KOCAELİ ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

Programlama Dilleri Prensipleri

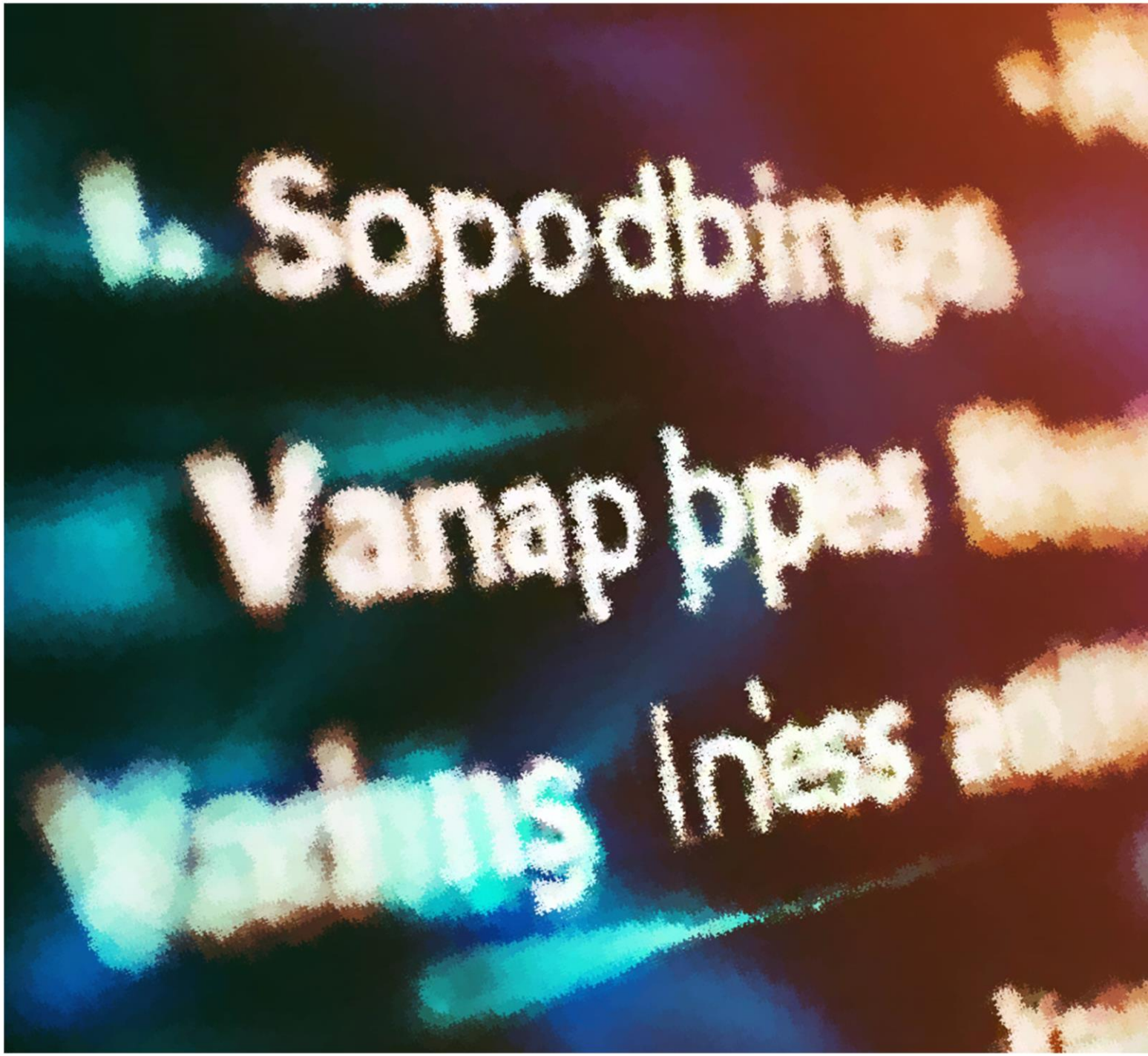
Programming Language Concepts

http://yapbenzet.kocaeli.edu.tr/

Dr. Öğr. Üyesi A. Burak İNNER

