

C Programlama

Öğretim Elemanı Bilgileri

- Dr. Öğr. Üyesi Sema ATASEVER
- Nevşehir Hacı Bektaş Veli Üniversitesi, Mühendislik Mim.Fak. Bilgisayar Mühendisliği
- Web sayfası : <https://biz.nevsehir.edu.tr/sema/tr>
- Email : sema@nevsehir.edu.tr | s.atasever@gmail.com

Ölçme Yöntemi

ARA SINAV

- Ara sınav : 100 puan üzerinden değerlendirilecektir , Katkı : %40

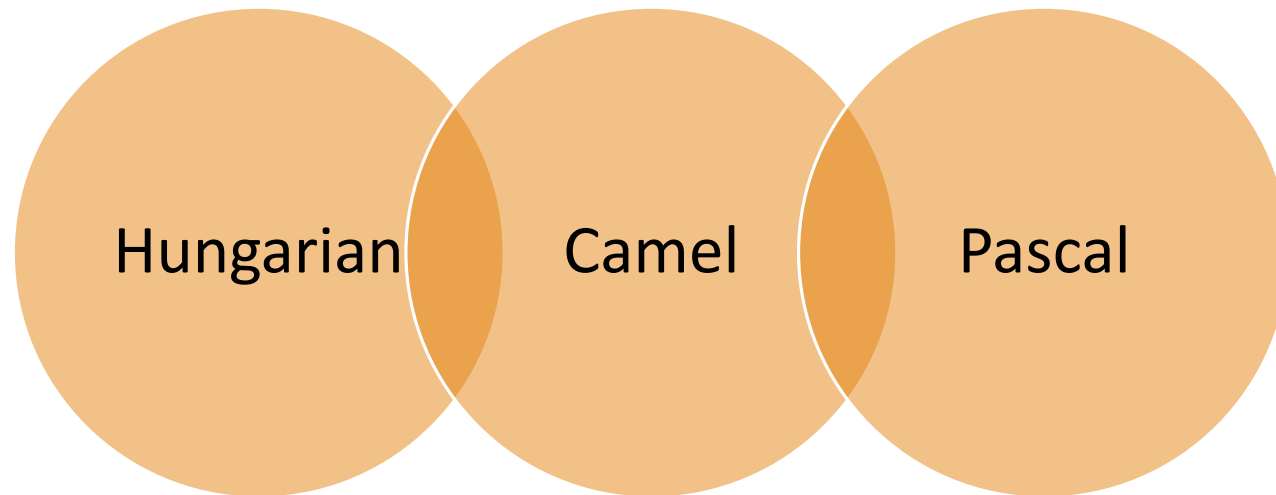
FİNAL ÖDEVİ

- Final Sınavı : 100 puan üzerinden değerlendirilecektir , Katkı : %60
- Nihai ders notu hesabı : Ara sınavın %40'ı, Final notunun %60'ı alınarak hesaplanmaktadır!

10.Hafta Konuları

- Kod Gösterimi (Code Notation), İşaretçiler (pointers) örnek kod uygulamaları ile tekrar.

Kod Gösterimi (Code Notation)



Macar Notasyonu | Hungarian Notation

- Charles Simonyi, Hungarian Notation'ı gündeme getiren ilk kişi oldu.
- Adında **değişkenle ilgili bilgileri** de içeren bir değişken adlandırma kuralıdır.

