

- 1- Bilgisayarda işlemcinin kendine özgü bir dili vardır ancak bu dili kullanmak ve çevirmek çok zordur. İnsanların konuştuklarını işlemci diline çeviren yazılımlarda derleyici denir. Assembler ise assembly dilini makina koduna çeviren programlardır. Bu ikisi arasındaki fark; derleyici, bir dildeki kodu başka bir dile çevirir; assembler ise assembly dilini makine diline çevirir.
- 2-
 - a- Emirsel programlama, takip edildiğinde veriyi işleyecek ve istenen sonucu üretecek bir dizi komutun geliştirilmesi için yapılacak programlama sürecini tanımlar. Sonra problemi çözmek için bir algoritma bulup ifade ederek programlama süreci yaklaşımını anlatır.
Bildirimsel programlama, emirsel programlamanın zıttıdır. Daha çok çözülecek problemi tanımlamaya yöneliktir.
Fonksiyonel programlamanın işlevi ise fonksiyonların daha basit fonksiyonlarla iç içe daha karmaşık yapıya dönüştürülerek oluşturulmasıdır.
 - b- Fonksiyonel: Lisp, Haskell, Common Lisp
Emirsel: C, F#
Bildirimsel: Prolog, SQL,
Nesne Yönelimli: C#, Java
 - c- SQL: SQL aslında verileri yönetmek ve onları tasarlamak için kullanılan bir alt dildir; programlama dili değildir. Tarihte veritabanı kavramının çıkmasıyla beraber bu veritabanındaki verileri işleyecek “veri sorgulama dili” ne ihtiyaç duyulmuştur. İlk başta IBM, SQUARE diye bir dil geliştirmiştir fakat farklı sorunlardan dolayı bunun yerini SQL (ilk adı SEQUEL) almıştır. Bu dil ile veritabanındaki verilere yeni veriler ekleyebilir, verileri güncelleyebilir, silebilir hatta istenilen şekilde listeleyebiliriz.

COMMON LISP: Common Lisp, 1980’li yıllarda Bob Engelmores ve Guy Lewis Steele Jr. tarafından geliştirilmiş bir dildir. Common Lisp genel amaçlı, çok paradigma bir programlama dilidir. Bu dil en az sınırlayıcı syntaxlara ve basit ve esnek bir yazılıma sahiptir. Ayrıca dinamik olup bütün girdiler her yönüyle değiştirilebilir ve hatalara yer vermez, Lisp makroları (kod dönüşümleri) ve okuyucu makroları (karakterler için giriş ayrıştırıcıları) gibi standart özellikler aracılığıyla genişletilebilir, Lisp’e geriye dönük uyumluluk sağlar.
- 3- A- 48
B- 9
C- 13
D- 0
- 4- **(topla(carp 0.6 (bol (liste_topla girdi_listesi) 2)(carp final_notu 0.4))**
- 5- faturalar=elektrik+doğalgaz+su+telefon+internet
yeni_bakiye=eski_bakiye+maas
kalan_bakiye=yeni_bakiye-faturalar