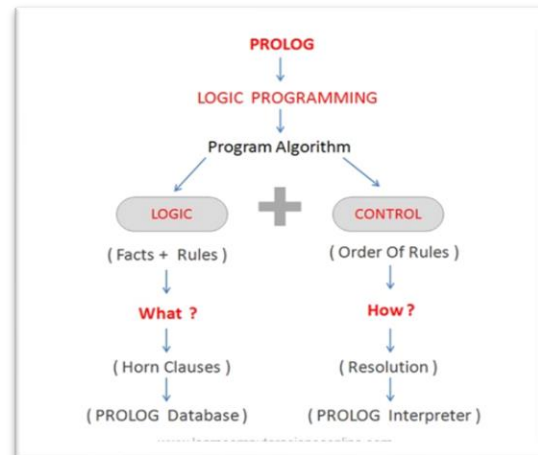
Programları genellikle bir mantığın teorileri olarak kabul eder. Paralel programlar yazmayı basitleştirebilir. Odak noktası, yapılması gerekenlerden ziyade nasıl yapılması gerektiğidir, temel olarak kod kodunun gerçekte ne yaptığını vurgular.

Sadece nasıl üretildiğini değil, istediğimiz sonucu bildirir. Zorunlu (nasıl yapılır) ve bildirimsel (ne yapılır) programlama paradigmaları arasındaki tek fark budur.

# 2. Bildirimsel Programlama Paradigması

## 2. a Mantıksal Programlama Paradigması (Logic Programming Paradigms):

Soyut bir hesaplama modeli olarak adlandırılabilir. Mantıksal programlamada daha önce bildiğimiz bir bilgi tabanımız var ve makineye verilen soru ve bilgi tabanı ile birlikte sonuç üretiyor. Normal programlama dillerinde böyle bir bilgi tabanı kavramı mevcut değildir ancak yapay zeka, makine öğrenimi kavramını kullanırken aynı mekanizmayı kullanan Algı modeli gibi bazı modellerimiz vardır. Mantıksal programlamada asıl vurgu bilgi tabanı ve problem üzerindedir. Programın yürütülmesi, matematiksel ifadenin kanıtına çok benzer, örneğin, Prolog



```
sum of two number in prolog:

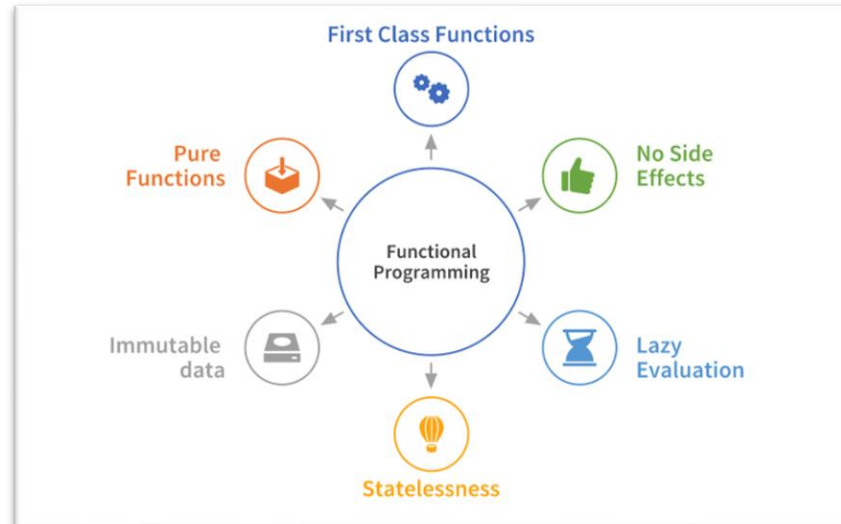
predicates
sumoftwonumber(integer, integer)
clauses

sum(0, 0).
sum(n, r):-
    n1=n-1,
    sum(n1, r1),
    r=r1+n
```

# 2. Bildirimsel Programlama Paradigması

## 2.b Fonksiyonel Programlama Paradigması (Functional Programming Paradigm):

Fonksiyonel programlama paradigmalarının kökleri matematiktedir ve dilden bağımsızdır. Bu paradigmaların temel ilkesi, bir dizi matematiksel fonksiyonun yürütülmesidir. Soyutlama için merkezi model, veri yapısı değil, bazı özel hesaplamalar için amaçlanan işlevdir. Veriler, işlevlere gevşek bir şekilde bağlıdır. İşlev, uygulamalarını gizler. Fonksiyon, programın anlamını değiştirmeden değerleriyle değiştirilebilir. Perl, Javascript, Scala, Lisp, Haskell, ML, Clojure gibi bazı diller çoğunlukla bu paradigmayı kullanır.





## 2. Bildirimsel Programlama Paradigması

### 2.c Veritabanı İşleme Yaklaşımı (Database Proecessing Approach)

Bu programlama metodolojisi, verilere ve hareketine dayanmaktadır. Program ifadeleri, bir dizi adımı sabit kodlamak yerine verilerle tanımlanır. Bir veritabanı programı, bir iş bilgi sisteminin kalbidir ve dosya oluşturma, veri girişi, güncelleme, sorgulama ve raporlama işlevleri sağlar. Çoğunlukla veritabanı uygulaması için geliştirilmiş birkaç programlama dili vardır. Örneğin SQL. Filtreleme, dönüştürme, toplama (bilgi işlem istatistikleri gibi) veya diğer programları çağırmak için yapılandırılmış veri akışlarına uygulanır. Bu yüzden kendi geniş uygulaması vardır.

