

## #37 Data structures (struct)

بإمكانني أنشئ datatype يستقبل كل الأشياء الي باستقبلها ال datatypes  
التانيين يعني اني اعمل datatype يمكنني من اني احط رقم عشري و int و  
char و string و كله ب datatype واحد والي هو (struct (structure،  
يعني لو بدي اكتب خصائص سيارة زي مثلا لونها و سرعتها و موديلها و اسمها  
هاد كله بحتاج datatype غير عن الثاني فبستخدم بهاي الحالة ال struct  
باني اول اشي بكتب struct قبل ال main و بعدين اسم المتغير و بعدين بفتح  
قوسين و بحط semicolon: { } وبكتب فيها ال datatypes الي بدي  
اجمعهم بالمتغير و بعمل initialization بال main:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{
```

```
    string name;
```

```
    string color;
```

```
    int maxspeed;
```

```
    int model;
```

```
};
```

```
int main() {
```

```
    car x;
```

```
    x.name = "rolls_royce";
```

```
    x.color = "red";
```

```
    x.model = 2019;
```

```
    x.maxspeed = 270;
```

```
    cout << x.name << endl;
```

```
    cout << x.model << endl;
```

```
    cout << x.maxspeed << endl;
```

```
    cout << x.color << endl;
```

```
return 0; }
```

```
--->
```



rolls\_royce

2019

270

red

و طريقة ثانية لل initialization اني افتح اقواس و اعمل pass لقيم كل data item لحاله (بس بهاي الحالة لازم اعمل pass بالترتيب):

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{
```

```
    string name;
```

```
    string color;
```

```
    int maxspeed;
```

```
    int model;
```

```
};
```

```
int main() {
```

```
    car x = { "BMW", "Blue", 250, 2016 };
```

```
    return 0; }
```

عشان اطبع او عشان اعمل cout لل data items هون بختلف عن ال array العادية بانو بال struct بنوصل لأي عنصر عن طريق اسمه مش ال index تبعه فما بزيط اني اعمل cout هيك: `cout<< x[0] << endl;` هاد غلط الطريقة الي اتوق الوحيدة هي اني اكتب cout لكل عنصر:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{
```

```
    string name;
```

```
    string color;
```

```
    int maxspeed;
```

```

    int model;
};

int main() {
    car x = { "BMW","Blue",250,2016 };
    cout<< x.name << endl;
    cout<< x.color << endl;
    cout<< x.maxspeed << endl;
    cout<< x.model << endl;
    return 0; }

```

---> 

BMW

Blue

250

2016

لو بدى اعمل assignment (إسناد) انى مثلا اخزن

ال **x**: { "BMW","Blue",250,2016 } car **x** داخل ال y بس بكتب:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{
```

```
    string name;
```

```
    string color;
```

```
    int maxspeed;
```

```
    int model;    };
```

```
int main() {
```

```
    car x = { "BMW","Blue",250,2016 };
```

```
    car y = x;
```

```
    cout << y.name;
```

```
    return 0; }
```

---> BMW

There r 2 ways to declare structure variable:

1. By struct keyword within main() function

2. By declaring a variable at the time of defining the structure

2: ↓

```
struct car
```

```
{  
    string name;  
    string color;  
    int maxspeed;  
    int model; }y;
```

اذا بدي أقارن بين two structures او بين عنصرين (comparsion) لازم  
اقارن كل data item **لوحده** يعني:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{  
    string name;  
    string color;  
    int maxspeed;  
    int model; }y;
```

```
int main() {
```

```
    car x = { "BMW","Blue",250,2016 };
```

```
    car y = { "Mercedes","Red",300,2016 };
```

```
    if (x.maxspeed > y.maxspeed)
```

```
        cout << "x greater than y\n";
```

```
    else
```

```
        cout << "y is greater than x";
```

```
    return 0; }
```

و بإمكانني اكتب: `if (x.maxspeed > y.model)`

لو بدني اعمل `pass structure` ل `function`:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{ string name;
```

```
  string color;
```

```
  int maxspeed;
```

```
  int model; };
```

```
void f(car f)
```

```
{ cout << "Name = " << f.name << endl;
```

```
  cout << "Color = " << f.color << endl; }
```

```
int main() {
```

```
  car v = { "no name","red",100,200 };
```

```
  f(v);
```

```
return 0; }
```

---> 

Name = no name

Color = red

الي صار اني لما كتبت `f(v)` هيك استدعيت ال `function` وعملت `pass` ل `v`  
عوضا عن ال `f` وكمل الكود بتعويض ال `f` مع ال `v`.