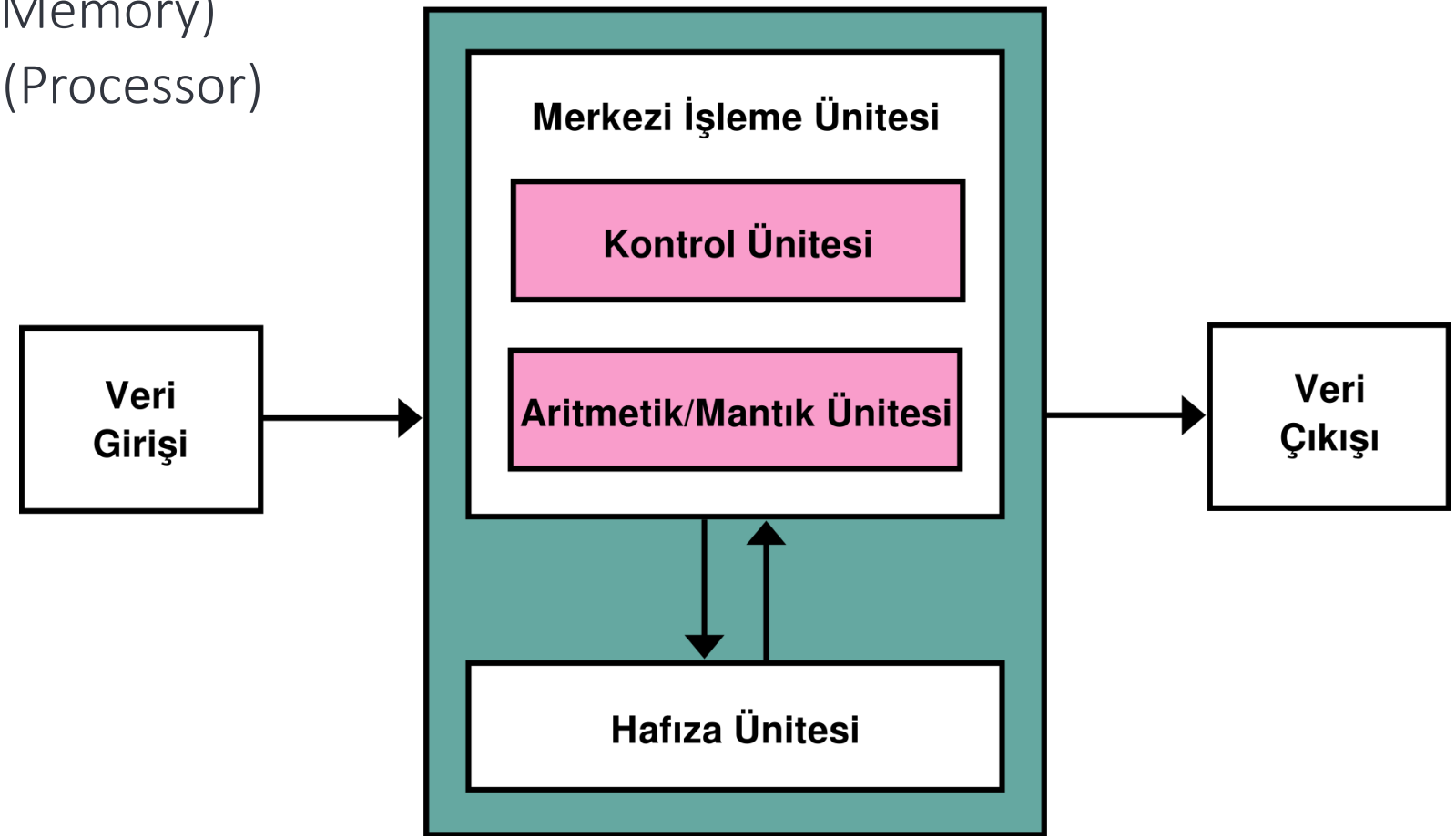
HAFTA 05

Names , Variables , Bindings, and Scopes



Giriş

- Von Neumann mimarisi, her bellek hücresinin özgün bir adres ile tanımlandığı ana bellek kavramına dayanmaktadır.
  - Hafıza (Memory)
  - İşlemci (Processor)



