

# Chương 01

## CHƯƠNG 1 – ÔN TẬP C++

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

Target

# 1. MỤC TIÊU

# Mục tiêu

- Ôn lại các khái niệm, các kiến thức nhập môn lập trình.
- Biến toàn cục (global variable).
- Biến cục bộ (local variable).
- Hàm và biến toàn cục.
- Tham số (parameter) và hàm (function).
- Trừu tượng hóa dữ liệu (data abstract).

Problem

## 2. BÀI TOÁN

# Bài toán

—Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

Global variable

## 3. BIẾN TOÀN CỤC

# Biến toàn cục

- Khái niệm: Biến toàn cục (**global variable**) là biến được khai báo bên ngoài tất cả các hàm và được hiểu bên trong tất cả các hàm.



- Tham khảo: [https://en.wikipedia.org/wiki/Global\\_variable](https://en.wikipedia.org/wiki/Global_variable)

# Biến toàn cục

- Khái niệm: Biến toàn cục (**global variable**) là biến được khai báo bên ngoài tất cả các hàm và được hiểu bên trong tất cả các hàm.
- Thông thường biến toàn cục được khai báo ở đầu chương trình.
- Lưu ý: Biến khai báo bên trong thân hàm **main** không là biến toàn cục mà là biến cục bộ của hàm **main**.

— Tham khảo: [https://en.wikipedia.org/wiki/Global\\_variable](https://en.wikipedia.org/wiki/Global_variable)



# Biến toàn cục

—Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Biến toàn cục

```
11.#include <iostream>
```

# Biến toàn cục

```
11.#include <iostream>
```

— Dòng lệnh 11 đọc là: Khai báo sử dụng thư viện **iostream**.

# Biến toàn cục

```
11.#include <iostream>
```

```
12.#include <string>
```

— Dòng lệnh 12 đọc là: Khai báo sử dụng thư viện **string**.

# Biến toàn cục

```
11.#include <iostream>  
12.#include <string>  
13.using namespace std;
```

— Dòng lệnh 13 đọc là: Khai báo sử dụng không gian tên **std**.

# Biến toàn cục

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.string HoTen;
```

— Dòng lệnh 14 đọc là: **HoTen** là một biến có kiểu dữ liệu **string**.

# Biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;

14.string HoTen;
15.int Toan;
```

— Dòng lệnh 15 đọc là: **Toan** là một biến có kiểu dữ liệu **int**.

# Biến toàn cục

```
11.#include <iostream>  
12.#include <string>  
13.using namespace std;
```

```
14.string HoTen;
```

```
15.int Toan;
```

```
16.int Van;
```

— Dòng lệnh 16 đọc là: **Van** là một biến có kiểu dữ liệu **int**.



# Biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;

14.string HoTen;
15.int Toan;
16.int Van;
17.float DiemTrungBinh;
```

Dòng lệnh 17 đọc là:  
**DiemTrungBinh** là  
một biến có kiểu dữ  
liệu **float**.

# Biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;

14.string HoTen;
15.int Toan;
16.int Van;
17.float DiemTrungBinh;
```

# Biến toàn cục

```
18.int main()  
19.{  
20.    cout << "Nhap ho ten: ";  
21.    getline(cin, HoTen);  
22.    cout << "Nhap Toan: ";  
23.    cin >> Toan;  
24.    cout << "Nhap Van: ";  
25.    cin >> Van;  
26.    DiemTrungBinh = (float)(Toan + Van) / 2;
```

# Biến toàn cục

```
27. | cout << "Ho ten: " << HoTen << endl;  
28. | cout << "Toan: " << Toan << endl;  
29. | cout << "Van: " << Van << endl;  
30. | cout << "Diem trung binh:"<<DiemTrungBinh<<endl;  
31. | return 0;  
32. | }
```

```
11. #include <iostream>
12. #include <string>
13. using namespace std;
14. string HoTen;
15. int Toan;
16. int Van;
17. float DiemTrungBinh;
18. int main()
19. {
20.     cout << "Nhap ho ten: ";
21.     getline(cin, HoTen);
22.     cout << "Nhap Toan: ";
23.     cin >> Toan;
24.     cout << "Nhap Van: ";
25.     cin >> Van;
26.     DiemTrungBinh = (float)(Toan + Van) / 2;
27.     cout << "\n Ho ten:" << HoTen;
28.     cout << "\n Toan: " << Toan;
29.     cout << "\n Van: " << Van;
30.     cout << "\n Diem Trung binh:" << DiemTrungBinh;
31.     return 0;
32. }
```

# Biến toàn cục

20. cout << "Nhập họ tên: ";

— Dòng lệnh 20 có hai cách đọc là:

- + Cách 01 – Ý nghĩa câu lệnh: Xuất chuỗi thông báo "Nhập họ tên: ".
- + Cách 02 – Kỹ thuật lập trình: Hàm toán tử xuất (operator <<) được gọi thực hiện với hai đối số cout và chuỗi thông báo "Nhập họ tên: ".

# Biến toàn cục

21. `getline(cin, HoTen);`

— Dòng lệnh 21 có hai cách đọc là:

+ Cách 01 – Ý nghĩa câu lệnh:

- Nhập họ tên.
- Nhập dữ liệu cho biến `HoTen`.

+ Cách 02 – Kỹ thuật lập trình: Hàm `getline` được gọi thực hiện với hai đối số `cin` và biến `HoTen`.



# Biến toàn cục

```
22. cout << "Nhập Toán: ";
```

— Dòng lệnh 22 có hai cách đọc là:

- + **Cách 01 – Ý nghĩa câu lệnh:** Xuất chuỗi thông báo "Nhập Toán: ".
- + **Cách 02 – Kỹ thuật lập trình:** Hàm toán tử xuất được gọi thực hiện với hai đối số **cout** và chuỗi thông báo "Nhập Toán: ".



# Biến toàn cục

23. `cin >> Toan;`

— Dòng lệnh 23 có hai cách đọc là:

+ **Cách 01 – Ý nghĩa câu lệnh:**

- Nhập toán.
- Nhập dữ liệu cho biến `Toan`.

+ **Cách 02 – Kỹ thuật lập trình:** Hàm toán tử vào `>>` được gọi thực hiện với hai đối số `cin` và biến `Toan`.

# Biến toàn cục

24. cout << "Nhap Van: ";

— Dòng lệnh 24 có hai cách đọc là:

- + Cách 01 – Ý nghĩa câu lệnh: Xuất chuỗi thông báo "Nhap Van: ".
- + Cách 02 – Kỹ thuật lập trình: Hàm toán tử xuất được gọi thực hiện với hai đối số cout và chuỗi thông báo "Nhap Van: ".

