Day3_Cprogramming

Day3

Type Hireriecy

Constrol Statement

- Loop
  - Open Loop
  - Close Loop

Arrays

String

1

في حالة استخدام أى Operation ينتج عنها Expresion وهو قيمة يجب ان يكون لها نوع بيانات

Size

Point

Type Hieriricy

Double → Float → Int → Char

char --> 1byte --> 255

int -- > 4byte --> 2b

float --> 4byte 2b.7p

double --> 8Byte 48

Float + int = Float

int + int = int

int + char = int

int + double = double

double + char = double

double + float = double

2

- Loop
  - Close Loop

في حالة وجود معلومية عدد مرات تكرار اللوجيك

For Loop

```
for( Start Point;Condtion; Increment - Decement )
{
        //Logic
}
```

```
float Balance= 1000;
int Years = 5;
float  Interest = 0.1;
float  Amount;

Amount = Balance * Interest;//100
Balance += Amount;//1100
printf("Amount In First Year = %f, Balanvce = %f", Amount , Balance)

Amount = Balance * Interest;//110
Balance += Amount;//1210
printf("Amount In Secound Year = %f, Balanvce = %f", Amount , Balance)

Amount = Balance * Interest;
Balance += Amount;
printf("Amount In Third Year = %f, Balanvce = %f", Amount , Balance)

Amount = Balance * Interest;
Balance += Amount;
printf("Amount In Forth Year = %f, Balanvce = %f", Amount , Balance)

Amount = Balance * Interest;
Balance += Amount;
printf("Amount In Fifth Year = %f, Balanvce = %f", Amount , Balance)
```

3

1. Years = n
2. اللوجيك داخل Body الـ For هيتنفذ n من المرات
3. increment or decrement هيتنفذ n من المرات
4. فحص الشرط هيتنفذ n+1 من المرات
5. Initlization - Start Point هتتنفذ مرة واحدة فقط

1

```
for(int i = 0;i<Years; i++)
{
    Amount = Balance * Interst;
    Balance += Amount;
    printf("Amount Of Ye %i = %.1f, Balance = %.1f\n",i+1 ,Amount, Balance);
}
```

Swap Variable That Equal 123 to 321;

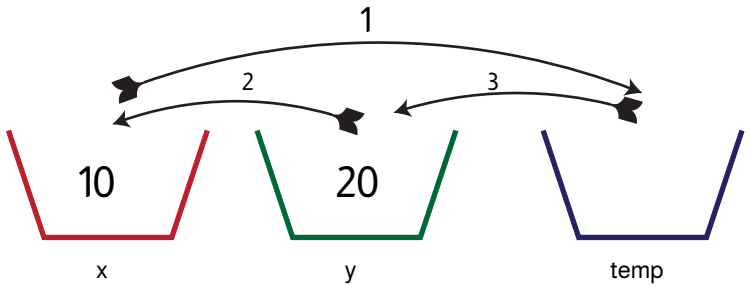
```
int x = 123;
int y=0;

y = y + x  % 10;
x = x / 10
y = y * 10;

y = y + x  % 10;
x = x / 10
y = y * 10;

y = y + x  % 10;
x = x / 10//12
y = y * 10;
```

V1 - Very Bad

```c
int x = 123;
int y=0;

y = y + x  % 10;
x = x / 10
y = y * 10;

y = y + x  % 10;
x = x / 10
y = y * 10;

y = y + x  % 10;
x = x / 10//12
y = y * 10;
```

V2 - Good

```c
int y=0;
for(int i=0;i<9;i++){
    y = y * 10;
    y = y + x  % 10;
    x = x / 10;
    printf("x = %i\n",x);
    printf("y = %i\n",y);
    printf("\n\n_________\n\n");
}
```


```c
int x = 10;
int y = 20;

printf("x = %i\n",x);
printf("y = %i\n",y);
//V1
temp = x;
x = y;
y = temp;
//V2
x = x + y;//30;
y = x - y;//10
x = x - y;//20
//V3
x = x * y;//30;
y = x / y;//10
x = x / y;//20
printf("x = %i\n",x);
printf("y = %i\n",y);
```