GİRİŞ (devam)

- Bir bellek hücresinin içeriği, bir değerin belirli bir yönteme göre kodlanmış gösterimidir.
- Bu içerik, programların çalışması sırasında okunabilir ve değiştirilebilir.
- Imperative programlama, von Neumann mimarisindeki bilgisayarlara uygun olarak programların işlem deyimleri ile bellekteki değerleri değiştirmesine dayanır.

Bir değişken, bir veya daha çok bellek hücresinin soyutlamasıdır.

DEĞİŞKEN ÖZELLİKLERİ	
İSİM	okunabilirlik
ADRES	Aliasing
DEĞER	Belllekte belirli yönteme göre kodlanmış
TİP	Tip uyuşmazlığı, tip dönüşümü
YAŞAM SÜRESİ	Bellekle ilişkili kaldığı süre
KAPSAM	Geçerli olduğu deyimler

İsimler

- İsimler, programlama dillerinde, değişkenlerin yanısıra, etiketler, altprogramlar, parametreler gibi program elemanlarını tanımlamak için kullanılırlar.
- Name: Değişkenleri, sabitleri, fonksiyonları, türleri, işlemleri vb. ifade etmemizi sağlayan tanımlayıcılar

İsimler

- İsimleri tasarlamak için programlama dillerinde

➢EN FAZLA UZUNLUK
➢BÜYÜK-KÜÇÜK HARF DUYARLILIĞI
➢ÖZEL KELİMELER

- gibi farklı yaklaşımlar uygulanmaktadır

En Fazla Uzunluk

- Programlama dillerinde bir ismin en fazla kaç karakter uzunluğunda olabileceği konusunda farklı yaklaşımlar uygulanmıştır.

Önceleri programlama dillerinde bir isim için izin verilen karakter sayısı daha sınırlı iken, günümüzdeki yaklaşım, en fazla uzunluğu kullanışlı bir sayıyla sınırlamak ve çoklu isimler oluşturmak için altçizgi "_" karakterini kullanmaktır

En Fazla Uzunluk

Programlama Dili	İzin verilen Maksimum isim uzunluğu
FORTRAN I	maksimum 6
COBOL	30
FORTRAN 90	31
Ada	limit yoktur, ve hepsi anlamlıdır(significant)
Java	limit yoktur, ve hepsi anlamlıdır(significant)
ANSI C	sınır yok ama yalnızca ilk 63 önemli; ayrıca, harici adlar maksimum 31 ile sınırlıdır
C++	limit yoktur fakat konabilir

Küçük-Büyük Harf Duyarlılığı (Case Sensitivity)

- Birçok programlama dilinde, isimler için kullanılan küçük ve büyük harfler arasında ayrım yapılmazken
- Bazı programlama dilleri (Örneğin; C, C++, Java) isimlerde küçük-büyük harf duyarlılığını uygulamaktadır
- Bu durumda, aynı harflerden oluşmuş isimler derleyici tarafından farklı olarak algılanmaktadır

- *TOPLAM, toplam,* ve *ToPlaM*, üç ayrı değişkeni göstermektedir

Names … Özel karakterler

- PHP: tüm değişken isimleri dolar ($) işaretleriyle başlamalıdır
- Perl: tüm değişken adları, değişkenin türünü belirten özel karakterlerle başlar
- Ruby: @ ile başlayan değişken isimleri örnek değişkenlerdir; @@ ile başlayanlar sınıf değişkenleridir

Özel Kelimeler

- Özel kelimeler, bir programlama dilindeki temel yapılar tarafından kullanılan kelimeleri göstermektedir.
- A)Anahtar kelimeler(Keywords) : bir dilde özel bir anlama sahiptir ve sözdiziminin bir parçasıdır.

