Core Softskills Esstinall

22 Module

1. Os ⟶ Exam
2. Linux ⟶ Exam
3. Windows ⟶ Exam
4. Intro To Programming Using c ⟶ Exam
5. OOP Uisng cpp ⟶ Exam
6. Datastructre ⟶ Exam
7. Database using sql server ⟶ Exam
8. Network
9. Cloud
10. Tesing
11. Flow Charts
12. HTML ⟶ Exam
13. CSS ⟶ Exam
14. JS ⟶ Exam
15. Responsive Web
16. Wordpress
17. PHP
18. MySQL
19. Cummincation SKILLS
20. Interviewing Skills
21. CV Writing
22. Free lnace
23. Gen AI

A. Project ⟶ Bussines Model

Day7_Cprogramming

Struct
Dynamic Allocation
Memory Type
Type Casting

Primitive

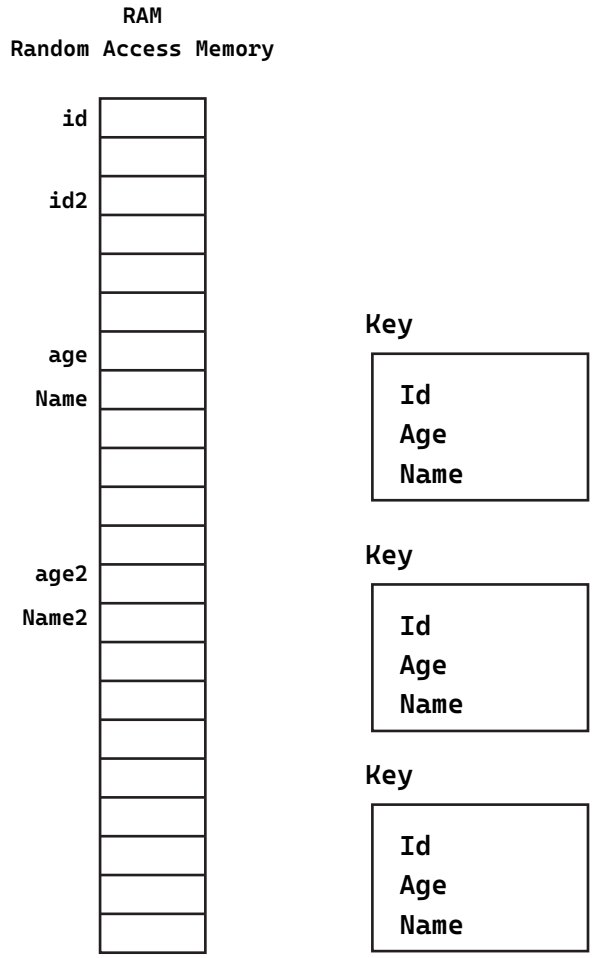
char ⟶ 1Byte
int ⟶ 4Byte
float ⟶ 4Byte
Double ⟶ 8Byte
Pointer ⟶ 8Byte

Non-Primitive

Array
Struct

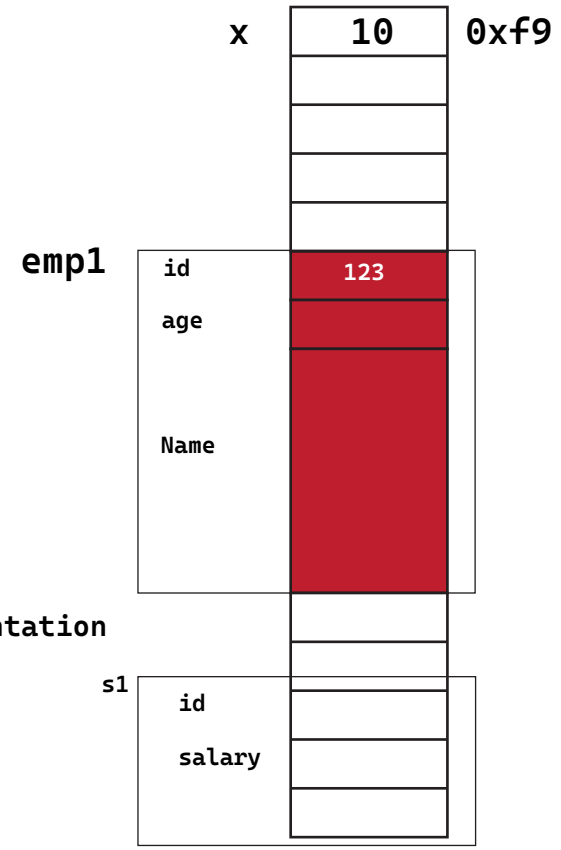
User Define Data

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

int main()
{
    float id;
    int age;
    char Name[20];

    float id2;
    int age2;
    char Name[20]2;

    float id3;
    int age3;
    char Name[20];

