

## مدخل إلى الخوارزميات والبرمجة هندسة الميكاترونكس سنة أولى

2023-2024

مدرس المقرر: د. عيسى الغنام

### Lecture No.9

#### Part1 C++ References

##### Creating References

A reference variable is a "reference" to an existing variable, and it is created with the & operator:

```
string food = "Pizza"; // food variable
string &meal = food; // reference to food
```

Now, we can use either the variable name food or the reference name meal to refer to the food variable:

Example

```
string food = "Pizza";
string &meal = food;

cout << food << "\n"; // Outputs Pizza
cout << meal << "\n"; // Outputs Pizza
```

##### C++ Memory Address

In the example from the previous page, the & operator was used to create a reference variable. But it can also be used to get the memory address of a variable; which is the location of where the variable is stored on the computer.

When a variable is created in C++, a memory address is assigned to the variable. And when we assign a value to the variable, it is stored in this memory address.

To access it, use the & operator, and the result will represent where the variable is stored:

Example

```
string food = "Pizza";
```

```
cout << &food; // Outputs 0x6dfed4
```

**Note:** The memory address is in hexadecimal form (0x..). Note that you may not get the same result in your program.

## C++ Pointers-Creating Pointers

You learned from the previous chapter, that we can get the **memory address** of a variable by using the & operator:

Example

```
string food = "Pizza"; // A food variable of type string

cout << food; // Outputs the value of food (Pizza)
cout << &food; // Outputs the memory address of food (0x6dfed4)
```

A **pointer** however, is a variable that **stores the memory address as its value**.

A pointer variable points to a data type (like int or string) of the same type, and is created with the \* operator. The address of the variable you're working with is assigned to the pointer:

Example

```
string food = "Pizza"; // A food variable of type string
string* ptr = &food; // A pointer variable, with the name ptr,
that stores the address of food

// Output the value of food (Pizza)
cout << food << "\n";

// Output the memory address of food (0x6dfed4)
cout << &food << "\n";

// Output the memory address of food with the pointer (0x6dfed4)
cout << ptr << "\n";
```

Example explained

Create a pointer variable with the name ptr, that **points to** a string variable, by using the asterisk sign \* (string\* ptr). **Note that the type of the pointer has to match the type of the variable you're working with.**

Use the & operator to store the memory address of the variable called food, and assign it to the pointer.

Now, ptr holds the value of food's memory address.

**Tip:** There are three ways to declare pointer variables, but the first way is preferred:

```
string* mystring; // Preferred
string *mystring;
string * mystring;
```

## C++ Dereference- Get Memory Address and Value

In the example from the previous page, we used the pointer variable to get the memory address of a variable (used together with the & **reference** operator). However, you can also use the pointer to get the value of the variable, by using the \* operator (the **dereference** operator):

Example

```
string food = "Pizza"; // Variable declaration
string* ptr = &food; // Pointer declaration

// Reference: Output the memory address of food with
the pointer (0x6dfed4)
cout << ptr << "\n";

// Dereference: Output the value of food with the
pointer (Pizza)
cout << *ptr << "\n";
```

Note that the \* sign can be confusing here, as it does two different things in our code:

When used in declaration (string\* ptr), it creates a **pointer variable**.

When not used in declaration, it acts as a **dereference operator**.

## C++ Modify Pointers-Modify the Pointer Value

You can also change the pointer's value. But note that this will also change the value of the original variable:

Example

```
string food = "Pizza";
string* ptr = &food;

// Output the value of food (Pizza)
```

```
cout << food << "\n";

// Output the memory address of food (0x6dfed4)
cout << &food << "\n";

// Access the memory address of food and output its value
(Pizza)
cout << *ptr << "\n";

// Change the value of the pointer
*ptr = "Hamburger";

// Output the new value of the pointer (Hamburger)
cout << *ptr << "\n";

// Output the new value of the food variable (Hamburger)
cout << food << "\n";
```

## And why is it useful to know the memory address?

**References** and **Pointers** (which you will learn about in the next chapter) are important in C++, because they give you the ability to manipulate the data in the computer's memory - **which can reduce the code and improve the performance.**

These two features are one of the things that make C++ stand out from other programming languages, like Python and Java.