# Biến toàn cục

25. `cin >> Van;`

— Dòng lệnh 25 có hai cách đọc là:

+ **Cách 01 – Ý nghĩa câu lệnh:**

- Nhập văn.
- Nhập dữ liệu cho biến `Van`.

+ **Cách 02 – Kỹ thuật lập trình:** Hàm toán tử vào `>>` được gọi thực hiện với hai đối số `cin` và biến `Van`.

# Biến toàn cục

```
31. return 0;
```

- Dòng lệnh 31 đọc là: Kết thúc lời gọi hàm (hàm ở đây là hàm **main**) và trả về giá trị 0.

# Biến toàn cục

- Khái niệm: Biến toàn cục (**global variable**) là biến được khai báo bên ngoài tất cả các hàm và được hiểu bên trong tất cả các hàm.



- Tham khảo: [https://en.wikipedia.org/wiki/Global\\_variable](https://en.wikipedia.org/wiki/Global_variable)

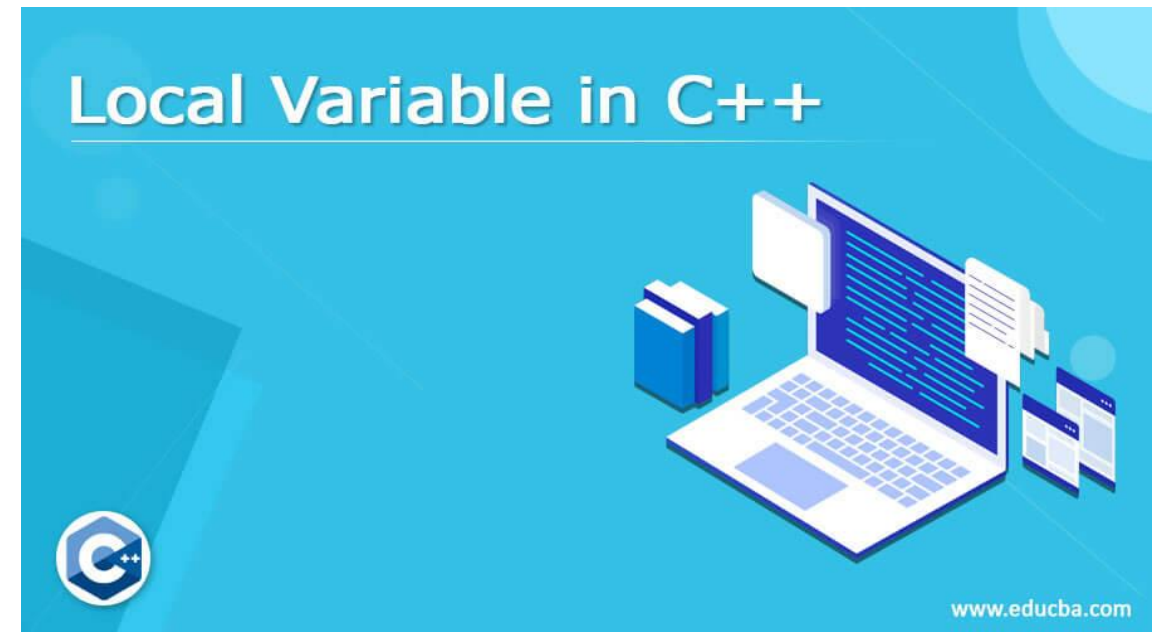
Local variable

## 4. BIẾN CỤC BỘ



# Biến cục bộ

- Khái niệm: Biến cục bộ (**local variable**) là biến được khai báo và được hiểu bên trong một phạm vi (scope) nào đó của chương trình, ra khỏi phạm vi này biến không còn được biết đến nữa vì không gian bộ nhớ cấp phát cho biến được tự động thu hồi.



- Tham khảo: [https://en.wikipedia.org/wiki/Local\\_variable](https://en.wikipedia.org/wiki/Local_variable)

# Biến cục bộ

- Thông thường biến cục bộ được khai báo bên trong thân của một hàm (function) hay một khối lệnh (block).
- Trong ngôn ngữ C/C++ khối lệnh (block) được bắt đầu bằng ký tự { và kết thúc bằng ký tự }.
- Lưu ý: Một biến được khai báo bên trong thân hàm main là biến cục bộ của hàm **main**.

— Tham khảo: [https://en.wikipedia.org/wiki/Local\\_variable](https://en.wikipedia.org/wiki/Local_variable)



# Biến cục bộ

—Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Biến cục bộ

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.int main()
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
```

# Biến cục bộ

```
20. | cout << "Nhap ho ten: ";  
21. | getline(cin, HoTen);  
22. | cout << "Nhap Toan: ";  
23. | cin >> Toan;  
24. | cout << "Nhap Van: ";  
25. | cin >> Van;
```

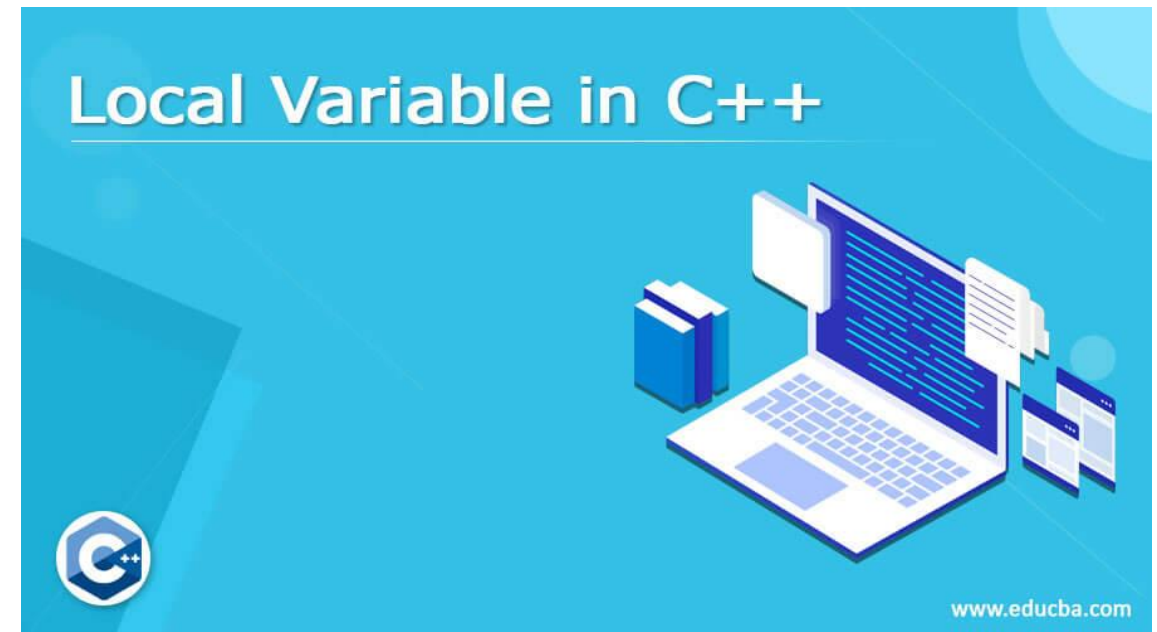
# Biến cục bộ