لو بدني أعمل عملية إدخال لعناصر ال `struct (cin)`، رح اعمل `function`  
يستقبل `struct` معين وياخذ القيم عن طريق ال `user` و بعدين يعمل `return`  
ل ال `struct`. عملت ال `function` يكون ال `return value type` تبعه `struct` او  
`car` واحط بين القوسين انو بدني اياه يستقبل `struct(s)` من  
ال `datatype(car)` وعملته `call by reference (&)` وعملت `cin` لكل `data`  
`item` لحال و عملت `return` ل ال `s` والي فيها ال `name, model`  
`maxspeed,color`, و عملت `struct` اسمه `v` واستدعيت  
ال `function (read_return)` وبعته `v` وعملت `struct` ثاني الي هو `h`

واسندته للv:

```
#include<iostream>
#include<string>
using namespace std;
struct car
{
    string name;
    string color;
    int maxspeed;
    int model; };
car read_return(car&s)
{
    cout << "Enter name:\n";
    cin >> s.name;
    cout << "Enter color:\n";
    cin >> s.color;
    cout << "Enter maxspeed:\n";
    cin >> s.maxspeed;
    cout << "Enter model:\n";
    cin >> s.model;
    return s; }
int main() {
    car v;
    read_return(v);
    car(h);
    h = v;
    cout << h.name << endl;
    return 0; }
```

---> 

Enter name:

kia

Enter color:

white

Enter maxspeed:

200

Enter model:

2010

kia

(compiler | user)

ممکن اني اعرف function داخل struct:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{ string name;
```

```
  string color;
```

```
  int maxspeed;
```

```
  int model;
```

```
  void fun(car n) {
```

```
    cout << n.name << endl;  }  };
```

```
int main() {
```

```
  car v = { "kia" };
```

```
  v.fun(v);
```

```
  return 0; }
```

---> kia

ممکن اربط struct pointer بعنوان متغير struct:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct car
```

```
{ string name;
```

```
  string color;
```

```

int maxspeed;
int model;  };

int main() {
    car* h;
    car b = { "A","C",200,2011 };
    h = &b;
    cout << h->color << endl;
    return 0;  }

```

---> C

لما اعمل cout ل struct pointer بحط سهم -> مش نقطة

بإمكاني احط struct داخل struct يعني ممكن اعرف 3 متغيرات مع بعض  
 مثلا تلت متغيرات عن ال name او ال address اني مثلا ساكنة بالقارة...  
 والمدينة... والشارع... . ومثال لو بدي اكتب ال first name و ال last name:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct student
```

```
{ int id;
```

```
fullname name;  };
```

```
struct fullname {
```

```
    string fName;
```

```
    string lName;  };
```

```
int main() { student a;
```

```
    a.name.fName = "Ahmad";
```

```
    a.name.lName = "Jabr";  }
```

## #38 String

ال datatype string بخزن قيمة نصية و هو احسن من ال array of character و عشان استفيد من كل مزاي ال string بكتب



`#include<string>` ال string هو نوع بيانات جديد باللغة معاه مجموعة

functions عشان تسهل العمل عال string.

دايمة لما اعمل إسناد او assignment بكتب:

```
string x="ahmad",y;
```

```
y = x;
```

```
cout << "y = " << y << endl;
```

بس بإمكانني احط بدل ال = نقطة و رح يطلع functions كتار فاخترت

assignment اختصار

```
string x="ahmad",y;
```

```
y.assign(x);
```

```
cout << "y = " << y << endl;
```

```
---> y = ahmad
```

دخلتهم بال cin و كتبت get line عشان ياخذ بعين الاعتبار ال space:

```
string x="ahmad",y;
```

```
getline(cin,x);
```

```
y.assign(x);
```

```
cout << "x = " << x << endl;
```

```
---> aa bb cc
```

```
y = aa bb cc
```

```
(compiler | user)
```

لو بدي اطبع بس اول حرف او character من ال string بستخدم function

ال at هو بستقبل رقم داخل اقواس والرقم ال index تبعه 0، ال string يبدأ

ال index تاعه من 0 يعني لما اكتب 0 بطبع اول حرف و اذا 1 بطبع ثاني و

هيك....:

```
string x="a8hmad",y;
```

```
cout << x.at(0) << endl;
```

```
---> a
```

و function ال length وال size بحكولي حجم او عدد ال string:

```
string x="a8hmad",y;
```

```
cout << x.length() << endl;
```

or

```
string x="a8hmad",y;
```

```
cout << x.size() << endl;
```

---> 6

بس ال size احسن لانو ال length بغلب بال complexity.

وفي ال substr وهو بطبع جزء من ال string و يستقبل 2 parameters يعني بكتب بالاقواس ال positions مثلا 1,3 هيك رح يطبع الأحرف الي بين ال positions:

```
string x="ahmad",y;
```

```
cout << x.substr(1,3) << endl;
```

---> hma

بس لو استقبل ال substr parameter واحد بس، بفهمها بانو يطبع او يبدأ من ال index مثلا 4 و طالع:

```
string x="ahmad12345",y;
```

```
cout << x.substr(4) << endl;
```

---> d12345

لما اعمل swap بدل ما اعرف متغير تالت و عشان أوفر مساحة تخزينية بستخدم ال function swap:

```
string x="ahmad12345",y="22";
```

```
x.swap(y);
```

```
cout << x << endl;
```

---> 22

في ال function find و هو اني بيعتله اشي و هو بدور عليه بانو يطبع ال position الي موجود فيه الاشي:

```
string x="ahmad12345",y="22";
```

```
cout << x.find("a") << endl;
```

---> 0

بس في اثنين a فلو بدي اياه يطبع ال a الثانية رح اخلي يدور من اليمين لليسا

باني بكتب rfind:

```
string x="ahmad12345",y="22";
```

```
cout << x.rfind("a") << endl;
```

---> 3

لو كتبت string مش حرف رح يطبع ال position الأول الي لاقى فيه اول حرف من ال string:

```
string x="ahmad12345",y="22";
```

```
cout << x.find("12") << endl;
```

---> 5

و function ال erase يستقبل 2 parameters مثلا 0,3 هيك رح يمحي الأشياء الي بال position من 0 ل3:

```
string x="ahmad12345",y="22";
```

```
cout << x.erase(0,3) << endl;
```

---> ad12345

قيمة ال x بتصير نفس القيمة بعد القص:

```
string x="ahmad12345",y="22";
```

```
cout << x.erase(0,3) << endl;
```

```
cout << x << endl;
```

---> ad12345

ad12345

اذا بعته parameter واحد مثلا 3 رح يطبع اول تلت أحرف بس:

```
string x="ahmad12345",y="22";
```

```
cout << x.erase(3) << endl;
```

---> ahm

و في function ال replace و يستقبل 3 parameters الأول و هو موقع الاستبدال يعني من وين ابدأ عملية الاستبدال فمثلا من ال 5 position يعني رح يبدأ يستبدل الي بعد ال position الخامس يعني من ستة رح يبدأ يعد لل parameter الثاني الي هو حجم الاستبدال فمثلا خمسة فيعني انو رح يعد من واحد لخمسة من بعد ال position المكتوب بال parameter الي قبل والي هو برضو خمسة، وال parameter الثالث هو ال string الي بدي ابدل

فيها:

```
string x="ahmad12345",y="22";  
cout << x.replace(5,5," jabr") << endl;
```

---> ahmad jabr

اسهل و مركب ع بعض اني استخدم function ال replace و ال find مع بعض:

```
string x="ahmad12345",y="22";  
cout << x.replace(x.find("d"), 6, " jabr") << endl;
```

---> ahma jabr

في function ال insert ب مستقبل 2 parameters الأول هو ال position  
للمكان الي بدي اعمله insert و بال parameters الثاني بكتب بين  
quotation marks الاشئ الي بدي اعمله insert:

```
string x="ahmad12345",y="22";  
cout << x.insert(5," jabr ") << endl;
```

---> ahmad jabr 12345