مثال 1: مقدمة عن المرجع :

```
#include <iostream>

using namespace std;

int main()
{
    int x = 6;

    int&y = x;
```

```

cout << "x=" << x << "\t" << "y=" << y << endl;

x++;

cout << "x=" << x << "\t" << "y=" << y << endl;

y*=3;

cout << "x=" << x << "\t" << "y=" << y << endl;

cout << "-----" << endl;

x=6;

cout << "x=" << x << "\t" << "y=" << y << endl;

cout << "-----" << endl;

int z=x;

cout << "x=" << x << "\t" << "z=" << z << endl;

x++;

cout << "x=" << x << "\t" << "z=" << z << endl;

z*=3;

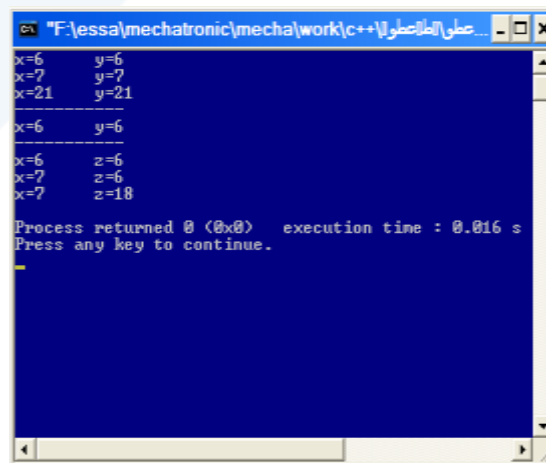
cout << "x=" << x << "\t" << "z=" << z << endl;

return 0;

}

```

خروج البرنامج:



```

F:\yessa\mechatronic\mecha\work\c++\عطى\المطاطو\...
x=6      y=6
x=7      y=7
x=21     y=21
-----
x=6      y=6
-----
x=6      z=6
x=7      z=6
x=7      z=18
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.

```

استدعاء التابع : مثال 2: يتضح في هذا المثال الفرق بين استدعاء التابع بالقيمة و استدعاء التابع بالمرجع:

```
#include <iostream>

using namespace std;

void f(int x,int &y)
{
    x=88;
    y=99;
}

int main()
{
    int a=22,b=33;

    cout<<"local a in main before calling f is:"<<a<<endl;

    cout<<"local b in main before calling f is:"<<b<<endl;

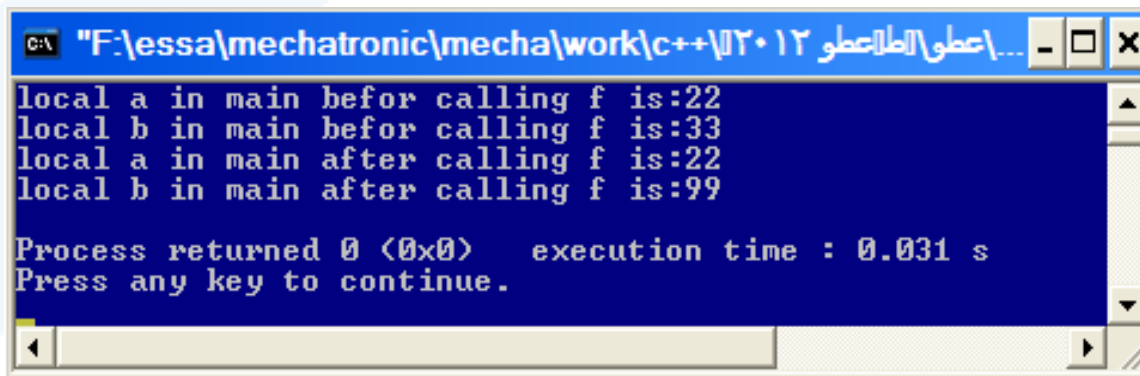
    f(a,b);

    cout<<"local a in main after calling f is:"<<a<<endl;

    cout<<"local b in main after calling f is:"<<b<<endl;

    return 0;}
```