```
26. DiemTrungBinh = (float)(Toan + Van) / 2;  
27. cout << "Ho ten: " << HoTen << endl;  
28. cout << "Toan: " << Toan << endl;  
29. cout << "Van: " << Van << endl;  
30. cout << "Diem trung binh:"<<DiemTrungBinh<<endl;  
31. return 0;  
32. }
```

```
11. #include <iostream>
12. #include <string>
13. using namespace std;
14. int main()
15. {
16.     string HoTen;
17.     int Toan;
18.     int Van;
19.     float DiemTrungBinh;
20.     cout << "Nhap ho ten:";
21.     getline(cin, HoTen);
22.     cout << "Nhap Toan:";
23.     cin >> Toan;
24.     cout << "Nhap Van:";
25.     cin >> Van;
26.     DiemTrungBinh = (float)(Toan + Van) / 2;
27.     cout << "\n Ho ten:" << HoTen;
28.     cout << "\n Toan: " << Toan;
29.     cout << "\n Van: " << Van;
30.     cout << "\n Diem Trung binh: " << DiemTrungBinh;
31.     return 0;
32. }
```

# Biến cục bộ

- Khái niệm: Biến cục bộ (**local variable**) là biến được khai báo và được hiểu bên trong một phạm vi (scope) nào đó của chương trình, ra khỏi phạm vi này biến không còn được biết đến nữa vì không gian bộ nhớ cấp phát cho biến được tự động thu hồi.



- Tham khảo: [https://en.wikipedia.org/wiki/Local\\_variable](https://en.wikipedia.org/wiki/Local_variable)

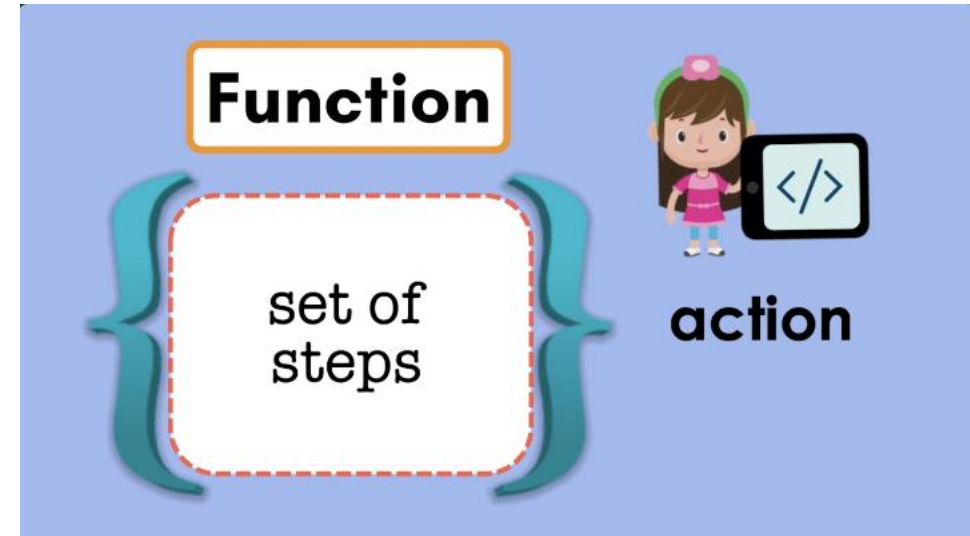
Function and global variable

## **5. HÀM VÀ BIẾN TOÀN CỤC**



# Hàm và biến toàn cục

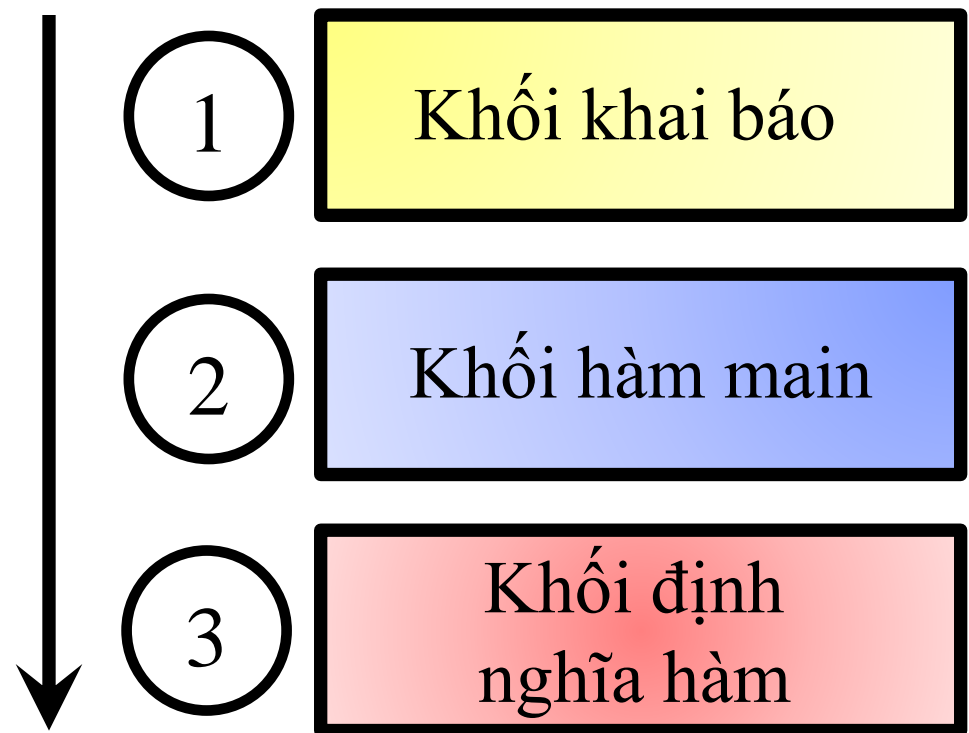
- Kiến trúc chương trình C.
- Khởi khai báo.
- Khối hàm main.
- Khối định nghĩa hàm.





# Hàm và biến toàn cục

- Kiến trúc của một chương trình C/C++ cơ bản bao gồm 3 khối lệnh chính như sau: khối khai báo, khối hàm main và khối định nghĩa hàm. Ba khối lệnh này được trình bày theo thứ tự của hình vẽ bên.



# Hàm và biến toàn cục

- **Khối khai báo (declaration–block):** chứa các khai báo hàm, khai báo biến toàn cục, khai báo sử dụng thư viện, khai báo hằng, khai báo kiểu dữ liệu...
- **Khối hàm main (main–block):** chứa duy nhất hàm `main` và thân hàm của nó. Trong thân hàm `main` chứa các lời gọi hàm cần thiết cho chương trình.
- **Khối định nghĩa hàm (function–definition–block):** chứa các định nghĩa hàm đã được khai báo trong khối khai báo.

# Hàm và biến toàn cục

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Hàm và biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.string HoTen;
15.int Toan;
16.int Van;
17.float DiemTrungBinh;
18.void Nhap();
19.void XuLy();
```

```
20.void Xuat();
21.int main()
22.{
23.    Nhap();
24.    XuLy();
25.    Xuat();
26.    return 0;
27.}
```