//Array

    return 0;
}
```



Struct ⟶ User Define Datatype, Can Carry Multy Variable From Def Datatype

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

struct Employee{ //Logical Representation
            float Id; ⟶ 4
            int Age; ⟶ 4
            char Name[20]; ⟶  20Byte
 }

int main()
{
   int x = 10;
 //Decliration
   struct Employee emp1;//Physcial Reprisintation
   emp1.id = 123;
   emp1.age = 28;
   //emp1.Name=”Mustafa”
   emp1.Name[0]=’M’;
   strcpy(emp1.Name,”Mustafa”);
    return 0;
}
```



Local Variable

x

كل المتغيرات المعرفة داخل main

Member Variable

emp1

كل المتغيرات المعرفة داخل Struct

struct DataTypeName{}

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
//Logical Represintation
//Blueprint
struct Employee{
    float id;
    int age;
    char Name[20];
};

int main()
{
//Decliration
    struct Employee e1;
    //Assignment
    e1.id = 123;
    e1.age = 28;
    strcpy(e1.Name,"Mustafa") ;
    //Call
    printf("Employee Id = %f\n",e1.id);
    printf("Employee Age = %i\n",e1.age);
    printf("Employee Name = %s\n",e1.Name);

    struct Employee e2;
    e2.id=456;
    strcpy(e2.Name,"Ahmed");
    e2.age=30;
    printf("Employee 2 Id = %.0f\n",e2.id);
    printf("Employee 2 Age = %i\n",e2.age);
    printf("Employee 2 Name = %s\n",e2.Name);
    return 0;
}
```

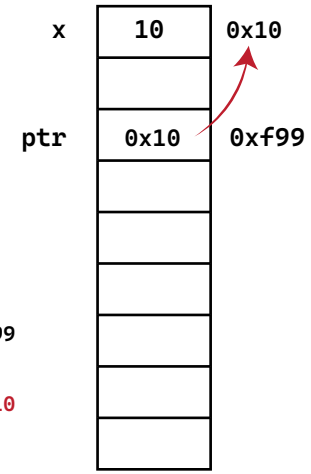
Array Of Struct


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


struct Employee{//28Byte
    int id;
    int age;
    char Name[20];
};




int main()
{
   struct Employee ArrOfEmp[5];//280
            for(int i = 0;i < 5;i++){
              printf(”Please Enter Employee Number %i Id”,i + 1);
              scanf(”%i”, &ArrOfEmp[i].id);
              printf(”Please Enter Employee Number %i Age”,i + 1);
              scanf(”%i”, &ArrOfEmp[i].age);
              printf(”Please Enter Employee Number %i Name”,i + 1);
              scanf(”%s”, &ArrOfEmp[i].Name);
            }

            for(int i=0; i<5; i++){
              printf(”Employee Number %i Id: %i”,i+1, ArrOfEmp[i].id);
              printf(”Employee Number %i Age: %i”,i+1, ArrOfEmp[i].age);
              printf(”Employee Number %i Name: %s”,i+1, ArrOfEmp[i].Name);
            }

    return 0;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
struct Employee{//28Byte
    int id;
    int age;
    char Name[20];
};
int main()
{
   struct Employee ArrOfEmp[5];//280
    for(int i = 0;i < 5;i++){
      printf("Please Enter Employee Number %i Id\n",i + 1);
      scanf("%i", &ArrOfEmp[i].id);
      printf("Please Enter Employee Number %i Age\n",i + 1);
      scanf("%i", &ArrOfEmp[i].age);
      printf("Please Enter Employee Number %i Name\n",i + 1);
      scanf("%s", &ArrOfEmp[i].Name);
    }

    for(int i=0; i<5; i++){
      printf("Employee Number %i Id: %i\n",i+1, ArrOfEmp[i].id);
      printf("Employee Number %i Age: %i\n",i+1, ArrOfEmp[i].age);
      printf("Employee Number %i Name: %s\n",i+1, ArrOfEmp[i].Name);
      printf("\n\n_____\n\n");
    }

    return 0;
}