خرج البرنامج:



```
C:\> "F:\essa\mechatronic\mecha\work\c++\2012\اعطوا\اطاعوا...
local a in main befor calling f is:22
local b in main befor calling f is:33
local a in main after calling f is:22
local b in main after calling f is:99

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

مثال 3: أكتب واختبر التابع intpower الذي يعيد قيمة العبارة  $base^{exponent}$  :

```
#include <iostream>

using namespace std;

void intpower(int base ,int exponent ,long int &result)
{
    result=1;
    for(int i=1;i<= exponent ;i++)
        result=result*base;
}

long int intpower(int base ,int exponent)
{
    long int result=1;
    for(int i=1;i<= exponent ;i++)
        result=result*base;
    return result;
}

int main()
{
    int b,e;
```

```
long int r;
while(true)
{
cout<<"enter a base: ";
cin>>b;
cout<<"enter a positive exponent: ";
cin>>e;
if(b==0 && e==0)break;
intpower(b,e,r);
cout<<b<<" power "<<e<<" = "<<r<<endl;
r=intpower(b,e);
cout<<b<<" power "<<e<<" = "<<r<<endl;
}
return 0;
}
```

مثال 4: طور البرنامج السابق ليصبح بإيجاد  $\text{base}^{\text{exponent}}$  مهما يكن العدد الصحيح exponent موجب أو سالب.

```
#include<iostream>
#include<stdlib.h>
using namespace std;
void intpower(int base ,int exponent ,double &result)
{
result=1;
if(exponent==0)result=1;
else if(exponent>0){
for(int i=1;i<= exponent ;i++)
```

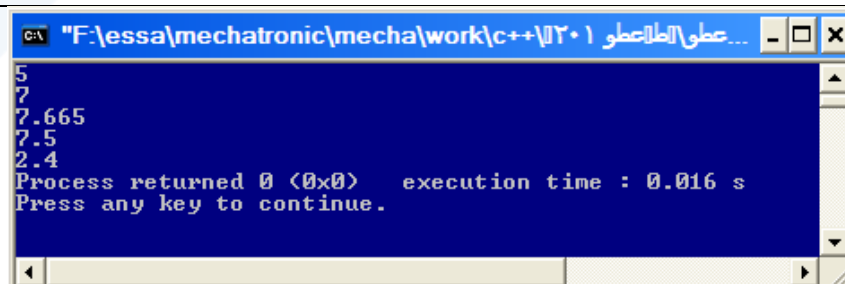


```
        result=result*base;}
    else {exponent=-1*exponent;
        for(int i=1;i<= exponent ;i++)
            result=result*base;
        result=1/result;}
}
int main()
{
    int b,e;
    double r;
    char ch;
    do{
        cout<<"enter a base: ";
        cin>>b;
        cout<<"enter a exponent: ";
        cin>>e;
        intpower(b,e,r);
        cout<<b<<" power "<<e<<" = "<<r<<endl;
        cout<<"press anything to continue and n to finish :";
        cin>>ch; if(ch=='n')break;}while(1);
    return 0;}
```

مثال 5:التحميل الزائد للتوابع:

```
#include <iostream>
using namespace std;
void f1(int a, int b)
```

```
{  
    cout<< a/b << endl;  
}  
void f1(double a, double b)  
{  
    cout<< a/b << endl;  
}  
double f1(int a, double b)  
{  
    return a/b ;  
}  
int main()  
{  
    f1(15,3);  
    f1(15,2);  
    f1(15.33,2.0);  
    f1(15.0,2.0);  
    cout<<f1(12,5.0);  
    return 0;  
}
```