# Hàm và biến toàn cục

```
28. void Xuat()  
29. {  
30.     cout << "\n Ho ten:" << HoTen;  
31.     cout << "\n Toan: " << Toan;  
32.     cout << "\n Van: " << Van;  
33.     cout << "\n Trung binh:" << DiemTrungBinh;  
34. }
```

# Hàm và biến toàn cục

```
35. void XuLy()  
36. {  
37. |   DiemTrungBinh = (float)(Toan + Van) / 2;  
38. }
```

# Hàm và biến toàn cục

```
39. void Nhap()  
40. {  
41.     cout << "Nhap ho ten:";  
42.     getline(cin, HoTen);  
43.     cout << "Nhap Toan:";  
44.     cin >> Toan;  
45.     cout << "Nhap Van:";  
46.     cin >> Van;  
47. }
```



# Hàm và biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.string HoTen;
15.int Toan;
16.int Van;
17.float DiemTrungBinh;
18.void Nhap();
19.void XuLy();
```

```
18.void Nhap();
```

— Dòng lệnh 18 đọc là: Khai báo hàm **Nhap**.

+ Hàm **Nhap** không có giá trị trả về.

+ Hàm **Nhap** không có tham số đầu vào.

# Hàm và biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.string HoTen;
15.int Toan;
16.int Van;
17.float DiemTrungBinh;
18.void Nhap();
19.void XuLy();
```

```
19.void XuLy();
```

— Dòng lệnh 19 đọc là: Khai báo hàm XuLy.

+ Hàm XuLy không có giá trị trả về.

+ Hàm XuLy không có tham số đầu vào.

# Hàm và biến toàn cục

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.string HoTen;
15.int Toan;
16.int Van;
17.float DiemTrungBinh;
18.void Nhap();
19.void XuLy();
```

```
20.void Xuat();
```

— Dòng lệnh 20 đọc là: Khai báo hàm **Xuat**.

+ Hàm **Xuat** không có giá trị trả về.

+ Hàm **Xuat** không có tham số đầu vào.

# Hàm và biến toàn cục

23. `Nhap();`

20. `void` `Xuat();`

21. `int` `main()`

22. {

23.     `Nhap();`

24.     `XuLy();`

25.     `Xuat();`

26.     `return 0;`

27. }

— Dòng lệnh 23 đọc là: Hàm `Nhap` được gọi thực hiện.

# Hàm và biến toàn cục

24. XuLy();

20.void Xuat();

21.int main()

— Dòng lệnh 24 đọc là: Hàm **XuLy** được gọi thực hiện.

22.{

23.     Nhap();

24.     XuLy();

25.     Xuat();

26.     return 0;

27.}

# Hàm và biến toàn cục

25. Xuat();

20.void Xuat();

21.int main()

22.{

23.    Nhap();

24.    XuLy();

25.    Xuat();

26.    return 0;

27.}

— Dòng lệnh 25 đọc là: Hàm **Xuat** được gọi thực hiện.

# Hàm và biến toàn cục

26. `return 0;`

20. `void` `Xuat()`;

21. `int` `main()`

— Dòng lệnh 26 đọc là: Kết thúc lời gọi hàm (hàm ở đây là hàm `main`) và trả về giá trị 0.

22. {

23.     `Nhap()`;

24.     `XuLy()`;

25.     `Xuat()`;

26.     `return 0;`

27. }

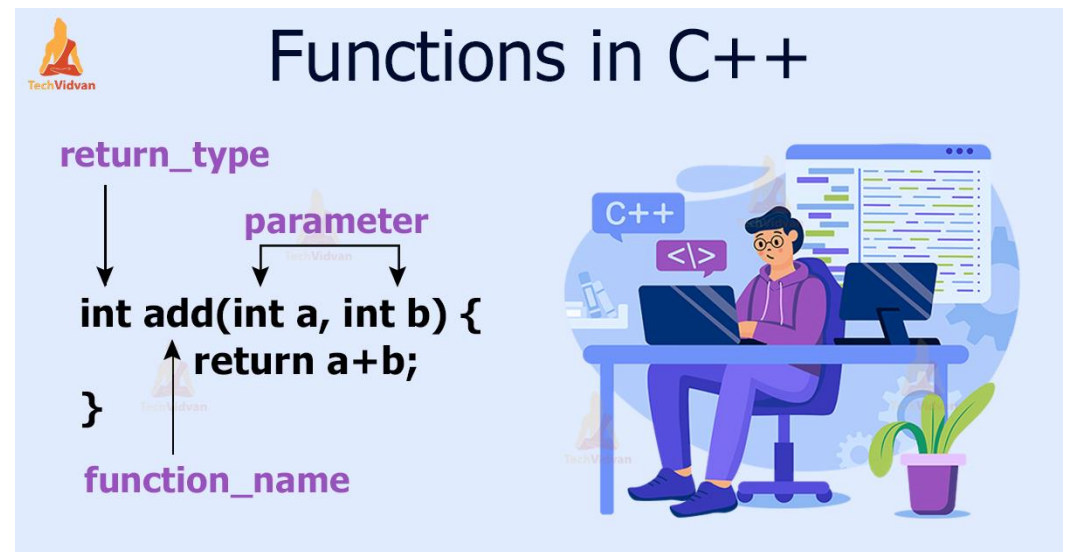


Parameter and function

## 6. THAM SỐ VÀ HÀM

# Tham số và hàm

- Khái niệm: Các thông số đầu vào của một hàm được gọi là tham số (**parameter**) của hàm.



- Tham khảo: <https://en.wikipedia.org/wiki/Parameter>

# Tham số và hàm

- Khái niệm: Các thông số đầu vào của một hàm được gọi là tham số (**parameter**) của hàm.
- Phân loại tham số: có 2 loại tham số là tham trị và tham biến.
  - + Tham trị (**argument passed by value**): Không đổi.
  - + Tham biến – tham chiếu (**argument passed by reference**): Thay đổi.

— Tham khảo: <https://en.wikipedia.org/wiki/Parameter>

# Tham số và hàm

- Cấp phát bộ nhớ:
  - + Tham trị (**argument passed by value**): Cấp phát bộ nhớ khi hàm được gọi thực hiện.
  - + Tham biến (**argument passed by reference**): Không cấp phát bộ nhớ khi hàm được gọi thực hiện mà sử dụng bộ nhớ của đối số tương ứng.

— Tham khảo: <https://en.wikipedia.org/wiki/Parameter>

# Tham số và hàm

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Tham số và hàm

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.void Nhap(string&, int&, int&);
```

```
15.void XuLy(int, int, float&);
```

```
16.void Xuat(string, int, int, float);
```

