

2 Type of classes

→ intrinsic { B.C
B.h

struct

Name .C

```
int age;  
str name;
```

```
void Name_ChangeName(str newName){  
    Name = newName;  
}
```

```
void Name_ChangeAge(int newAge){  
    age = newAge;  
}
```

```
str Name_GetName(void){  
    return name;  
}
```

```
int Name_GetAge(void){  
    return age;  
}
```

(I)

Name.h

```
void Name_ChangeName (Str newName);
```

```
void Name_ChangeAge (int newAge);
```

```
Str Name_getName (Void);
```

```
int Name_getAge (Void);
```

This is something like OOP.

✓ we have encapsulated methods, variables (fields)

✓ we can access variables through methods

✗ it is NOT possible to make instances



☆ We can fix this problem by
using structs !



Super Important

Name.c

(I)

will include just methods ✓

Name.h

will include

1. the struct

all of the variables/fields
are inside this struct

2. prototype of methods

Name.c

```
void Name_ChangeName (* Name_Struct, str NewName) {
```

```
    Name_Struct.name = NewName;
```

```
}
```

```
void Name_ChangeAge (* Name_Struct, int NewAge) {
```

```
    Name_Struct.age = NewAge;
```

```
}
```

```
int Name_getAge (* Name_Struct) {
```

```
    return Name_Struct.name;
```

```
}
```

```
str Name_getName (*Name_struct) {  
    return Name_struct.Name;  
}
```

Name.h

```
typedef struct {
```

```
    str name;
```

```
    int age;
```

```
} Name_typedef
```

```
void Name_ChangeName (*Name_struct, str NewName);
```

```
void Name_ChangeAge (*Name_struct, int NewAge);
```

```
int Name_getAge (*Name_struct);
```

```
str Name_getName (*Name_struct);
```

Now to make instances out of this class; you need to `#include "Name.h"`. Now you can make instances of your class and use its methods by passing the address of object. For example:

main.c

```
#include "Name.h"
```

```
Name_typedef Person1;
```

```
Name_typedef Person2;
```

```
Person1.name = "Ali";
```

```
Person1.age = 23;
```

```
Person2.name = "Elone";
```

```
Person2.age = 23;
```

```
Name_getName(*person1);
```

We can put all
of these initializations
inside

```
Name_init(*Name_struct  
          , str name  
          , int age)
```

which is inside

Name.c



this will return "Ali"