```

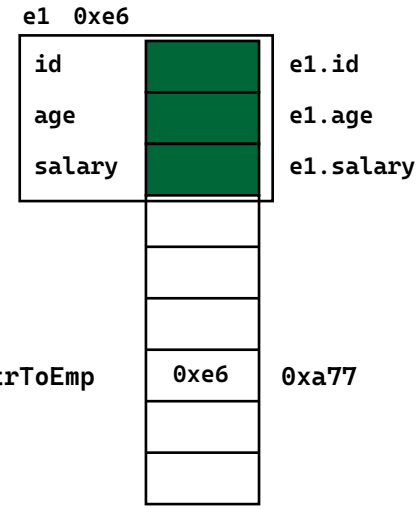
Pointer To Struct

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

int main()
{
    int x = 10;
    int *ptr = &x;

    printf("%p\n",ptr);//0x10
    printf("%p\n",&ptr);//0xf99
    printf("%i\n",*ptr);//10
    printf("%i\n",*(0x10));//10

    return 0;
}
```



Name	value	Address
x	10	0x10
ptr	0x10	0xf99
e1	0xe6	0xe6
ptrToEmp	0xe6	0xa77

```c
#include <stdio.h>
#include <stdlib.h>
  struct Employee{//12Byte
    int id;
    int age
    float salary
  }
int main()
{
 struct Employee e1;
struct Employee *ptrToEmp = &e1;
ptrToEmp//0xe6
(*ptrToEmp).id
ptrToEmp→id;
ptrToEmp→age;
ptrToEmp→salary;
    return 0;
}
```



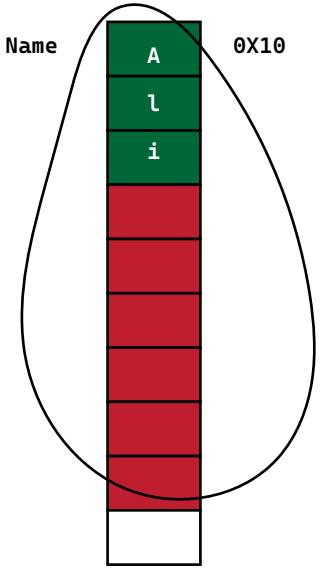

```c
#include <stdio.h>
#include <stdlib.h>
struct Employee{//28Byte
    int id;
    int age;
    float salary;
};
```

```c
int main()
{
   struct Employee e1;
    //   e1.id=10;
    //   e1.age=28;
    //   e1.salary=5000;
    //   printf("Id: %i\n",e1.id);
    //   printf("Age: %i\n",e1.age);
    //   printf("Salary: %f\n",e1.salary);
    struct Employee *ptrToEmp = &e1;

    //(*ptrToEmp).id=123;
    //printf("Id: %i\n",(*ptrToEmp).id);

    ptrToEmp->id = 123;
    ptrToEmp->age=28;
    ptrToEmp->salary=6000;

    printf("Id: %i\n",ptrToEmp->id);
    printf("Age: %i\n",ptrToEmp->age);
    printf("Salary: %f\n",ptrToEmp->salary);

    return 0;
}
```

```c
ptrToEmp→id;
ptrToEmp→age;
ptrToEmp→salary;
```

Dynamic Allocation

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

int main()
{
    char Name[100];
    printf("Please Enter Your Name\n");
    fgets(Name,sizeof(Name),stdin);


    printf("Your Name Is %s",Name);
    return 0;
}
```



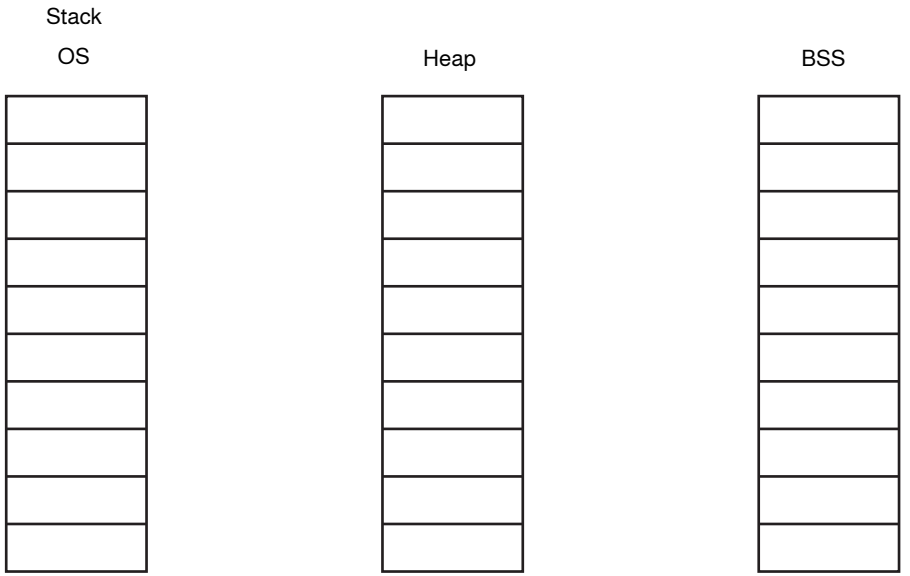
OS → RAM

os Has Full Control To Manage Ram
الـos هو المسؤول عن تحديد المساحة المحجوزة فى الذاكرة