# Tham số và hàm

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.void Nhap(string&, int&, int&);
```



# Tham số và hàm

14. void Nhap(string&, int&, int&);

— Dòng lệnh 14 đọc là: Khai báo hàm Nhap.

+ Hàm Nhap không có giá trị trả về.

+ Hàm Nhap có ba tham số đầu vào.

+ Tham số thứ nhất có kiểu dữ liệu string và là tham biến.

+ Tham số thứ hai có kiểu dữ liệu int và là tham biến.

+ Tham số thứ ba có kiểu dữ liệu int và là tham biến.

# Tham số và hàm

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.void Nhap(string&, int&, int&);
```

# Tham số và hàm

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.void Nhap(string&, int&, int&);
```

```
15.void XuLy(int, int, float&);
```

# Tham số và hàm

```
15. void XuLy(int, int, float&);
```

- Dòng lệnh 15 đọc là: Khai báo hàm **XuLy**.
  - + Hàm **XuLy** không có giá trị trả về.
  - + Hàm **XuLy** có ba tham số đầu vào.
  - + Tham số thứ nhất có kiểu dữ liệu **int** và là tham trị.
  - + Tham số thứ hai có kiểu dữ liệu **int** và là tham trị.
  - + Tham số thứ ba có kiểu dữ liệu **float** và là tham biến.

# Tham số và hàm

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.void Nhap(string&, int&, int&);
```

```
15.void XuLy(int, int, float&);
```

# Tham số và hàm

```
11.#include <iostream>
```

```
12.#include <string>
```

```
13.using namespace std;
```

```
14.void Nhap(string&, int&, int&);
```

```
15.void XuLy(int, int, float&);
```

```
16.void Xuat(string, int, int, float);
```

# Tham số và hàm

```
16. void Xuat(string, int, int, float);
```

- Dòng lệnh 16 đọc là: Khai báo hàm **Xuat**.
  - + Hàm **Xuat** không có giá trị trả về.
  - + Hàm **Xuat** có bốn tham số đầu vào.
  - + Tham số thứ nhất có kiểu dữ liệu **string** và là tham trị.
  - + Tham số thứ hai có kiểu dữ liệu **int** và là tham trị.
  - + Tham số thứ ba có kiểu dữ liệu **int** và là tham trị.
  - + Tham số thứ tư có kiểu dữ liệu **float** và là tham trị.



# Tham số và hàm

```
17.int main()  
18.{  
19.    string ht;  
20.    int t, v;  
21.    float tb;  
22.    Nhap(ht, t, v);  
  
25.    return 0;  
26.}
```

# Tham số và hàm

22.      Nhap(ht, t, v);

— Dòng lệnh 22 đọc là: Hàm **Nhap** được gọi thực hiện với ba đối số **ht**, **t** và **v**.

# Tham số và hàm

```
17.int main()  
18.{  
19.    string ht;  
20.    int t, v;  
21.    float tb;  
22.    Nhap(ht, t, v);  
  
25.    return 0;  
26.}
```

# Tham số và hàm

```
17.int main()  
18.{  
19.    string ht;  
20.    int t, v;  
21.    float tb;  
22.    Nhap(ht, t, v);  
23.    XuLy(t, v, tb);  
  
25.    return 0;  
26.}
```

# Tham số và hàm

23. XuLy(t, v, tb);

— Dòng lệnh 23 đọc là: Hàm XuLy được gọi thực hiện với ba đối số t, v và tb.

# Tham số và hàm

```
17.int main()  
18.{  
19.    string ht;  
20.    int t, v;  
21.    float tb;  
22.    Nhap(ht, t, v);  
23.    XuLy(t, v, tb);  
  
25.    return 0;  
26.}
```

# Tham số và hàm

```
17.int main()  
18.{  
19.    string ht;  
20.    int t, v;  
21.    float tb;  
22.    Nhap(ht, t, v);  
23.    XuLy(t, v, tb);  
24.    Xuat(ht, t, v, tb);  
25.    return 0;  
26.}
```



# Tham số và hàm

24.      `Xuat(ht, t, v, tb);`

— Dòng lệnh 24 đọc là: Hàm **Xuat** được gọi thực hiện với bốn đối số **ht**, **t**, **v** và **tb**.

# Tham số và hàm

```
11. void    Xuat(string    HoTen,int    Toan,int    Van,  
           float DiemTrungBinh)  
12. {  
13.     cout << "\n Ho ten:" << HoTen;  
14.     cout << "\n Toan: " << Toan;  
15.     cout << "\n Van: " << Van;  
16.     cout << "\n Diem Trung binh:" << DiemTrungBinh;  
17. }
```

# Tham số và hàm

```
11. void XuLy(int Toan, int Van, float& DiemTrungBinh)
12. {
13. |   DiemTrungBinh = (float)(Toan + Van) / 2;
14. }
```

# Tham số và hàm

```
11. void Nhap(string& HoTen, int& Toan, int& Van)
12. {
13.     cout << "Nhap ho ten:";
14.     getline(cin, HoTen);
15.     cout << "Nhap Toan:";
16.     cin >> Toan;
17.     cout << "Nhap Van:";
18.     cin >> Van;
19. }
```

## 7. KIỂU CẤU TRÚC

# Kiểu cấu trúc

- Khái niệm: Kiểu cấu trúc là một phương pháp tổng hợp các kiểu dữ liệu đơn (**simple datatype**), các kiểu dữ liệu có sẵn nhằm mô tả, biểu diễn một khái niệm (**concept**) hay một đối tượng (**object**) trong thế giới thực.



