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!

12. Hafta Konuları

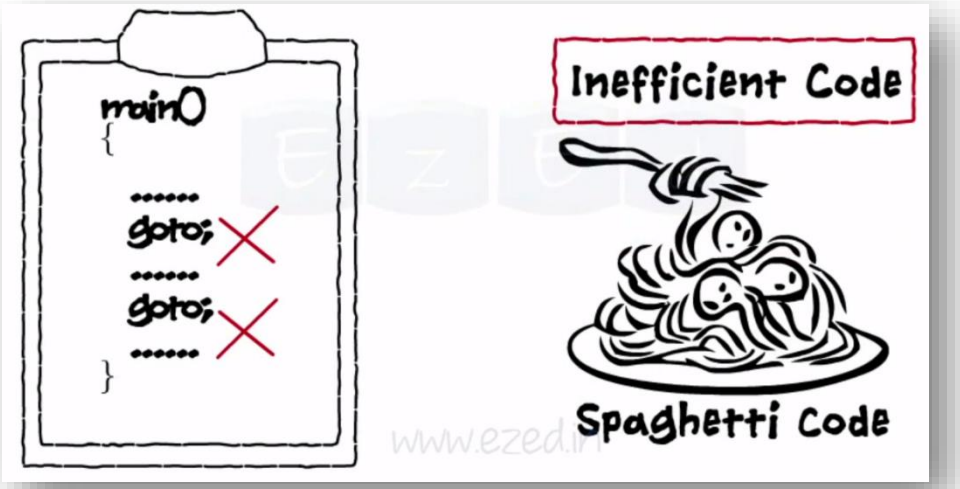
- Yapısal programlama, C dilinin özellikleri, Goto ve null ifadesi,Union veri türü, örnek kod uygulamaları.

Goto İfadesi (The goto Statement)

- Yapısal programlamayı (Structured Programming) öğrenmiş olan herkes için, goto ifadesinin kötü bir şöhreti vardır. Neredeyse her bilgisayar dilinde böyle bir ifade yer alır. Bir goto ifadesinin çalıştırılması, programda belirli bir noktaya doğrudan dallanma yapılmasına neden olur. Bu dallanma, goto'nun yürütülmesi üzerine derhal ve koşulsuz yapılır. Dallanmanın programda nerede yapılacağını belirlemek için bir etiket gereklidir. Bir etiket, değişken isimleriyle aynı kurallarla oluşturulan ve hemen ardından iki nokta üst üste konulması gereken bir isimdir.

Örnek:

```
out_of_data: printf ("Unexpected end of data.\n");
...
...
goto out_of_data;
```



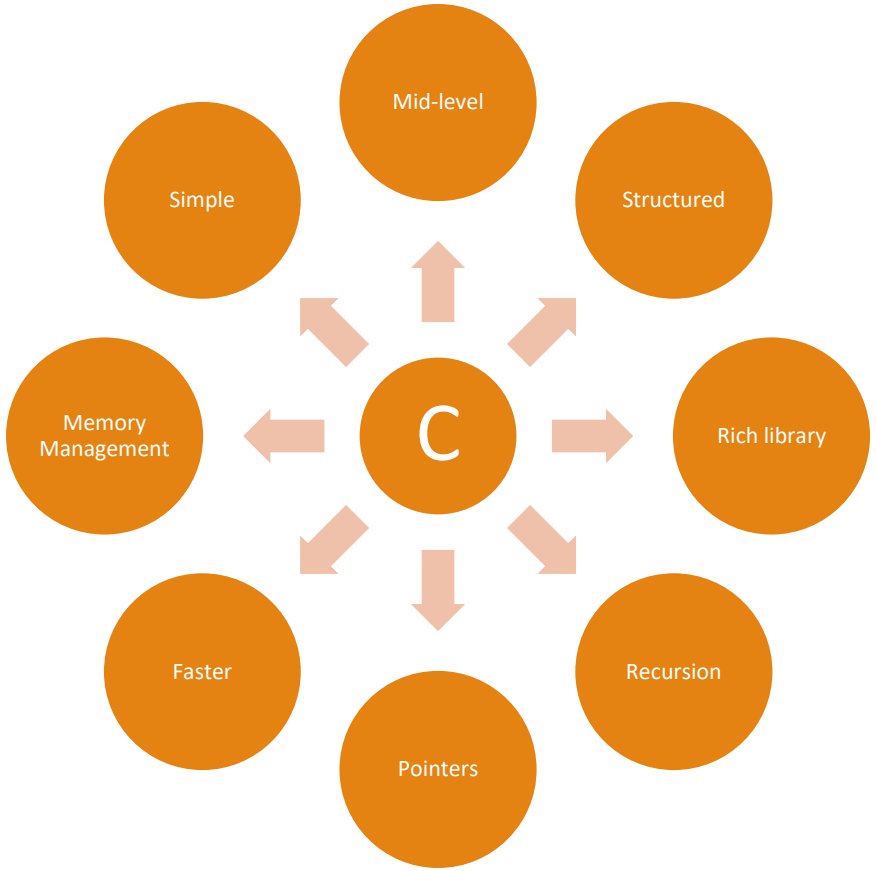
Şekil : https://www.youtube.com/watch?v=TmtyFcLWXwo