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

int main()
{
    int size;
    printf("Please Enter Name Number Of Character\n");
    scanf("%i",&size);
    _flushall();

    char Name[size];
    printf("Please Enter Your Name\n");
    fgets(Name,sizeof(Name),stdin);


    printf("Your Name Is %s",Name);
    return 0;
}
```

Array is A Compile Time Data Type
معناها ان لازم وقت الـCompile يكون OS عارف هيروح يحجز مساحة كام BYTE للمصفوفة

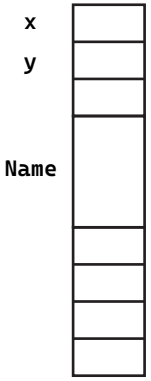
الحل:
استخدام Dynamic Allocation



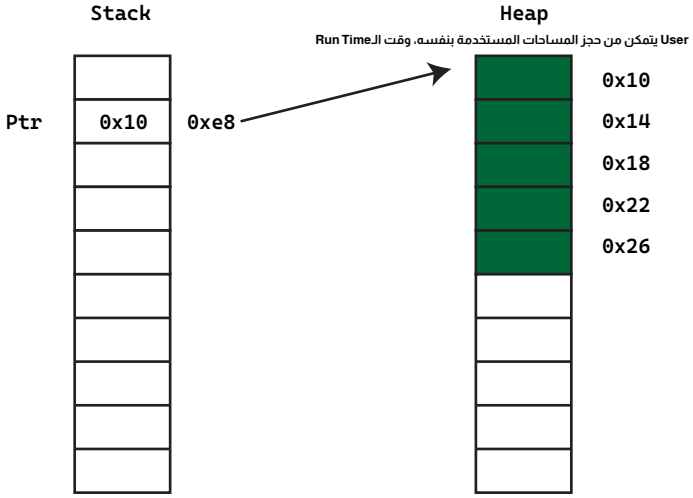
Stack

os هو المسؤول عن حجز المساحات فى Stack وقت الـCompile Time

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

int main()
{
    int x =10;
    int y = 20;
    char Name[30];
    return 0;
}
```

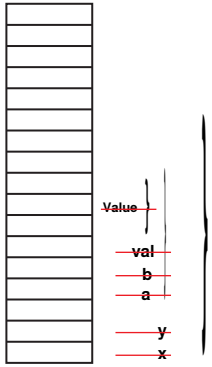


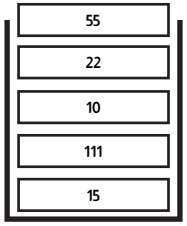

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    int *Ptr;
    Ptr = malloc(20);

    return 0;
}
```



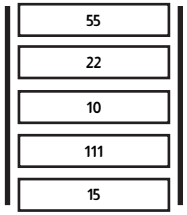
Stack
LIFO
Last in First Out

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int ReturnValue(){
    int value;
}
void Swap(int a, int b){
    int val = ReturnValue();
}
int main()
{
   int x;
   int y;
   swap(x, y);
    return 0;
}
```