```
C:\> "F:\essa\mechatronic\mecha\work\c++\1301\اطلاعو... - - X  
5  
7  
7.665  
7.5  
2.4  
Process returned 0 (0x0) execution time : 0.016 s  
Press any key to continue.
```

مثال: اكتب برنامج يقوم بالبحث عن سلسلة ضمن سلسلة أخرى ويعيد تابعه  $\text{int}^* \text{loc}(a1, a2, s1, s2)$  مؤشراً على مكان وجود السلسلة الجزئية ضمن الأخرى.

```
int a1[9]={11,11,11,11,11,22,33,44,55};
```

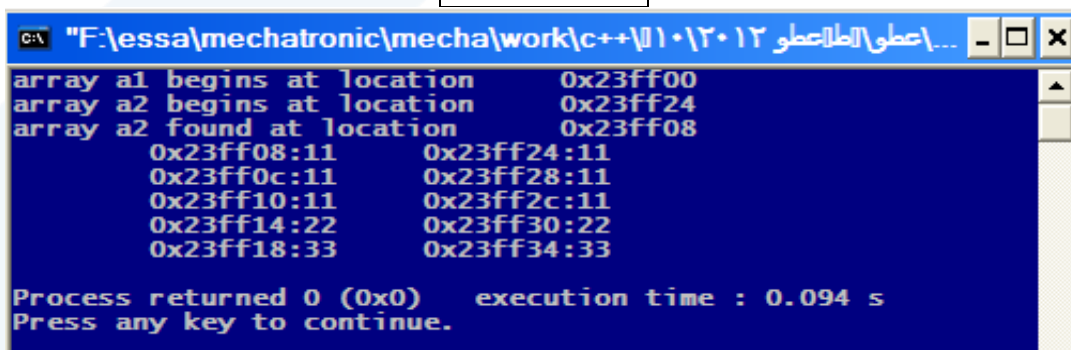
```
int a2[5]={11,11,11,22,33};
```

**Array a1 begins at location 0x23ff00**

0x23ff00:11	0x23ff14:22
0x23ff04:11	0x23ff18:33
0x23ff08:11	0x23ff1c:44
0x23ff0c:11	0x23ff20:55
0x23ff10:11	

**Array a2 begins at location 0x23ff24**

0x23ff24:11
0x23ff28:11
0x23ff2c:11
0x23ff30:22
0x23ff34:33



```

C:\ "F:\essa\mechatronic\mecha\work\c++\10\2012\اعطوا\اطعوا\... - □ ×
array a1 begins at location 0x23ff00
array a2 begins at location 0x23ff24
array a2 found at location 0x23ff08
0x23ff08:11 0x23ff24:11
0x23ff0c:11 0x23ff28:11
0x23ff10:11 0x23ff2c:11
0x23ff14:22 0x23ff30:22
0x23ff18:33 0x23ff34:33

Process returned 0 (0x0) execution time : 0.094 s
Press any key to continue.

```

```
#include <iostream>

using namespace std;
```

```
int* loc(int*a1,int *a2,int n1,int n2)
{
    int *endl=a1+n1;
    int j;
    for(int *p1=a1;p1<endl;p1++)
        if(*p1==*a2){
            for(j=0;j<n2;j++)
                if(p1[j]!=a2[j])break;
            if(j==n2)return p1;}
    return 0;
}

int main()
{
    int a1[9]={11,11,11,11,11,22,33,44,55};
    int a2[5]={11,11,11,22,33};
    cout<<"array a1 begins at location\t"<<a1<<endl;

    cout<<"array a2 begins at location\t"<<a2<<endl;
    int *p=loc(a1,a2,9,5);
    if(p){
        cout<<"array a2 found at location\t"<<p<<endl;
        for(int i=0;i<5;i++)
            cout<<"\t"<<&p[i]<<":"<<p[i]
            <<"\t"<<&a2[i]<<":"<<a2[i]<<endl;
```