Yapısal Programlama (Structured Programming)

```
Structured:
IF x<=y THEN
  BEGIN
     z := y-x;
     q :=SQRT(z);
  END
ELSE
  BEGIN
     z := x-y;
     q := -SQRT(z)
  END;
WRITELN(z,q);
```

```
Unstructured:
IF x>y THEN GOTO 2;
z := y-x;
q := SQRT(z);
GOTO 1;
2: z:= x-y;
q:=-SQRT(z);
1: writeln(z,q);
```

C Dilinin Özellikleri



https://www.javatpoint.com/features-of-c-language

Goto İfadesi (The goto Statement)

```c
// Program to calculate the sum and average of positive numbers
// If the user enters a negative number, the sum and average are displayed.

#include <stdio.h>

int main() {

   const int maxInput = 100;
   int i;
   double number, average, sum = 0.0;

   for (i = 1; i <= maxInput; ++i) {
      printf("%d. Enter a number: ", i);
      scanf("%lf", &number);

      // go to jump if the user enters a negative number
      if (number < 0.0) {
         goto jump;
      }
      sum += number;
   }

jump:
   average = sum / (i - 1);
   printf("Sum = %.2f\n", sum);
   printf("Average = %.2f", average);

   return 0;
}
```

Output

```
1. Enter a number: 3
2. Enter a number: 4.3
3. Enter a number: 9.3
4. Enter a number: -2.9
Sum = 16.60
Average = 5.53
```

https://www.programiz.com/c-programming/c-goto-statement

null İfadesi (The null or empty statement)

- Null veya boş ifade, yalnızca noktalı virgül içeren bir ifadedir.
- ; → null statement
- Örneğin, aşağıdaki ifadenin amacı, standart girdiden okunan tüm karakterleri, bir satır sonu karakteriyle karşılaşılıncaya kadar metnin işaret ettiği karakter dizisine depolamaktır.

```
while ( (*text++ = getchar ()) != '\n' )
    ;
```

null İfadesi (The null or empty statement)

```c
#include <stdio.h>

int main() {
    int line[10];

    for ( int i = 0; i < 10; line[i++] = 0 )
        ; // null statement

    for ( int i = 0; i < 10; i++ )
        printf("line[%d]=%d\n", i, line[i]);

    return 0;
}
```

Output

```
line[0]=0
line[1]=0
line[2]=0
line[3]=0
line[4]=0
line[5]=0
line[6]=0
line[7]=0
line[8]=0
line[9]=0
```

Veri Türleri (struct)