- B)Ayrılmış kelimeler(Reserved words): dil tarafından ayrıldıkları için tanımlayıcı (değişkenler, işlevler vb.) olarak kullanılamayan sözcüklerdir.

- Java'da, goto ayrılmış bir kelimedir, ancak bir anahtar kelime değildir (sonuç olarak, onu hiç kullanamazsınız)
- Fortran'da ayrılmış kelimeler yoktur, tüm anahtar kelimeler (eğer, o zaman, vb.) tanımlayıcı olarak kullanılabilir

Anahtar Kelime

- Bir anahtar kelime (*keyword*), bir programlama dilinin sadece belirli içeriklerde özel anlam taşıyan kelimelerini göstermektedir.
- Örneğin FORTRAN'da *REAL* kelimesi, bir deyimin başında yer alıp, bir isim tarafından izlenirse, o deyimin tanımlama deyimi olduğunu gösterir. (*REAL apple)*
- Eğer *REAL* kelimesi, atama işlemcisi "=" tarafından izlenirse, bir değişken ismi olarak görülür. *REAL = 10.05* gibi.
- Bu durum dilin okunabilirliğini azaltır.

- if if then then else else

Ayrılmış Kelime:

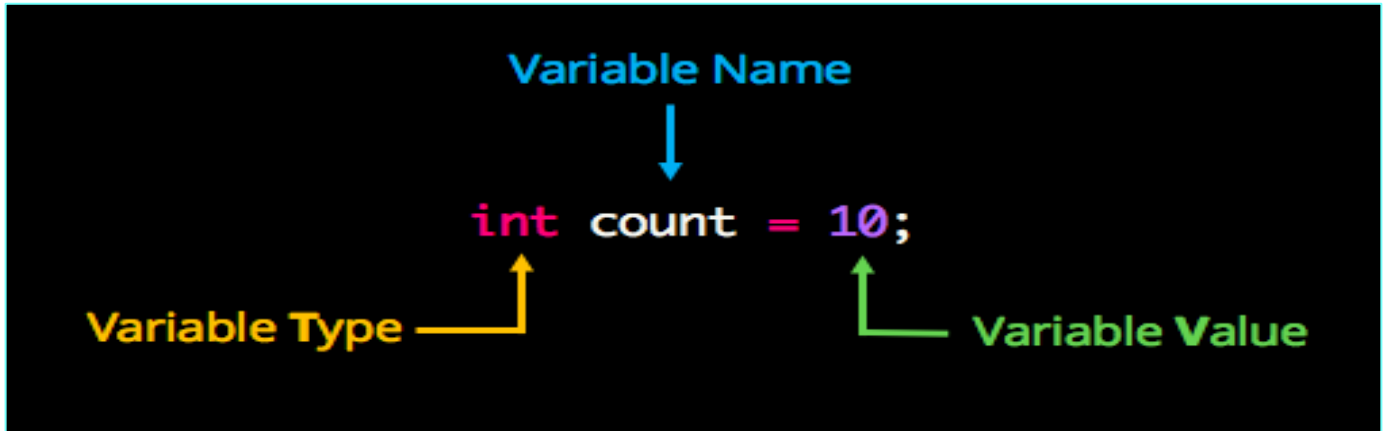
- Öte yandan, ayrılmış kelime (*reserved word*), bir programlama dilinde bir isim olarak kullanılamayacak özel kelimeleri göstermektedir.

- C++ dilindeki do, for , while gibi ve
- PASCAL'da procedure, begin, end gibi kelimere ayrılmış kelime denir.