```

    }

    else cout<<"not found.\n";

    return 0;

}

```

المثال 1 : تمرير مؤشر لتابع:

```

#include <iostream>

using namespace std;

void cube(int *p)
{*p=*p**p**p;}

int main(){

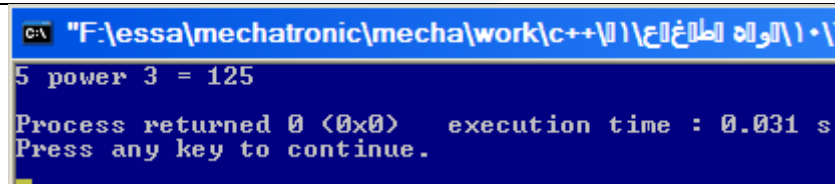
int number=5;

cube(&number);

cout<<"5 power 3 = "<<number<<endl;

return 0;}

```



```

C:\F:\essa\mechatronic\mecha\work\c++\الاولاء\الاغلاغ\1017
5 power 3 = 125
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.

```

المثال 2 : المصفوفات الديناميكية: توقع خرج البرنامج التالي:

```

#include <iostream>

using namespace std;

int main()

{int *p1=NULL,v1,v2;

p1=&v1;

*p1=5;

cout<<v1<<endl;

```

```
cout<<p1<<endl;

v2=9;

p1=&v2;

cout<<*p1<<endl;

cout<<p1<<endl;

cout<<v2<<endl;

p1=new int[v2];

for(int i=0;i<v2;i++)

p1[i]=3*i;

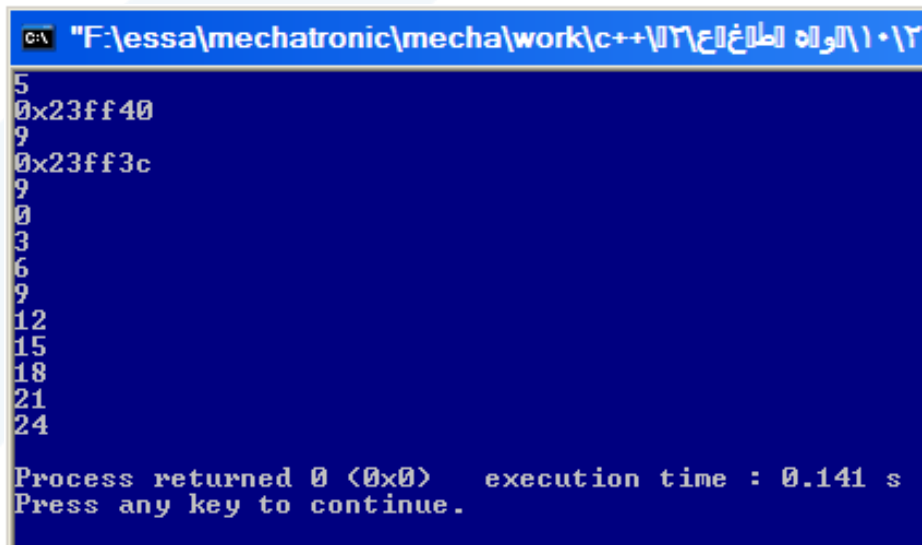
for(int i=0;i<v2;i++)

cout<<*(p1+i)<<endl;

delete p1;

p1=NULL;

return 0;}
```



```
C:\F:\essa\mechatronic\mecha\work\c++\الاول\10\12\اطلاع\2023\
5
0x23ff40
9
0x23ff3c
9
0
3
6
9
12
15
18
21
24