Nested If

```c

int age =16;
char gender ='m';

if(age>=16){

    if(gender =='m'){
        //Allow Male
    }else{
        //Allow Female
    }
}
```

Nested For

```c
for(int i = 0; i<5; i++){

    for(int j = 0; j<5; j++){
        //logic
    }
}
```

Loop Num1	Loop Num2	Loop Num3	Loop Num3	Loop Num3
i = 0	i = 1	i = 2	i = 3	i = 4
j = 0	j = 0	j = 0	j = 0	j = 0
j = 1	j = 1	j = 1	j = 1	j = 1
j = 2	j = 2	j = 2	j = 2	j = 2
j = 3	j = 3	j = 3	j = 3	j = 3
j = 4	j = 4	j = 4	j = 4	j = 4
j = 5	j = 5	j = 5	j = 5	j = 5

```c
for(int i = 0; i<5; i++){

    for(int j = 0; j<5; j++){
        printf("*");
    }
    printf("\n");

}
```

```
    j j j j j
i  * * * * *
i  * * * * *
i  * * * * *
i  * * * * *
i  * * * * *
```

7

```c
for(int i = 0; i<5; i++){

    for(int j = 0; j<i+1; j++){
        printf("*");
    }
    printf("\n");
}

```

```
      j
i     *
i    **
i   ***
i  ****
i *****
```

```c
for(int i = 0; i<5; i++){

    for(int j = 5; j>i; j--){
        printf("*");
    }
    printf("\n");
}
```

```
*****
****
***
**
*
```

- Loop
    - Open Loop

فى حالة عدم وجود معلومية عدد مرات تكرار اللوجيك

While

While

do While

While(Condition){
//Logic

}

القاعدة بتقول ان طول ما الشرط داخل اللوب = true هيدخل ينفذ الـBody
2- دائما فى الشرط اكتب عكس البيزنس المطلوب
3- اولا يفحص الشرط وثانيا ينفذ الكود بداخل الـBody


```c
//Bussiness Logic  age > 18
printf("Please Enter Your Age\n");
scanf("%i",&Age);

while(Age<=18){
    printf("Please Enter Age > 18\n");
    printf("Please Enter Your Age\n");
    scanf("%i",&Age);
}
printf("Your Age Is %i\n",Age);
```

do While

do{

}while();

القاعدة بتقول ان طول ما الشرط داخل اللوب = true هيدخل ينفذ الـBody
2- دائما فى الشرط اكتب عكس البيزنس المطلوب
3- اولآ ينفذ الكود داخل الـBody وثانيا يفحص الشرط
4- يبقى فى حالة استخدام Do While الكود هيتنفذ مرة واحدة على الاقل حتى اذا كان الشرط = False

```c
int Age;
//Bussiness Logic  age > 18
do{
    printf("Please Enter Your Age\n");
    scanf("%i",&Age);
}while(Age<=18);
```

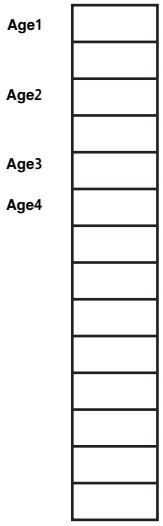
Arrays

فى حالة انك عايز تخزن مجموعة من القيم والقيم ده من نفس نوع البيانات
يبقى احسن نوع بيانات لتخزينها هو المصفوفات Arrays

Very Bad

```
1 int age1 = 25;
2 int age2= 27;
3 int age3= 27;
4 int age4= 27;
5 int age5= 27;
6 int age6= 27;
7
```

RAM --> Random Access Memory
stack



Array?