# Kiểu cấu trúc

## — Cú pháp:

```
1. struct KiểuDuLieu
2. {
3.     Thành phần 1;
4.     Thành phần 2;
5.     ...
6. };
7. typedef struct KiểuDuLieu KIEUDULIEU;
```



# Kiểu cấu trúc

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Kiểu cấu trúc

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct HocSinh
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
20.};
21.typedef struct HocSinh HOCSINH;
```

# Kiểu cấu trúc

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct HocSinh
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
20.};
21.typedef struct HocSinh HOCSINH;
```

— Dòng 14 đọc là: Bắt đầu khai báo kiểu dữ liệu cấu trúc HocSinh.

# Kiểu cấu trúc

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct HocSinh
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
20.};
21.typedef struct HocSinh HOCSINH;
```

— Dòng 16 đọc là: Thành phần thứ nhất của kiểu dữ liệu cấu trúc **HocSinh** có tên **HoTen** và có kiểu dữ liệu **string**.

# Kiểu cấu trúc

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct HocSinh
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
20.};
21.typedef struct HocSinh HOCSINH;
```

— Dòng 17 đọc là: Thành phần thứ hai của kiểu dữ liệu cấu trúc **HocSinh** có tên **Toan** và có kiểu dữ liệu **int**.

# Kiểu cấu trúc

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct HocSinh
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
20.};
21.typedef struct HocSinh HOCSINH;
```

— Dòng 18 đọc là: Thành phần thứ ba của kiểu dữ liệu cấu trúc **HocSinh** có tên **Van** và có kiểu dữ liệu **int**.



# Kiểu cấu trúc

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct HocSinh
15.{
16.    string HoTen;
17.    int Toan;
18.    int Van;
19.    float DiemTrungBinh;
20.};
21.typedef struct HocSinh HOCSINH;
```

— Dòng 19 đọc là: Thành phần thứ tư của kiểu dữ liệu cấu trúc **HocSinh** có tên **DiemTrungBinh** và có kiểu dữ liệu **float**.



# Kiểu cấu trúc

```
11. void Nhap(HOCSINH&);  
12. void XuLy(HOCSINH&);  
13. void Xuat(HOCSINH);  
14. int main()  
15. {  
16.     HOCSINH hs;  
17.     Nhap(hs);  
18.     XuLy(hs);  
19.     Xuat(hs);  
20.     return 0;  
21. }
```

# Kiểu cấu trúc

```
11. void Xuat(HOCSINH x)
12. {
13.     cout << "\n Ho ten:" << x.HoTen;
14.     cout << "\n Toan: " << x.Toan;
15.     cout << "\n Van: " << x.Van;
16.     cout << "\n Trung binh:" << x.DiemTrungBinh;
17. }
```

# Kiểu cấu trúc

```
11. void XuLy(HOCSINH& x)
12. {
13. |   x.DiemTrungBinh = (float)(x.Toan + x.Van) / 2;
14. }
```

# Kiểu cấu trúc

```
11. void Nhap(HOCSINH& x)
12. {
13.     cout << "Nhap ho ten:";
14.     getline(cin, x.HoTen);
15.     cout << "Nhap Toan:";
16.     cin >> x.Toan;
17.     cout << "Nhap Van:";
18.     cin >> x.Van;
19. }
```

## 8. ỨNG DỤNG 1

# Ứng dụng 1

—Ứng dụng 1: Viết chương trình nhập vào một phân số. Rút gọn phân số đó và xuất kết quả.

# Ứng dụng 1

```
11.#include <iostream>
12.#include <string>
13.using namespace std;
14.struct PhanSo
15.{
16.    int Tu;
17.    int Mau;
18.};
19.typedef struct PhanSo PHANSO;
```



# Ứng dụng 1

```
20.void Nhap(PHANSO&);  
21.void Xuat(PHANSO);  
22.void RutGon(PHANSO&);  
23.int main()  
24.{  
25.    PHANSO ps;  
26.    Nhap(ps);  
27.    RutGon(ps);  
28.    Xuat(ps);  
29.    return 0;  
30.}
```

# Ứng dụng 1

```
31. void Xuat(PHANSO x)
32. {
33.     cout << "\n Tu:" << x.Tu;
34.     cout << "\n Mau: " << x.Mau;
35. }
```

# Ứng dụng 1

```
36. void RutGon(PHANSO& x)
37. {
38.     int a = abs(x.Tu);
39.     int b = abs(x.Mau);
40.     while (a * b != 0)
41.     {
42.         if (a > b)
43.             a = a - b;
44.         else
45.             b = b - a;
46.     }
47.     x.Tu = x.Tu / (a + b);
48.     x.Mau = x.Mau / (a + b);
49. }
```

# Ứng dụng 1

```
50. void Nhap(PHANSO& x)
51. {
52.     cout << "Nhap tu:";
53.     cin >> x.Tu;
54.     cout << "Nhap mau:";
55.     cin >> x.Mau;
56. }
```

## 9. ỨNG DỤNG 2

## Ứng dụng 2

—Ứng dụng 2: Viết chương trình nhập vào tọa độ 2 điểm trong mặt phẳng Oxy. Tính khoảng cách giữa chúng và xuất kết quả

# Ứng dụng 2

```
11.#include <iostream>
12.#include <cmath>
13.using namespace std;
14.struct Diem
15.{
16.    float x;
17.    float y;
18.};
19.typedef struct Diem DIEM;
```



# Ứng dụng 2

20.void Nhap(DIEM&);

21.void Xuat(DIEM);

22.float KhoangCach(DIEM, DIEM);

# Ứng dụng 2

```
23.int main()  
24.{  
25.    DIEM A, B;  
26.    Nhap(A);  
27.    Nhap(B);  
28.    float kq = KhoangCach(A, B);  
29.    Xuat(A);  
30.    Xuat(B);  
31.    cout << "\n Ket Qua = " << kq;  
32.    return 0;  
33.}
```

## Ứng dụng 2

```
34. void Xuat(DIEM P)
35. {
36.     cout << "\n x = " << P.x;
37.     cout << "\n y = " << P.y;
38. }
39. float KhoangCach(DIEM P, DIEM Q)
40. {
41.     return sqrt((Q.x - P.x) * (Q.x - P.x) +
42.         (Q.y - P.y) * (Q.y - P.y));
43. }
```

# Ứng dụng 2

```
44. void Nhap(DIEM& P)
45. {
46.     cout << "Nhap x: ";
47.     cin >> P.x;
48.     cout << "Nhap y: ";
49.     cin >> P.y;
50. }
```

# 10. BÀI TẬP

# Bài tập

- Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.
- Bài 2: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.
- Bài 3: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và xuất kết quả.