Veri Tipi Kavramı

- Bir **veri tipi**, aynı işlemlerin tanımlı olduğu değerler kümesini göstermektedir.
- Bir değişkenin tipi, değişkenin tutabileceği değerleri ve o değerlere uygulanabilecek işlemleri gösterir.
- Örneğin; tamsayı (*integer*) tipi, dile bağımlı olarak belirlenen en küçük ve en büyük değerler arasında tamsayılar içerebilir ve sayısal işlemlerde yer alabilir.

- Veri tipleri, programlama dillerinde önemli gelişmelerin gerçekleştiği bir alan olmuş ve bunun sonucu olarak, programlama dillerinde çeşitli veri tipleri tanıtılmıştır.
- Tipler *basit tipler* ve bileşik tipler olarak gruplandırılabilir.



Veri Tipi Kavramı

- Basit tipler
- İlkel tipler, çoğu programlama dilinde yer alan ve diğer tiplerden oluşmamış veri tiplerini göstermektedir.

- Tam sayı, Mantıksal, Karakter, Karakter katarı, Kullanıcı tanımlı tipler örnek olarak verilebilir.

- Bileşik Tipler

- Çeşitli veri tiplerinde olabilen bileşenlerden oluşmuştur.

- Diizi, Kayıt, Pointer

Tablo Temel C++ değişken tipleri

	Numerik aralık			Bellek alanı
Keyword	Alt sınır	Üst sınır	Ondalık kısım	byte
char	–128	127	yok	1
short	–32,768	32,767	yok	2
int	–2,147,483,648	2,147,483,647	yok	4
long	–2,147,483,648	2,147,483,647	yok	4
float	3.4 x 10–38	3.4 x 1038	7	4
double	1.7 x 10–308	1.7 x 10308	15	8
long double	3.4 x 10–4932	1.1 x 104932	19	10

SABİTLER

- Bir **sabit**, belirli bir tipteki bir değerin kodlanmış gösterimini içeren ancak programın çalıştırılması sırasında değiştirilemeyen bellek hücresine veya hücrelerine verilen isimdir.
- isimlendirilmiş sabit
- Okunabilirliğe olumlu katkı
- Pascal'da *const* ve C'de *#define* kullanılır