Process returned 0 (0x0)   execution time : 0.141 s
Press any key to continue.
```

**المثال 3:** نريد تطوير برنامج لدائرة الامتحانات ليسمح بإدخال درجات عدد من الطلاب يحدده المستخدم ثم يعرض الخيارات التالية :

1. عرض جدول الدرجات (الرقم , الدرجة).
2. عدد الطلاب الناجحين.
3. عدد الطلاب الراسبين.
4. نسبة النجاح.
5. حساب متوسط الدرجات.
6. أعلى درجة ورقم صاحبها.
7. أدنى درجة ورقم صاحبها.

**ملاحظة :** اكتب تابعا للإدخال و تابعا لكل خيار و استخدم فكرة المصفوفات الديناميكية و البحث الخطي.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int findMin(int[],int);
int findMax(int[],int);
float avg(int[],int);
int failed(int[],int);
int succeeded(int[],int);
void viewGrades(int[],int);
void menu(int[],int);
void fillArray(int[],int);
int main()
{
    int studentsNum,*G;
```

```
cout << "welcome to exam statistic program!\n"
    << "enter the number of students:";
cin>>studentsNum;
G=new int[studentsNum];
fillArray(G,studentsNum);
while(1){
    menu(G,studentsNum);
}
return 0;
}
int failed(int G[],int S)
{
    int count=0;
    for(int i=0;i<S;i++)
        if(G[i]<60)
            count++;
    return count;
}
int succeeded(int G[],int S)
{
    int count=0;
    for(int i=0;i<S;i++)
        if(G[i]>=60)
            count++;
    return count;
}
```



```
}  
  
void viewGrades(int G[],int S)  
{  
    system("CLS");  
    cout<<"NO.\tGrade\n";  
    for(int i=0;i<S;i++)  
        cout<<i+1<<"\t"<<G[i]<<endl;  
}  
  
float avg(int G[],int S)  
{  
    float sum=0;  
    for(int i=0;i<S;i++)  
        sum+=G[i];  
    return sum/S;  
}  
  
int findMax(int G[],int S)  
{  
    int Maxindex=0;  
    for(int i=1;i<S;i++)  
        if(G[Maxindex]<G[i])  
            Maxindex=i;  
    return Maxindex;  
}  
  
int findMin(int G[],int S)
```

```
{
    int Minindex=0;
    for(int i=1;i<S;i++)
        if(G[Minindex]>G[i])
            Minindex=i;
    return Minindex;
}

void menu(int G[],int S)
{
    int c=0;
    system("CLS");
    cout<<"1- View grades table.\n"
        <<"2- number of succeeded students.\n"
        <<"3- number of failed students.\n"
        <<"4- success rate.\n"
        <<"5- Average\n"
        <<"6- Max grade and owner NO.\n"
        <<"7- Min Grade and owner NO.\n"
        <<"8- exit.\n"
        <<"enter your choice:";
    cin>>c;
    switch(c)
    {
        case 1:
            viewGrades(G,S);break;
    }
```

```
case 2:
    cout<<succeeded(G,S)<<endl;break;
case 3:
    cout<<failed(G,S)<<endl;break;
case 4:
    cout<<(float)succeeded(G,S)/failed(G,S)<<endl;break;
case 5:
    cout<<avg(G,S)<<endl;break;
case 6:
    cout<<"max grade is "<<G[findMax(G,S)]
        <<" and NO.="<<findMax(G,S)+1<<endl;break;
case 7:
    cout<<"min grade is "<<G[findMin(G,S)]
        <<" and NO.="<<findMin(G,S)+1<<endl;break;
case 8:
    delete G;exit(1);break;
}
system("PAUSE");
system("CLS");
}
void fillArray(int G[],int S)
{
    system("CLS");
    cout<<"enter students grades from student NO. 1 to "<<S<<"\n";
    for(int i=0;i<S;i++)
```

```
{cin>>G[i];  
  if(G[i]<0 || G[i]>100){cout<<"wrong try again:";i--;}  
}  
}
```

انتهت المحاضرة