# Bài tập

- Bài 4: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.
- Bài 5: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.
- Bài 6: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.



# Bài tập

- Bài 7: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.
- Bài 8: Viết chương trình nhập tọa độ 3 đỉnh A,B,C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.
- Bài 9: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.

**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**

# Chương 01

## CHƯƠNG 1 – ÔN TẬP – PYTHON

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

Target

# 1. MỤC TIÊU – PYTHON

# Mục tiêu

- Ôn lại các khái niệm, các kiến thức nhập môn lập trình.
- Biến toàn cục (global variable).
- Biến cục bộ (local variable).
- Hàm và biến toàn cục.
- Tham số (parameter) và hàm (function).
- Trừu tượng hóa dữ liệu (data abstract).

Problem

## 2. BÀI TOÁN – PYTHON

# Bài toán

—Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.



Global variable

## 3. BIẾN TOÀN CỤC – PYTHON

# Biến toàn cục

- Khái niệm: Biến toàn cục (**global variable**) là biến được khai báo bên ngoài tất cả các hàm và được hiểu bên trong tất cả các hàm.



- Tham khảo: [https://en.wikipedia.org/wiki/Global\\_variable](https://en.wikipedia.org/wiki/Global_variable)

# Biến toàn cục

—Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Biến toàn cục

```
1. HoTen = input("Nhap ho ten: ")
2. Toan = int(input("Nhap diem Toan: "))
3. Van = int(input("Nhap diem Van: "))
4. DiemTrungBinh = (Toan + Van) / 2
5. print("Ho ten:", HoTen)
6. print("Diem Toan:", Toan)
7. print("Diem Van:", Van)
8. print("Diem trung binh =", DiemTrungBinh)
```

# Biến toàn cục

- Khái niệm: Biến toàn cục (**global variable**) là biến được khai báo bên ngoài tất cả các hàm và được hiểu bên trong tất cả các hàm.



- Tham khảo: [https://en.wikipedia.org/wiki/Global\\_variable](https://en.wikipedia.org/wiki/Global_variable)

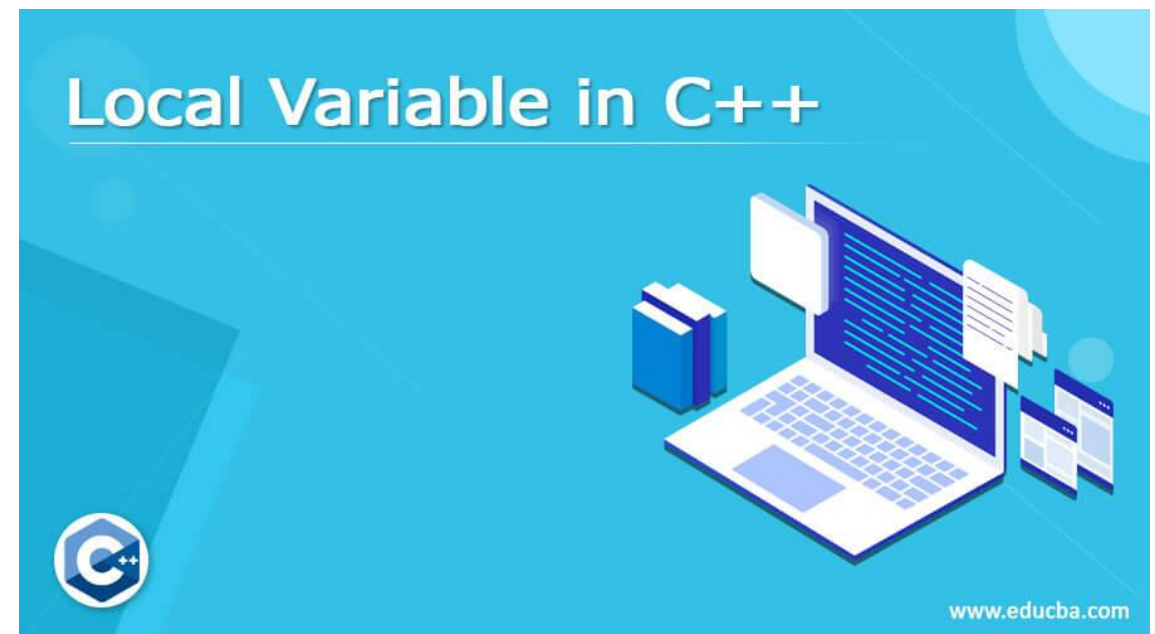
Local variable

## 4. BIẾN CỤC BỘ – PYTHON



# Biến cục bộ

- Khái niệm: Biến cục bộ (**local variable**) là biến được khai báo và được hiểu bên trong một phạm vi (scope) nào đó của chương trình, ra khỏi phạm vi này biến không còn được biết đến nữa vì không gian bộ nhớ cấp phát cho biến được tự động thu hồi.



- Tham khảo: [https://en.wikipedia.org/wiki/Local\\_variable](https://en.wikipedia.org/wiki/Local_variable)