İŞLEMCİLER

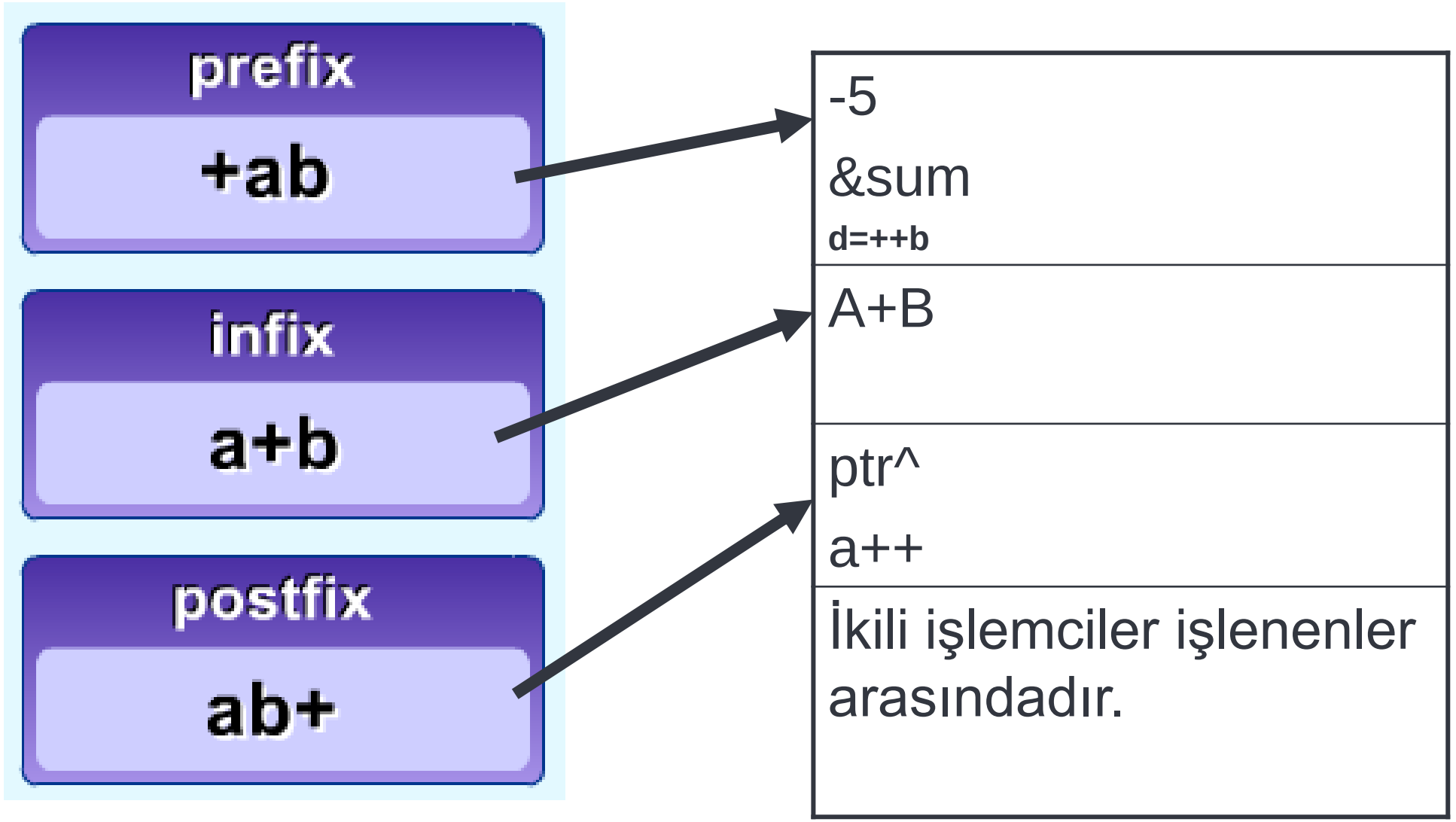
Genel Özellikler	İşlenen sayısı
	İşlemcinin yeri
	İşlem önceliği
	Birleşme Özelliği
İşlenenlere (Niteliğine) göre	Sayısal işlemciler
	İlişkisel işlemciler
	Mantıksal işlemciler

Tekli işlemler

- Yalnızca tek değişkenlere uygulanırlar..

```
i = +1;
j = -i;
```

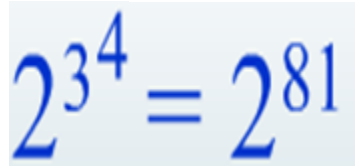
İşlemcinin yeri (devam)



Öncelik

	FORTRAN	PASCAL	C	ADA
Enyüksek öncelik	** (exponentation)	*, / , div , mod	++, -- (postfix)	** , abs
	* , /	+ , -	++, -- (prefix)	*, /, mod
	+ , -		Tekli (unary) +, -	Tekli(unary) +, -
			*, /, %	İkili(Binary)+, -
			İkili(Binary)+, -	
En düşük öncelik				

Associativity



Programlama Dili	Birleşme Kuralı	İşlemciler
FORTRAN	Sol Birleşmeli	*, /, +, -
	Sağ Birleşmeli	**
PASCAL	Sol Birleşmeli	Bütün işlemciler
	Sağ Birleşmeli	
C	Sol Birleşmeli	Postfix++, postfix--, *, /, %, ikili +, ikili-
	Sağ Birleşmeli	Prefix++, pretfix--,tekli +, tekli-
C++	Sol Birleşmeli	*, /, %, ikili +, ikili-
	Sağ Birleşmeli	++, --,tekli +, tekli-
ADA	Sol Birleşmeli	** dışındakiler
	Sağ Birleşmeli	** birleşme özelliği yok