Queue
FIFO
First In First Out



Casting

هو عملية تحويل القيم من نوع بيانات لنوع اخر

Implicit

الكومبايلر هو الى بيقوم بعملية الكاستنج

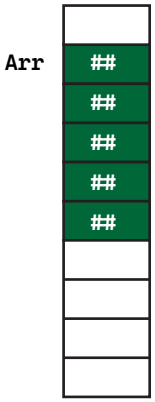
Explicit

Dynamic Allocation

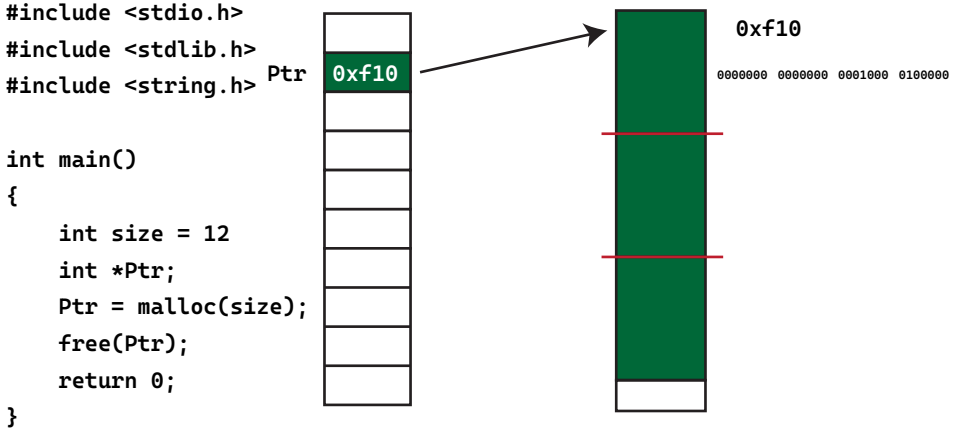
بتسمح للمستخدم وقت الـRun Time بحجز وتعديل المساحة المطلوبة
Life time بتاعها غير متوقف على تحرير المستخدم لهذه المساحة

```
void malloc(20)
```

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

int main()
{
    int Arr[5];
    return 0;
}
```




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

int main()
{
    int size = 12
    int *Ptr;
    Ptr = malloc(size);
    free(Ptr);
    return 0;
}
```



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

int main()
{
    int *Ptr;
//    printf("%i",sizeof(int));
    int size;
    printf("Please Enter Size Of Array\n");
    scanf("%i",&size);
    Ptr = malloc(sizeof(int) * size);

    for(int i=0;i<size;i++){
        printf("Please Enter Value Number %i\n",i+1);
        scanf("%i",&Ptr[i]);

    }

    for(int i=0;i<size;i++){
        printf("%i\n",Ptr[i]);
        printf("%p\n",&Ptr[i]);

    }

    return 0;
}

```

```c
#include <stdio.h>
#include <stdlib.h>
struct Employee{
    int id;//4
    char Name[20];//20
    float Salary;//28
};
int main()
{
    int size;
    printf("Please Enter Number Of Employee\n");
    scanf("%i",&size);

    struct Employee *Ptr;
    Ptr = malloc(sizeof(struct Employee) * 1);

    for(int i=0;i<size;i++){
        printf("Please Enter Employee Number %i, Id\n",i+1);
        scanf("%i",&Ptr[i].id);
        printf("Please Enter Employee Number %i, Salary\n",i+1);
        scanf("%f",&Ptr[i].Salary);
        printf("Please Enter Employee Number %i, Name\n",i+1);
        scanf("%s",&Ptr[i].Name);
    }
//Print All Data
    for(int i=0;i<size;i++){
       printf("Employee Number %i, Id: %i\n",i+1,Ptr[i].id);
       printf("Employee Number %i, Name: %s\n",i+1,Ptr[i].Name);
       printf("Employee Number %i, Salary: %.0f\n",i+1,Ptr[i].Salary);
    }

```

```c

//Search
printf("\n\n\n_____________\n\n\n");
    int Id;
    printf("Please Enter Id Value To Seach\n");
    scanf("%i",&Id);
    for(int i=0;i<size;i++){
        if(Ptr[i].id ==Id){
        printf("Employee Number %i, Id: %i\n",i+1,Ptr[i].id);
        printf("Employee Number %i, Name: %s\n",i+1,Ptr[i].Name);
        printf("Employee Number %i, Salary: %.0f\n",i+1,Ptr[i].Salary);
        break;
        }
    }

//Edit
printf("\n\n\n_____________\n\n\n");
    Id;
    printf("Please Enter Id Value To Edit\n");
    scanf("%i",&Id);
    for(int i=0;i<size;i++){
        if(Ptr[i].id ==Id){
        printf("Please Enter Employee Number %i, Id\n",i+1);
        scanf("%i",&Ptr[i].id);
        printf("Please Enter Employee Number %i, Salary\n",i+1);
        scanf("%f",&Ptr[i].Salary);
        printf("Please Enter Employee Number %i, Name\n",i+1);
        scanf("%s",&Ptr[i].Name);
        break;
        }
    }

printf("\n\n\n_____________\n\n\n");
    for(int i=0;i<size;i++){
       printf("Employee Number %i, Id: %i\n",i+1,Ptr[i].id);
       printf("Employee Number %i, Nam e: %s\n",i+1,Ptr[i].Name);
       printf("Employee Number %i, Salary: %.0f\n",i+1,Ptr[i].Salary);
    }


free(Ptr);



// Try All What We Learn In Lecture
    return 0;
}

```