

# 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](mailto:sema@nevsehir.edu.tr) | [s.atasever@gmail.com](mailto: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!

# 7.Hafta Konuları

---

- Pointer kavramı, Yapı kavramı (The concept of structures), Üçlü operatör (Ternary Operator), örnek kodlar ile uygulama.

# Üçlü operatör (Ternary Operator)

---

- Programcılar, daha uzun if ve else koşullu ifadeler yerine karar vermek için üçlü operatörü kullanır.

```
#include <stdio.h>
int main() {

    int a = 10, b = 20, c;

    c = (a < b) ? a : b;

    printf("%d", c);
    |
    return 0;
}
```

**Output: ?**

# Üçlü operatör (Ternary Operator)

- Programcılar, daha uzun if ve else koşullu ifadeler yerine karar vermek için üçlü operatörü kullanır.

```
#include<stdio.h>
main()
{
    int first,second,largest;
    printf("Please provide 2 numbers=>\n");
    scanf("%d %d", &first , &second);

    largest = (first > second) ? first : second;
    printf("%d", largest);
    printf(" is the largest number from %d and %d",first,second);
}
```

Output: ?

# Pointer kavramı

## Örnek

- *int \*p;*
- *int a=10;*
- *p=&a;*

Adres değeri	içeriği	matıksal ismi
A101		
A102		
A103		
A104		
A105		
A106		
A107	A116	p
A108		
A109		
A110		
A111		
A112		
A113		
A114		
A115		
A116	10	a
A117		
A118		
A119		
A120		

# Pointer kavramı

---

```
#include<stdio.h>
main()
{
    int *p;
    int a=10;
    p=&a;

    printf("*p --> %d \n", *p);
    printf("&p --> %d \n", &p);
}
```

**Output: ?**



```
#include <stdio.h>

int main()
{
    int *ptr;
    int x;

    ptr = &x;
    *ptr = 0;

    printf(" x = %d\n", x);
    printf(" *ptr = %d\n", *ptr);

    *ptr += 5;
    printf(" x = %d\n", x);
    printf(" *ptr = %d\n", *ptr);

    (*ptr)++;
    printf(" x = %d\n", x);
    printf(" *ptr = %d\n", *ptr);

    return 0;
}
```

**Output: ?**

# (Yapılarla Çalışma) Working with Structures

---

```
#include <stdio.h>
#include <stdlib.h>

/*

    struct etiketadi {
        degisken tipi degisken1;
        degisken tipi degisken1;
        degisken tipi degisken1;
        ...
        ...
        ...
    };

*/

int main() // Deneme
{
    return 0;
}
```

# (Yapılarla Çalışma) Working with Structures

---

- Yapı (**Structure**): heterojen unsurları bir arada gruplandırmak için bir araçtır.
- Dizi (**Array**): homojen öğeleri bir arada gruplandırmak için bir araçtır.
- Örnek: takvim tarihlerini kaydetme (gün, ay, yıl)
- Sürüm 1: bağımsız değişkenler kullanma:
  - `int ay = 9, gün = 25, yıl = 2004;`
  - Bu yöntemi kullanarak, programda kullandığınız her tarih için mantıksal olarak ilişkili değişkenler olan üç ayrı değişkeni izlemelisiniz. Bu üç değişkenli kümeleri bir şekilde gruplandırmanız çok daha iyi olur. C'deki yapının yapmanıza izin verdiği şey budur!

# Example: structures

---

```
struct date
{
    int month;
    int day;
    int year;
};
```

int türünde 3 alanlı struct date türünü tanımlar. Alanların adları yapı bağlamında yereldir. Bir struct bildirimi bir türü tanımlar: ardından bir değişkenler listesi gelmezse, depolama alanı ayırmaz; sadece bir yapının şablonunu veya şeklini tanımlar.

`struct date today, purchaseDate;` → Struct date türünde 3 değişken tanımlar

```
today.year = 2004;
today.month = 10;
today.day = 5;
```

Struct date türünde bir değişkenin alanlarına erişir. Belirli bir yapının bir üyesine, bir ifadede `structurename.member` biçiminde bir yapı ile başvurulur.

# (Yapılarla Çalışma) Working with Structures

---

```
#include <stdio.h>
#include <stdlib.h>

struct ogrenci {
    char ad[15];
    char soyad[20];
    int num;
    int yas;
};

int main() // Deneme
{
    struct ogrenci rabia = {"Rabia", "Keser", 21, 18} ;
    printf("%s %s %d %d\n", rabia.ad, rabia.soyad, rabia.num, rabia.yas);

    return 0;
}
```

## Program 9.1 Illustrating a Structure

---

```
// Program to illustrate a structure

#include <stdio.h>

int main (void)
{
    struct date
    {
        int  month;
        int  day;
        int  year;
    };

    struct date  today;

    today.month = 9;
    today.day = 25;
    today.year = 2004;

    printf ("Today's date is %i/%i/%.2i.\n", today.month, today.day,
           today.year % 100);

    return 0;
}
```

---

## Store Information and Display it Using Structure

```
#include <stdio.h>
struct student {
    char name[50];
    int roll;
    float marks;
} s;

int main() {
    printf("Enter information:\n");
    printf("Enter name: ");
    fgets(s.name, sizeof(s.name), stdin);

    printf("Enter roll number: ");
    scanf("%d", &s.roll);
    printf("Enter marks: ");
    scanf("%f", &s.marks);

    printf("Displaying Information:\n");
    printf("Name: ");
    printf("%s", s.name);
    printf("Roll number: %d\n", s.roll);
    printf("Marks: %.1f\n", s.marks);

    return 0;
}
```

### Output

```
Enter information:
Enter name: Jack
Enter roll number: 23
Enter marks: 34.5
Displaying Information:
Name: Jack
Roll number: 23
Marks: 34.5
```

# Dizi içeren yapılar (Structures containing arrays)

---

```
struct month
{
    int numberOfDays;
    char name[3];
};
```

```
struct month aMonth;
```

```
. . .
```

```
aMonth.numberOfDays = 31;
aMonth.name[0] = 'J';
aMonth.name[1] = 'a';
aMonth.name[2] = 'n';
```