NİTELİĞİNE GÖRE İŞLEMCİLER

SAYISAL İŞLEMCİLER

sembol	İşlev	formül	sonuç
*	Çarpma	4*2	8
/	Bölme ve tamsayı bölme	64/4	16
%	Modul veya kalan	13%6	1
+	Toplama	12+9	21
-	Çıkarma	80-15	65

İLİŞKİSEL İŞLEMCİLER

Anlamı	C++	PASCAL
Büyüktür	>	>
Küçüktür	<	<
Eşittir	= =	=
Eşit değildir	!=	<>
Büyük veya eşititr	>=	>=
Küçüktür veya eşititr	<=	<=

MANTIKSAL İŞLEMCİLER

C dili Mantıksal İşlemcileri		
&&	VE	AND
||	VEYA	OR
!	DEĞİL	NOT

İşlemcilerin öncelikleri

- Sayısal ifadeler, ilişkisel ifadelerin işlenenleri olabileceği ve ilişkisel ifadeler de Boolean ifadelerin işlenenleri olabileceği için, üç işlemci grubunun kendi aralarında öncelikleri vardır.
- İlişkisel işlemcilerin önceliği, her zaman sayısal işlemcilerden düşüktür.
  - X+20<= k*2

- İlişkisel ifadeler ise mantıksal ifadeler için bir operand olabileceğinden ilişkisel ifadeler mantıksal ifadelerden önce yapılmalıdır.

ADA Programlama Dili	
En yüksek	**, abs, not
	*, /, mod, rem
	+, - (tekli)
	+, -, & (ikili)
	=, /=, >, <, <=, >=, in, not in
Endüşük	AND, OR, XOR, AND THEN, OR ELSE

(A<B) and (A>C) or X=0
Doğru

A<B and A>C or X=0 Yanlış

İşlemci Yükleme

- İşlemcilerin anlamlarının, işlenenlerin sayısına ve tipine bağlı olarak belirlenmesine **işlemci yüklemesi** (*operator overloading*) denir.
- "+", hem tamsayı hem de kayan-noktalı toplama için kullanılır ve bazı dillerde, sayısal işlemlere ek olarak karakter dizgilerin birleştirilmesi için de kullanılır.
- Okunabilirlik zayıflıyor.

İFADELER

- İfadeler, yeni değerler oluşturmak için değerleri ve işlemcileri birleştirmeye yarayan sözdizimsel yapılardır.
- Bir ifade, bir sabit, bir değişken, bir değer döndüren bir fonksiyon çağrımı veya bir işlemciden oluşabilir.
    - SAYISAL İFADELER (a+b*c)
    - İLİŞKİSEL İFADELER (a>=b)
    - MANTIKSAL İFADELER ((A>10) and (C<2))

DEYİMLER

- **Deyimler**, bir programdaki işlemleri göstermek ve akışı yönlendirmek için kullanılan yapılardır.
- **Basit Deyimler(**atama deyimi)
- **Birleşik Deyimler (**(*if-then- else* ve *case* deyimleri ve *while* ve *for* deyimleri gibi

Atama İşlemi

- Programlama dillerinde atama sembolünün anlamı, sağ taraftaki değerin sol taraftaki değişkene aktarılmasıdır.

- =, :=, ==

Atama deyimi

- <hedef_değişken> <atama_işlemcisi> <ifade>

Sum=++count	Count=count+1 Sum=count
Sum, total=0	Sum=0 ve total=0
puan+=500;	Puan=Puan+500

Sum=count ++	Sum=count Count=count+1
count ++	count+1
F ? count1:count2=0	F=1 ise count1=0 F=0 ise count2=0

Özet

- Temel programlama dili elemanları olarak **değişkenler, sabitler, işlemciler, ifadeler ve deyimler** incelenmiştir.
- İşlemcilerin sınıflandırılmış (sayısal işlemciler, ilişkisel işlemciler, mantıksal işlemciler) ve işlemci yükleme kavramı açıklanmıştır.
- Atama deyimi farklı sözdizimlerle örneklendirilmiştir.