- Aynı türdeki öğeleri tek bir mantıksal varlıkta gruplamanıza izin verir.

```c
// Program to add two distances (feet-inch)
#include <stdio.h>
struct Distance
{
    int feet;
    float inch;
} dist1, dist2, sum;

int main()
{
    printf("1st distance\n");
    printf("Enter feet: ");
    scanf("%d", &dist1.feet);

    printf("Enter inch: ");
    scanf("%f", &dist1.inch);
    printf("2nd distance\n");

    printf("Enter feet: ");
    scanf("%d", &dist2.feet);

    printf("Enter inch: ");
    scanf("%f", &dist2.inch);

    // adding feet
    sum.feet = dist1.feet + dist2.feet;
    // adding inches
    sum.inch = dist1.inch + dist2.inch;

    // changing to feet if inch is greater than 12
    while (sum.inch >= 12)
    {
        ++sum.feet;
        sum.inch = sum.inch - 12;
    }

    printf("Sum of distances = %d\'-%.1f\"", sum.feet, sum.inch);
    return 0;
}
```

Output

```
1st distance
Enter feet: 12
Enter inch: 7.9
2nd distance
Enter feet: 2
Enter inch: 9.8
Sum of distances = 15'-5.7"
```

https://www.programiz.com/c-programming/c-structures

Veri Türleri (union)

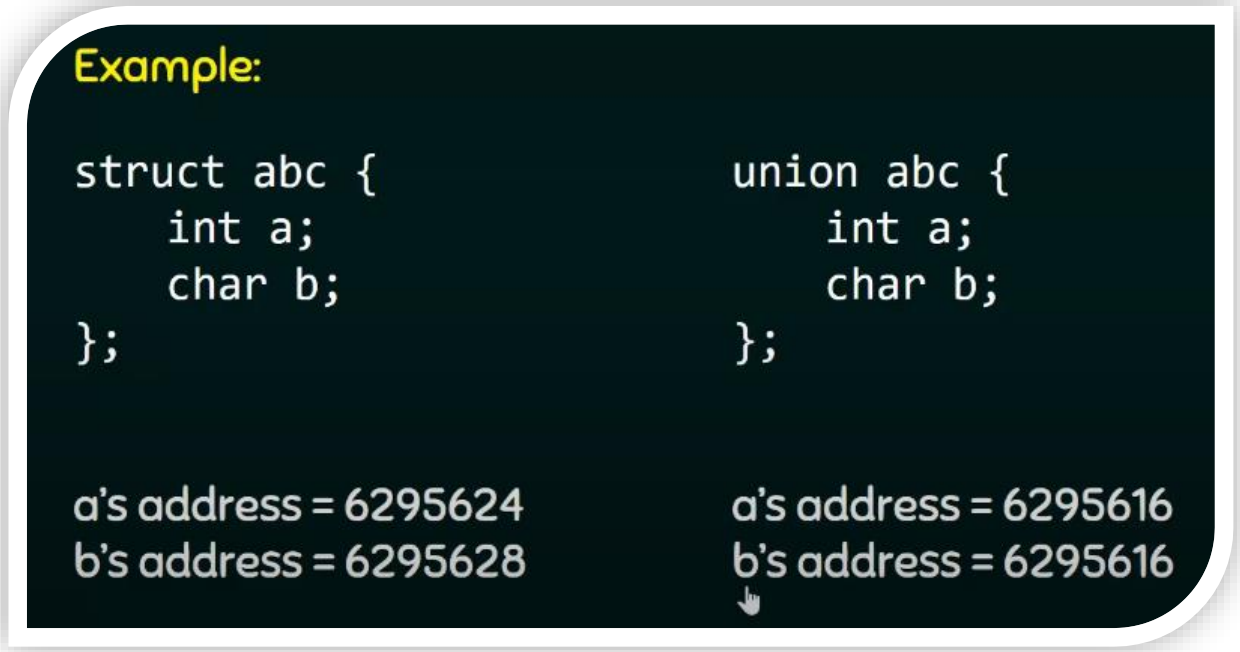
- C programlama dilindeki en alışılmadık yapılardan biridir.
- Bu yapı, temel olarak aynı depolama alanında farklı veri türlerini depolamanın gerekli olduğu daha gelişmiş programlama uygulamalarında kullanılır.
- Örneğin, tek bir karakteri, kayan noktalı bir sayıyı veya bir tamsayıyı saklamak için kullanılabilecek x adında tek bir değişken tanımlamak istiyorsanız, önce, belki de mixed adında bir union tanımlayabilirsiniz:
- union mixed x;

```
union  mixed
{
    char   c;
    float  f;
    int    i;
};
```

Stephen Kochan, Programming in C, 3rd Edition, Sams Publishing, 2005.

