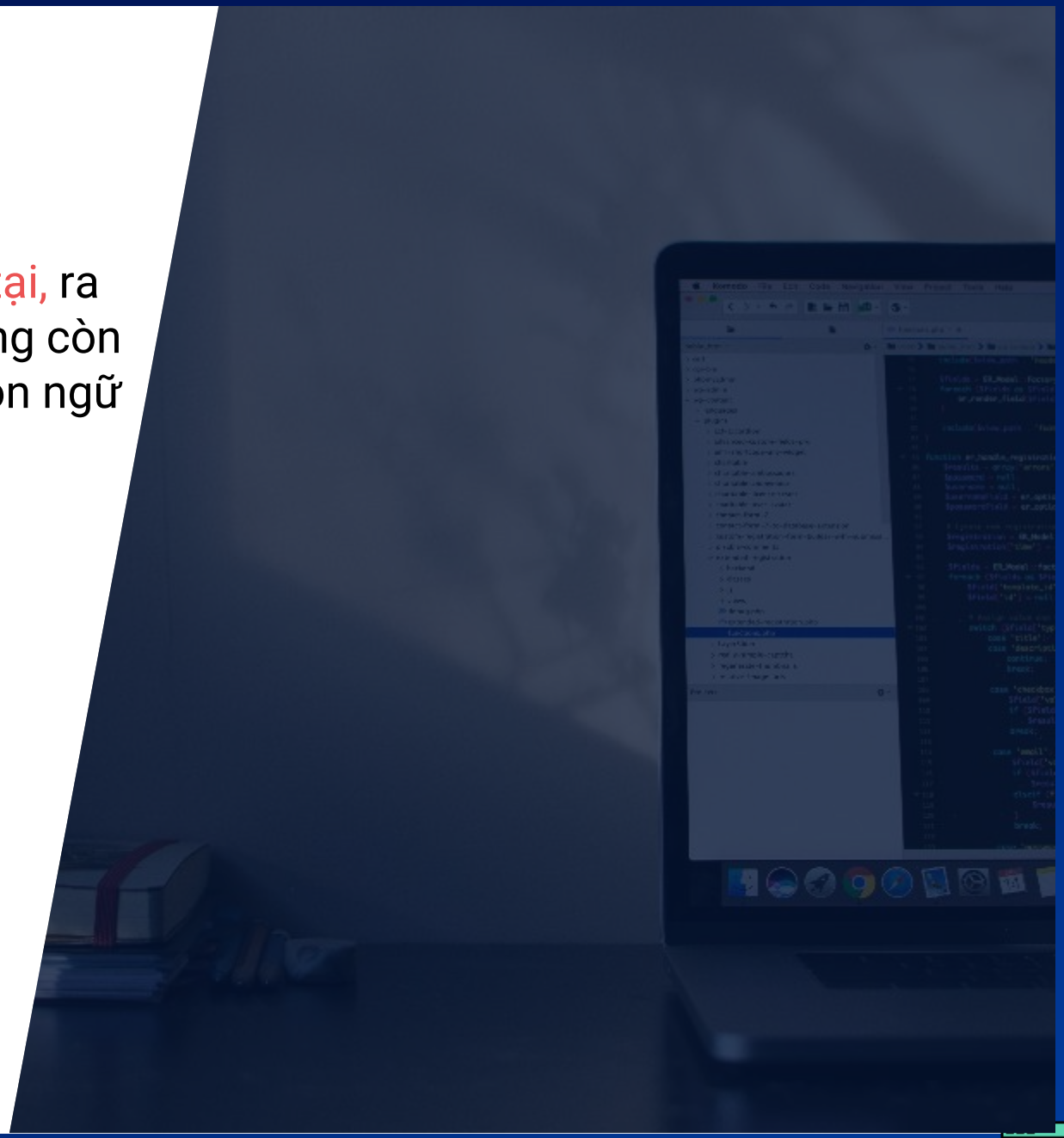
PHẠM VI CỦA BIẾN (SCOPE OF VARIABLE)

Phạm vi của một biến:

Là phạm vi mà tại đó biến được khai báo và tồn tại, ra ngoài phạm vi mà biến được khai báo nó sẽ không còn được biết đến nữa. Có 3 phạm vi chính trong ngôn ngữ lập trình C đó là:

- **Phạm vi trong đóng mở ngoặc nhọn (Enclosing scope)**
- **Phạm vi địa phương (Local scope)**
- **Phạm vi toàn cục (Global scope)**