1. هى عبارة عن Fixed Numbers Of Variable
2. يعني فى حالة حددت Size الخاص بالمصفوفة فهو لا يمكن تغييره
3. جميع القيم داخل المصفوفة هما من نفس نوع البيانات
4. بيتم حفظ البيانات بداخل المصفوفة بشكل Sequntial
5. Compile Time Data Type يعني لازم الـos يكون عارف وقت ما اعمل Build هيجز مساحة كام
6. Dervid Data Type يعني اقدر اتحكم تحجز مساحة اجمالية كام فى الذاكرة وقت التطوير
7. ان اسم المصفوفة نفسها هو بيمثل عنوان اول عنصر فى المصفوفة.

int Age[10] --> 40Byte

```c
//Decliration
int Age[5];
```

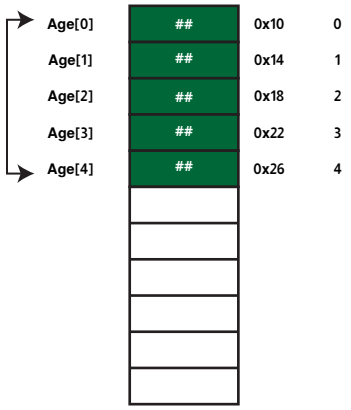
Age = 0x10

Age[0] = ##

&Age[0] = 0x10

RAM --> Random Access Memory

stack

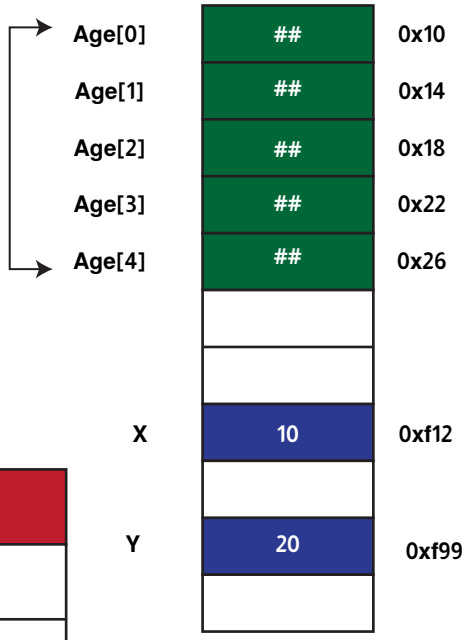



```c
//Array
    int Age[5];

//Assignment
    Age[0]=10;
    Age[1]=12;
    Age[2]=144;
    Age[3]=15;
    Age[4]=123;


//Access
    printf("%i\n",Age[0]);
    printf("%i\n",Age[1]);
    printf("%i\n",Age[2]);
    printf("%i\n",Age[3]);
    printf("%i\n",Age[4]);
```

```c
//Decliration
    int x = 10;
    int y = 20;
//Array
    int Age[5];
```

RAM --> Random Access Memory

stack



Variable	Value	Address
x	10	0xf12
y	20	0xf99
Age	0x10	0x10
Age[0]	##	0x10

11

For Loop →

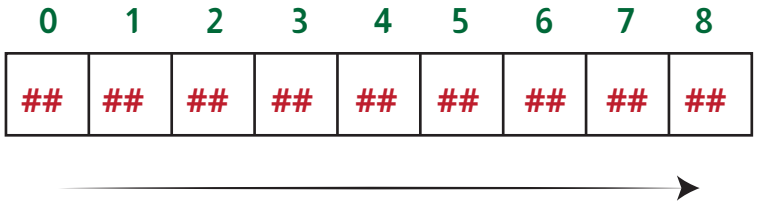
12

```c
for(int i =0;i<5;i++){
    printf("%i\n",Age[i]);
}
```

```
int Arr[5]={123, 6, 9, 4, 10};
int Arr[];//Fale
int Arr[] ={123, 6, 9, 4, 10};//True
int Arr[5]={123};//123,0,0,0,0;
int Arr[5];//### True
int Arr[5]={};//0,0,0,0,0 //True
int Arr = {1,2,3,4};//Flase
```

```
1  int Arr[5]={123, 6, 9, 4, 10};
2  int Arr[];//Fale
3  int Arr[] ={123, 6, 9, 4, 10};//True
4  int Arr[5]={123};//123,0,0,0,0;
5  int Arr[5];//### True
6  int Arr[5]={};//0,0,0,0,0 //True
7  int Arr = {1,2,3,4};//Flase
```