Veri Türleri (union)

- union, kullanıcı tanımlı bir veri türüdür, ancak struct'lardan farklı olarak, union üyeleri aynı bellek konumunu paylaşır.



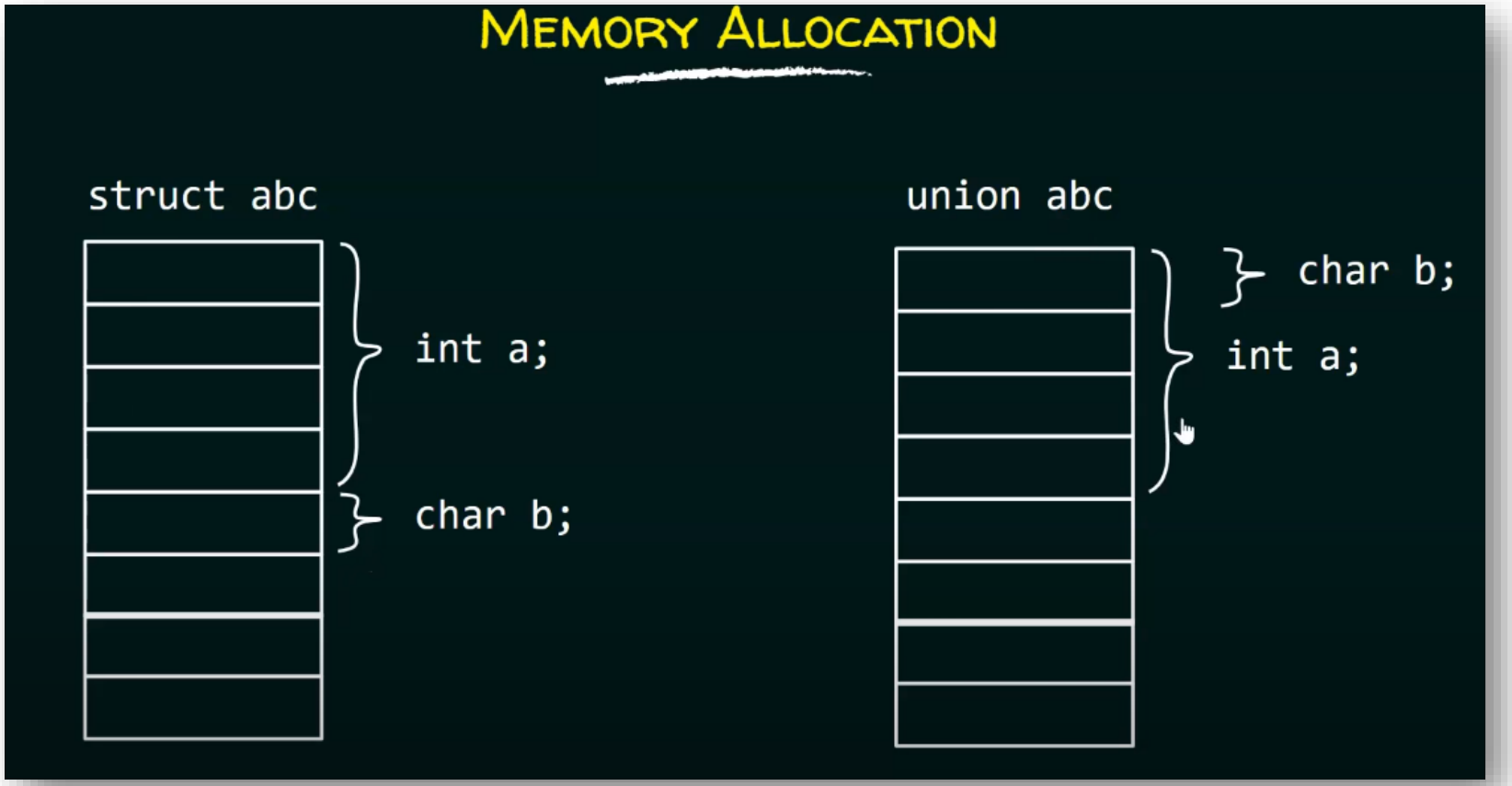
https://www.youtube.com/watch?v=oySsPUDr35U