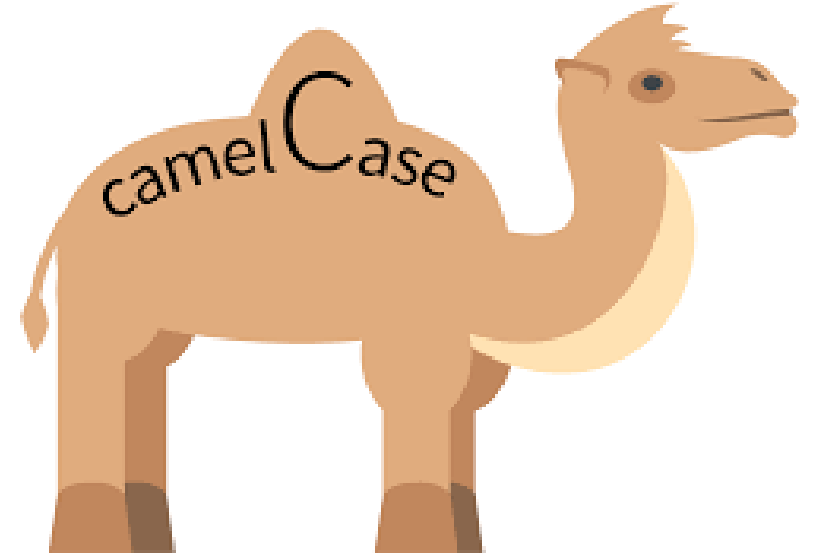
Macar Notasyonu | Hungarian Notation

Prefix	Type	Example
b	boolean	bool bStillGoing;
c	character	char cLetterGrade;
str	C++ String	string strFirstName;
si	short integer	short siChairs;
i	integer	int iCars;
li	long integer	long liStars;
f	floating point	float fPercent;
d	double-precision floating point	double dMiles;
ld	long double-precision floating point	long double ldLightYears;
sz	Old-Style Null Terminated String	char szName[NAME_LEN];
if	Input File Stream	ifstream ifNameFile;
is	Input Stream	void fct(istream &risIn);
of	Output File Stream	ofstream ofNameFile;
os	Output Stream	void fct(ostream &rosIn);
S	declaring a struct	struct SPoint {
C	declaring a class	class CPerson {
struct name or abbrev	declaring an instance of a struct	SPoint pointLeft; SPoint ptLeft; // or abbrev. (be consistent)
class name or abbrev	declaring an instance of a class	CPerson personFound; CPerson perFound; // or abbrev. (be consistent)

Pre-prefix	Type	Example
u	unsigned	unsigned short usiStudents;
k	constant formal parameter	void fct(const long kliGalaxies)
r	reference formal parameter	void fct(long &rliGalaxies)
s	static	static char scChoice;
rg	array (stands for range)	float rgfTemp[MAX_TEMP];
m_	member variable of a struct or class	char m_cLetterGrade;
p	pointer to a single thing	char *pcGrade;
prg	dynamically allocated array	char *prgcGrades;

Deve Notasyonu | Camel Notation - camelCase

- Camel harf durumu **camelCase** olarak stilize edilmiştir.
- Kelimelerin tek bir büyük harfle ayrıldığını belirten, boşluk veya noktalama olmadan ifadeler yazma pratiğidir.
- Yaygın örnekler arasında "iPhone", "JavaScript" ve "eBay" bulunur.
- Örnek:
 - string adiSoyadi
 - int dersNotu



Pascal Notasyonu | Pascal Notation

- Pascal gösteriminde, tanımlayıcının ilk harfi büyük harfle başlar, ardından ikinci kelimenin başlangıcına kadar küçük harfle devam eder. Eğer varsa, ikinci sözcüğün ilk harfi için büyük harfe başlar ve sonra küçük harfle devam eder.

Örnek:

- MyClass

Pascal Notasyonu | Pascal Notation

- Pascal notasyonu, sınıf, sabit, temsilci, enum türü, sıralama değeri, olay, olay işleyicisi, istisna, statik üye değişkeni, arabirim, yöntem, ad alanı ve özelliği tanımlamak için kullanılır. Tablo Pascal gösterimini kullanan tanımlayıcıları listeler.

Table 1-1. *Identifiers—Pascal Notation*

Identifier	Notation	Example
Class	Pascal	Car
Constant	Pascal	MaximumValue
Delegate	Pascal	ChangeInformation
Enum Type	Pascal	ColorChoice
Enum Value	Pascal	OnlyBlack
Event (Delegate)	Pascal	ChangeDirection
Event Handler	Pascal	OnChangeDirection
Exception	Pascal	OutOfOrderException
Static Member Variable	Pascal	ThisValue
Interface	Pascal	IEngine
Method	Pascal	StopEngine()
Namespace	Pascal	Enterprise.BusinessRules
Property	Pascal	RadiatorCap