Phạm vi trong đóng mở ngoặc nhọn (Enclosing scope)

Ví dụ 1:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    int n = 100;
    if(n == 100){
        int m = 200;
        cout << m << endl; // OK
    }
    cout << m << endl; // Lỗi
}
```

Khi bạn khai báo một biến bên trong một đóng mở ngoặc nhọn, ví dụ như bên trong if else, for, while...

Giá trị của m được khai báo bên trong phạm vi đóng mở ngoặc nhọn của if vì thế ra bên ngoài phạm vi này, m sẽ không được biết đến và vì thế không thể truy cập giá trị

Phạm vi trong đóng mở ngoặc nhọn (Enclosing scope)

Ví dụ 2:

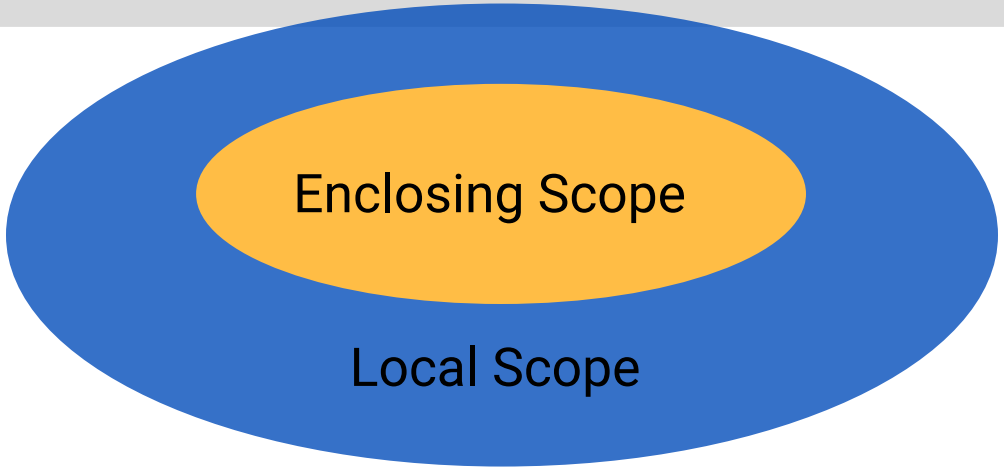
```
#include <bits/stdc++.h>
using namespace std;

int main(){
    for(int i = 1; i <= 10; i++){
        cout << "28tech !\n";
    }
    cout << i << endl; // Lỗi
    int i;
    for(i = 1; i <= 10; i++){
        cout << "28tech !\n";
    }
    cout << i << endl; // OK
}
```

Lỗi vì biến i chỉ được khai báo trong vòng for

Hợp lệ vì biến i được khai báo bên trên vòng for

Phạm vi địa phương (Local scope)

Phạm vi địa phương hay local scope là phạm vi khi biến được khai báo trong hàm. Phạm vi này lớn hơn phạm vi đóng mở ngoặc nhỏ ở mục trên.



Hai biến ở trong 2 phạm vi đóng mở ngoặc nhọn hoặc phạm vi địa phương có thể cùng tên, khi bạn truy cập giá trị của 1 biến cùng tên ở trong 2 phạm vi này thì quy tắc tìm kiếm sẽ tìm từ phạm vi nhỏ hơn tới phạm vi lớn hơn.

Phạm vi địa phương (Local scope)

Ví dụ 1:

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    int n = 100;
    if(n == 100){
        int n = 1000;
        cout << n << endl; //Enclosing scope
    }
    cout << n << endl; // Local
}
```