Veri Türleri (union) - Örnek

```c
#include <stdio.h>

struct sPerson
{
    int age;
    float weight;
};

union uPerson
{
    int age;
    float weight;
};
```

```c
int main() {
    struct sPerson sPerson1;
    union uPerson uPerson1;

    printf("Enter age: ");
    scanf("%d", &sPerson1.age);

    printf("Enter weight: ");
    scanf("%f", &sPerson1.weight);

    printf("\nDisplaying:\n");
    printf("Age: %d | address =%p\n", sPerson1.age, &sPerson1.age);
    printf("weight: %f | address=%p", sPerson1.weight, &sPerson1.weight);

    printf("\n\n****** union *********\n");
    printf("Enter age: ");
    scanf("%i", &uPerson1.age);

    printf("Enter weight: ");
    scanf("%f", &uPerson1.weight);

    printf("\nDisplaying:\n");
    printf("Age: %i | address =%p\n", uPerson1.age, &uPerson1.age);
    printf("weight: %f | address=%p\n", uPerson1.weight, &uPerson1.weight);

    return 0;
}
```

Veri Türleri (union) - Örnek



- Union'ın boyutu union'ın en büyük üyesinin boyutuna göre belirlenir.

https://www.youtube.com/watch?v=oySsPUDr35U

Veri Türleri (union) - Örnek

```c
#include <stdio.h>

union abc
{
   int a;
   char b;
}var;

int main() {
    var.a=65;

    printf("\nDisplaying:\n");
    printf("a: %d | address =%p\n", var.a, &var.a);
    printf("b: %c | address=%p\n", var.b, &var.b);

    return 0;
}
```

Output

```
Displaying:
a: 65 | address =0000000000407970
b: A | address=0000000000407970
```

https://www.youtube.com/watch?v=oySsPUDr35U

Veri Türleri (union) - Örnek

```c
#include <stdio.h>

union abc
{
    int a;
    char b;
    double c;
    float d;
};

int main() {
    int intType;
    float floatType;
    double doubleType;
    char charType;

    printf("In my machine:\n");
    printf("Size of int: %ld bytes\n", sizeof(intType));
    printf("Size of float: %ld bytes\n", sizeof(floatType));
    printf("Size of double: %ld bytes\n", sizeof(doubleType));
    printf("Size of char: %ld byte\n", sizeof(charType));

    printf("\n***** union *******\n");
    printf("Size of union: %ld \n", sizeof(union abc));



    return 0;
}
```

Output

```
In my machine:
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte

***** union *******
Size of union: 8
```

- Union'ın boyutu union'ın en büyük üyesinin boyutuna göre belirlenir.