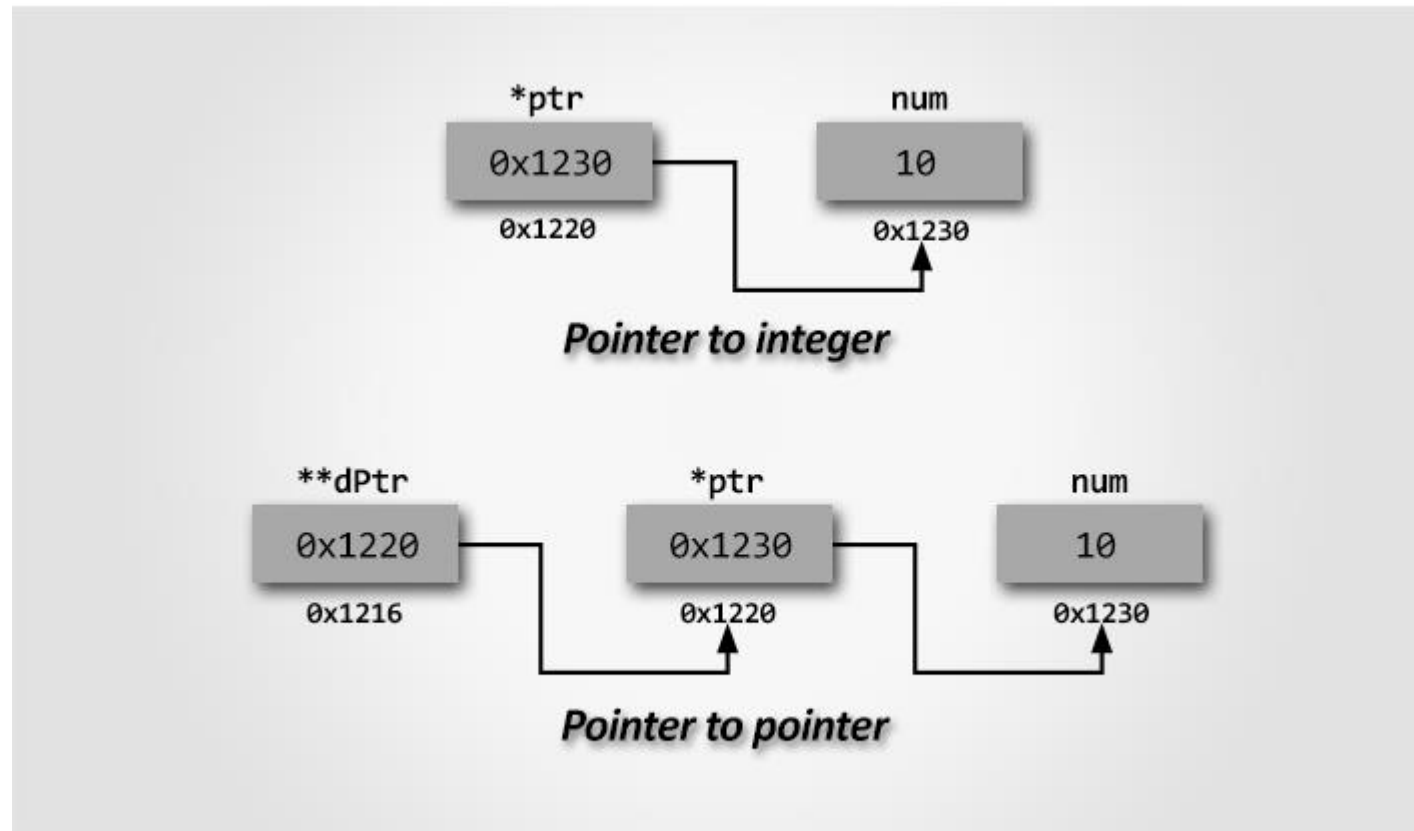
İşaretçiler (pointers)

- C programlama dilinin en karmaşık özelliklerinden biridir.
- C'nin işaretçilerle uğraşırken sağladığı güç ve esneklik, onu diğer birçok programlama dilinden ayırmaya yarar.
- İşaretçiler, karmaşık veri yapılarını etkili bir şekilde temsil etmenize, işlemlere bağımsız değişken olarak aktarılan değerleri değiştirmenize, "dinamik" olarak tahsis edilmiş bellekle çalışmanıza olanak sağlar.
- İşaretçilerin işleyiş şeklini anlamak için öncelikle dolaylı kavramı anlamak gerekir.

İşaretçiler (pointers)

- İşaretçi, **bellek adreslerini depolayan** bir değişkendir.
- Normal değişkenlerin aksine, kullanıcı tarafından verilen veya işlenen değeri saklamaz, bunun yerine **geçerli bilgisayar hafıza adresini saklar**.
- İşaretçiler, dizileri ve yapıları işlemede daha etkilidir.
- İşaretçi, C'de dinamik bellek tahsisine ve serbest bırakmaya (çalışma zamanında değişkenlerin oluşturulması ve silinmesi) izin verir. **Bu şüphesiz işaretçilerin en büyük avantajıdır.**
- Yeni başlayanlar için, iyi uygulanmazsa işaretçiler zorlayıcı olabilir. Ancak, bir kez uzmanlaştıktan sonra, C programlama dilinde yapmak istediğiniz pek çok şeyi yapabilirsiniz.

İşaretçiler (pointers)



Pointer to Pointer (Double pointer) memory representation



Figure 11.1 Pointer to an integer.