OUTPUT: 1000
100

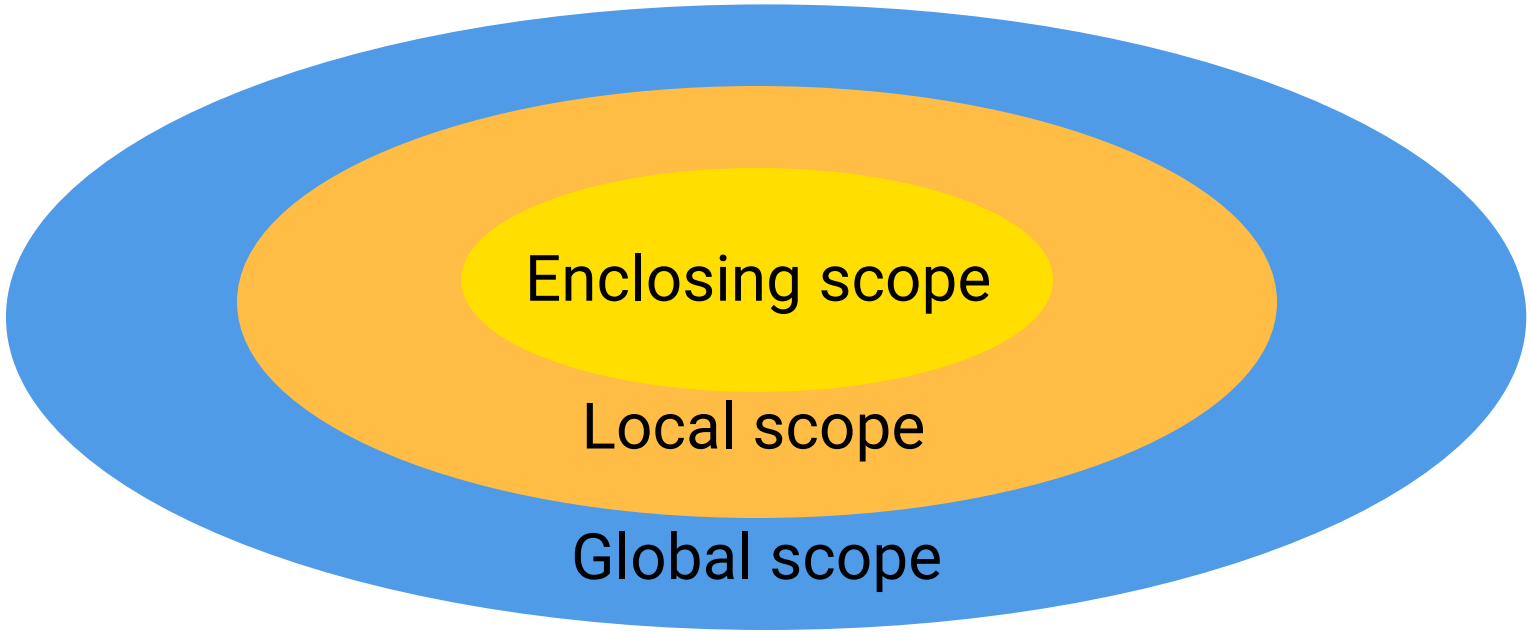
Ví dụ 2:

```
#include <bits/stdc++.h>
using namespace std;

void scope(){
    int n = 100;
}

int main(){
    scope();
    cout << n << endl; // 100 ??
}
```

Bạn không thể truy cập giá trị của n ở 1 phạm vi địa phương khác, trong trường hợp này là phạm vi địa phương của hàm scope.

Phạm vi toàn cục (Global scope)

Một biến được khai báo bên ngoài phạm vi các hàm, biến này sẽ có vùng hoạt động là cả chương trình và được gọi là biến toàn cục. Phạm vi toàn cục có phạm vi lớn hơn so với phạm vi địa phương và phạm vi đóng mở ngoặc nhọn



Phạm vi toàn cục (Global scope)

Ví dụ 1:

```
#include <bits/stdc++.h>
using namespace std;

int n = 10; // global

int main(){
    int n = 100; // local
    if(1){
        int n = 1000; //enclosing scope
        cout << n << endl;
    }
    cout << n << endl;
}
```

OUTPUT: 1000
100

Ví dụ 2:

```
#include <bits/stdc++.h>
using namespace std;

int n = 10; // global

int main(){
    if(1){
        int n = 1000; //enclosing scope
        cout << n << endl;
    }
    cout << n << endl;
}
```

OUTPUT: 1000
10

Khi ở các vùng phạm vi này có một biến có cùng tên thì quy tắc tìm kiếm khi bạn truy cập là từ vùng phạm vi nhỏ nhất tới vùng có phạm vi lớn nhất

Phạm vi toàn cục (Global scope)

Ví dụ 3:

```
#include <bits/stdc++.h>
using namespace std;
int n = 10; // global

void scope1(){
    cout << n << endl;
}

void scope2(){
    int n = 100;
    cout << n << endl;
}

int main(){
    scope1();
    scope2();
}
```

OUTPUT:

10
100

Vì biến toàn cục thuộc về phạm vi của toàn chương trình nên ta hoàn toàn có thể truy cập nó từ các hàm nhỏ cũng như hàm main