https://www.youtube.com/watch?v=oySsPUDr35U

Veri Türleri (union) - Örnek

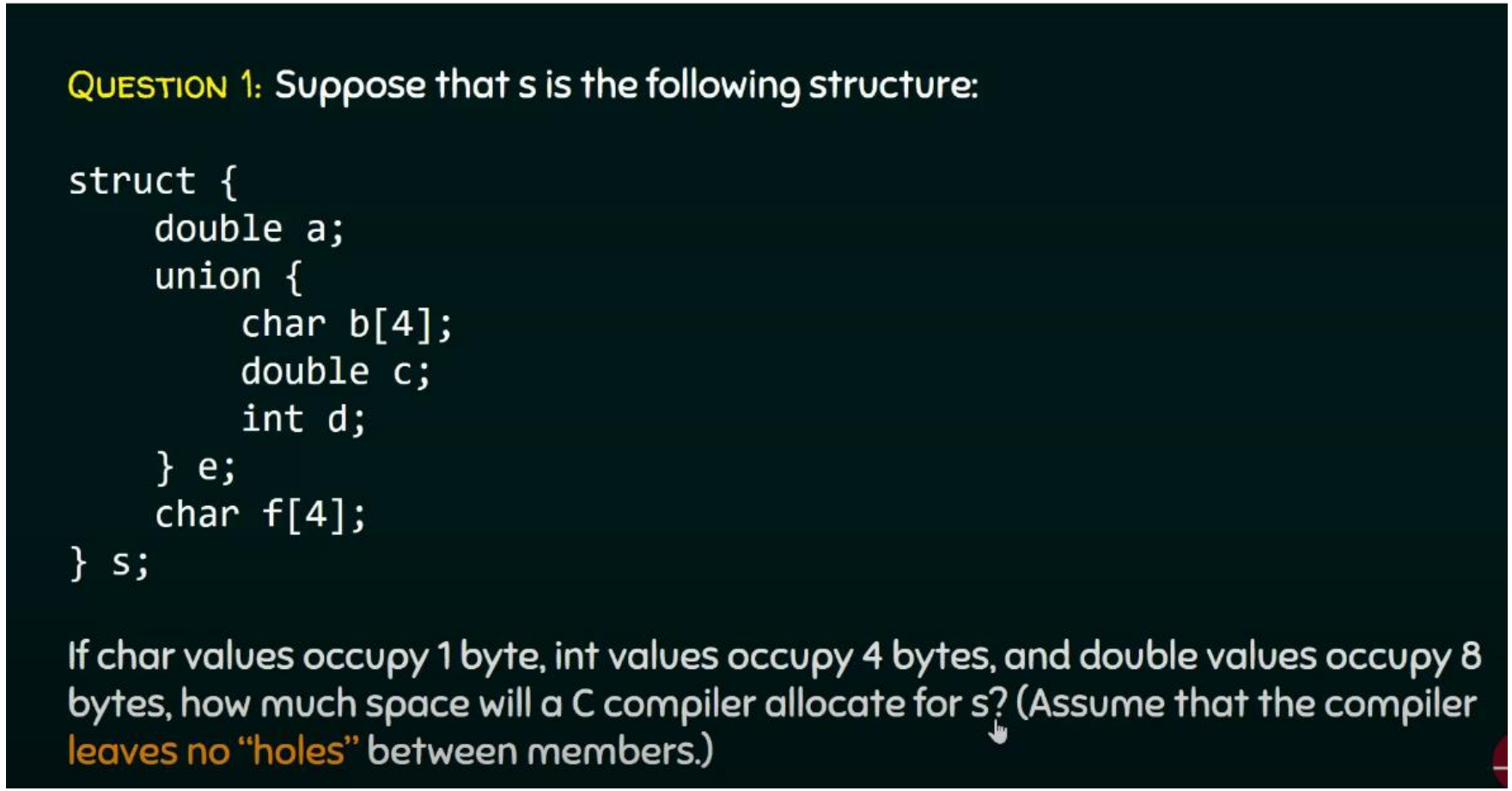
```
union abc {
     int a;
     char b;
};

int main() {
     union abc var;
     var.a = 90;
     union abc *p = &var;
    printf("%d %c", p->a, p->b);
    return 0;
}
```

OUTPUT: 90 Z

- Ok (->) kullanarak işaretçiler (pointers) aracılığıyla union üyelerine erişebiliriz

https://www.youtube.com/watch?v=oySsPUDr35U

- Soru : Aşağıdaki C koduna göre, derleyici **s** veri türü için hafızada ne kadarlık bir yer ayırır? Struct ve union veri türlerinin farklarını düşünerek bu soruyu çözünüz.



https://www.youtube.com/watch?v=gSYFHluoeHs&list=PLBlnK6fEyqRggZZgYpPMUxdY1CYkZtARR&index=168