Program 11.1 Illustrating Pointers

```
// Program to illustrate pointers

#include <stdio.h>

int main (void)
{
    int    count = 10, x;
    int    *int_pointer;

    int_pointer = &count;
    x = *int_pointer;

    printf ("count = %i, x = %i\n", count, x);

    return 0;
}
```

Program 11.1 Output

```
count = 10, x = 10
```

Program 11.2 More Pointer Basics

```
// Further examples of pointers

#include <stdio.h>

int main (void)
{
    char  c = 'Q';
    char  *char_pointer = &c;

    printf ("%c %c\n", c, *char_pointer);

    c = '/';
    printf ("%c %c\n", c, *char_pointer);

    *char_pointer = '(';
    printf ("%c %c\n", c, *char_pointer);

    return 0;
}
```

Program 11.2 Output

```
Q Q
/ /
( (
```

C Pointer Örnekleri

```
#include<stdio.h>
#include<conio.h>
void dizinin_tersi( int *pointer[]);
int main()
{
    int a[10]={1,2,3,4,5,6,7,8,9,10};
    int i;
    int *ptr[10];

    for(i=0;i<10;i++)
        ptr[i]=&a[i];

    dizinin_tersi (ptr);

    getch();
    return 0;
}

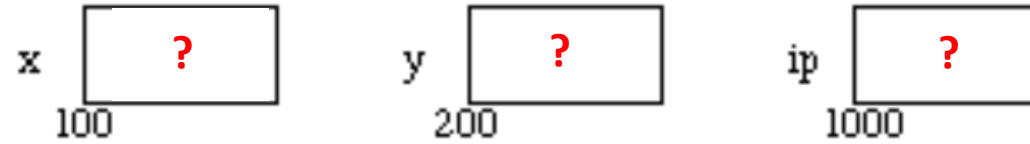
void dizinin_tersi ( int *pointer[])
{
    int i;

    for(i=9;i>=0;i--){
        printf("%d", *pointer[i]);
        printf("\n");
    }
}
```

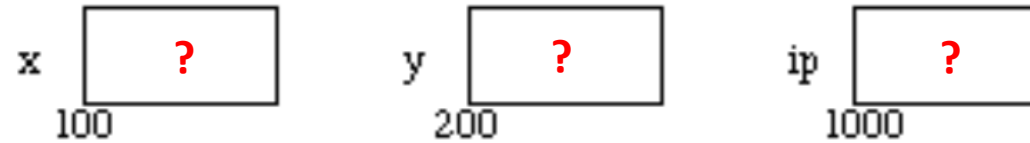

Soru?

```
int x = 1, y = 2;  
int *ip;
```

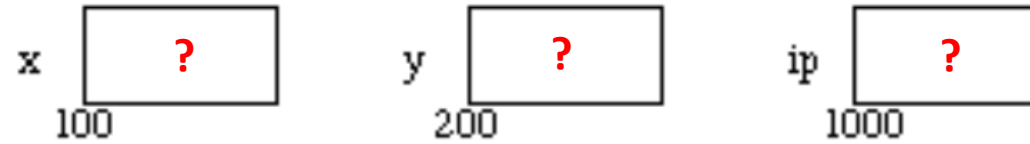
```
ip = &x;
```



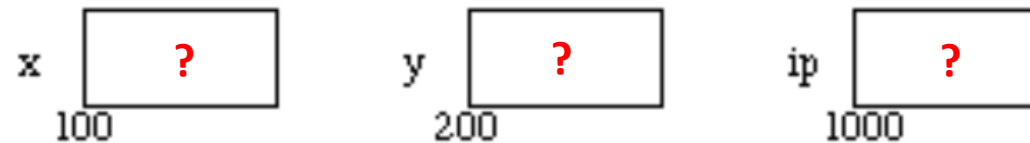
```
y = *ip;
```



```
x = ip;
```



```
*ip = 3
```



```

int main (void)
{
    int x = 1, y = 2, *ip;;
    ip = &x;

    printf("x'in adresi = %p      |   y'nin adresi = %p\n\n", &x, &y);
    printf("x = %d   |   y=%d       |   ip=%p\n", x, y, ip);

    y = *ip;

    printf("*****\n");
    printf("x'in adresi = %p      |   y'nin adresi = %p\n\n", &x, &y);
    printf("x = %d   |   y=%d       |   ip=%p\n", x, y, ip);

    x = ip;

    printf("*****\n");
    printf("x'in adresi = %p      |   y'nin adresi = %p\n\n", &x, &y);
    printf("x = %d   |   y=%d       |   ip=%p\n", x, y, ip);

    *ip = 3;

    printf("*****\n");
    printf("x'in adresi = %p      |   y'nin adresi = %p\n\n", &x, &y);
    printf("x = %d   |   y=%d       |   ip=%p\n", x, y, ip);

    return 0;
}

```