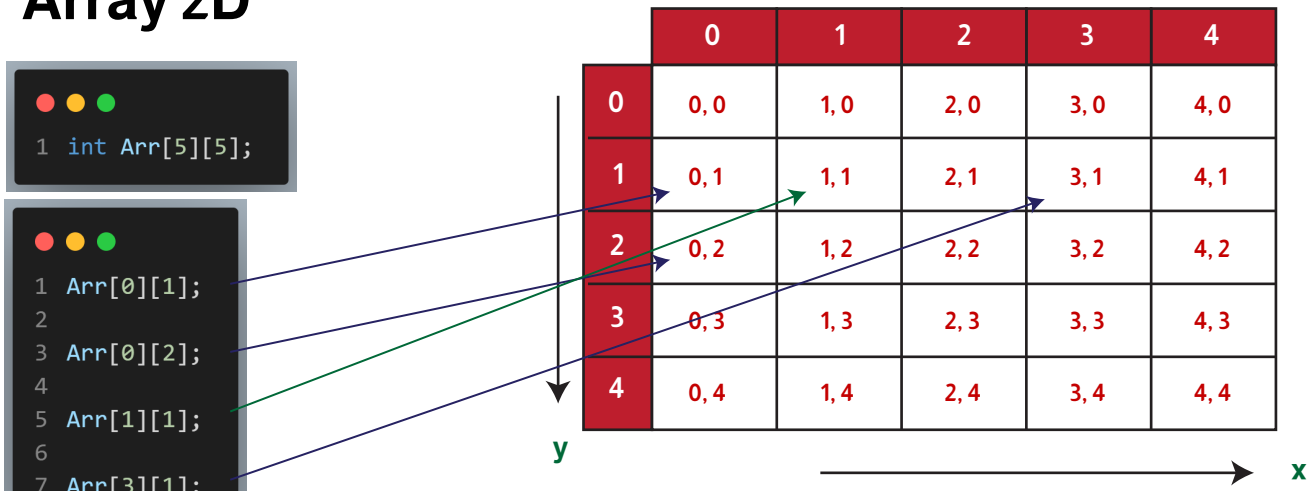
Array 1D



Array 2D

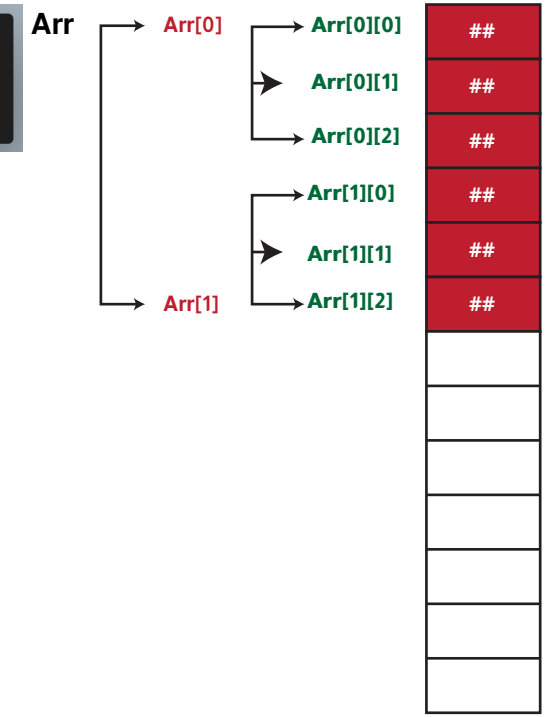
```
1  int Arr[5][5];
```

```
1  Arr[0][1];
2
3  Arr[0][2];
4
5  Arr[1][1];
6
7  Arr[3][1];
```



RAM --> Random Access Memory
stack

```
1  int Arr[2][3]
```



String--> Next Session

A	L	I
M	A	Y