# Biến cục bộ

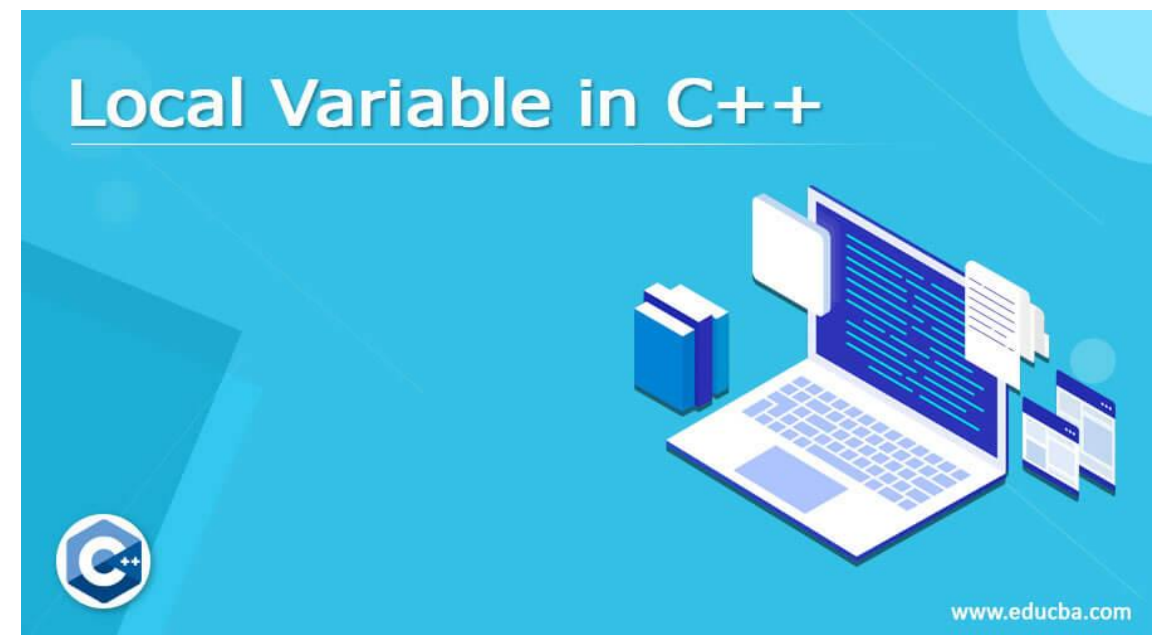
```
1. def main():
2.     HoTen = input("Nhap ho ten: ")
3.     Toan = int(input("Nhap diem Toan: "))
4.     Van = int(input("Nhap diem Van: "))
5.     DiemTrungBinh = (Toan + Van) / 2
6.     print("Ho ten:", HoTen)
7.     print("Diem Toan:", Toan)
8.     print("Diem Van:", Van)
9.     print("Diem trung binh =", DiemTrungBinh)
```

# Biến cục bộ

```
1. if __name__ == "__main__":  
2.     main()
```

# Biến cục bộ

- Khái niệm: Biến cục bộ (**local variable**) là biến được khai báo và được hiểu bên trong một phạm vi (scope) nào đó của chương trình, ra khỏi phạm vi này biến không còn được biết đến nữa vì không gian bộ nhớ cấp phát cho biến được tự động thu hồi.



- Tham khảo: [https://en.wikipedia.org/wiki/Local\\_variable](https://en.wikipedia.org/wiki/Local_variable)

Function and global variable

## **5. HÀM VÀ BIẾN TOÀN CỤC – PYTHON**

# Hàm và biến toàn cục

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Hàm và biến toàn cục

1. HoTen = None
2. Toan = None
3. Van = None
4. DiemTrungBinh = None

# Hàm và biến toàn cục

```
5. def Nhap():  
6.     global HoTen, Toan, Van, DiemTrungBinh  
7.     HoTen = input("Nhap ho ten: ")  
8.     Toan = int(input("Nhap diem Toan: "))  
9.     Van = int(input("Nhap diem Van: "))
```



# Hàm và biến toàn cục

```
10. def XuLy():  
11.     global HoTen, Toan, Van, DiemTrungBinh  
12.     DiemTrungBinh = (Toan + Van) / 2
```

# Hàm và biến toàn cục

```
1. def Xuat():  
2.     global HoTen, Toan, Van, DiemTrungBinh  
3.     print("Ho ten:", HoTen)  
4.     print("Diem Toan:", Toan)  
5.     print("Diem Van:", Van)  
6.     print("Diem trung binh =", DiemTrungBinh)
```

# Hàm và biến toàn cục

```
1. def main():  
2.     Nhap()  
3.     XuLy()  
4.     Xuat()  
  
5. if __name__ == "__main__":  
6.     main()
```

Parameter and function

## 6. THAM SỐ VÀ HÀM – PYTHON

# Tham số và hàm

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Tham số và hàm

```
1. def Nhap():  
2.     HoTen = input("Nhap ho ten: ")  
3.     Toan = int(input("Nhap Toan: "))  
4.     Van = int(input("Nhap Van: "))  
5.     return HoTen, Toan, Van
```

# Tham số và hàm

```
6. def XuLy(Toan, Van):  
7.     DiemTrungBinh = (Toan + Van) / 2  
8.     return DiemTrungBinh
```



# Tham số và hàm

```
9. def Xuat(HoTen, Toan, Van, DiemTrungBinh):  
10.     print("Ho ten:", HoTen)  
11.     print("Diem Toan:", Toan)  
12.     print("Diem Van:", Van)  
13.     print("Diem trung binh =", DiemTrungBinh)
```

# Tham số và hàm

```
14.def main():  
15.    ht, t, v = Nhap()  
16.    dtb = XuLy(t, v)  
17.    Xuat(ht, t, v, dtb)  
  
18.if __name__ == "__main__":  
19.    main()
```

## 7. KIỂU CẤU TRÚC – PYTHON

# Kiểu cấu trúc

- Bài toán: Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

# Kiểu cấu trúc

```
1. from collections import namedtuple
2. HocSinh = namedtuple('HS', 'HoTen, Toan, Van,
3.                                     DiemTrungBinh')
```

# Kiểu cấu trúc

```
4. def Nhap():  
5.     HoTen = input("Nhap ho ten: ")  
6.     Toan = int(input("Nhap Toan: "))  
7.     Van = int(input("Nhap Van: "))  
8.     x = HocSinh(HoTen, Toan, Van, 0)  
9.     return x
```

# Kiểu cấu trúc

```
10. def XuLy(x):  
11.     dtb = (x.Toan + x.Van) / 2  
12.     x = x._replace(DiemTrungBinh = dtb)  
13.     return x
```

# Kiểu cấu trúc

```
14. def Xuat(x):  
15.     print("Ho ten:", x.HoTen)  
16.     print("Diem Toan:", x.Toan)  
17.     print("Diem Van:", x.Van)  
18.     print("Diem trung binh =", x.DiemTrungBinh)
```



# Kiểu cấu trúc

```
19. def main():  
20.     hs = Nhap()  
21.     hs = XuLy(hs)  
22.     Xuat(hs)  
  
23. if __name__ == "__main__":  
24.     main()
```

## 8. ỨNG DỤNG 1 – PYTHON

# Ứng dụng 1

—Ứng dụng 1: Viết chương trình nhập vào một phân số. Rút gọn phân số đó và xuất kết quả.

# Ứng dụng 1

```
1. from collections import namedtuple
2. PhanSo = namedtuple('PS', 'Tu, Mau')

3. def Nhap():
4.     tutu = int(input("Nhap tu: "))
5.     maumau = int(input("Nhap mau: "))
6.     x = PhanSo(tutu, maumau)
7.     return x
```

# Ứng dụng 1

```
8. def Xuat(x):  
9.     print(x.Tu, '/', x.Mau)  
10. def ucln(a,b):  
11.     a = abs(a)  
12.     b = abs(b)  
13.     while (b > 0):  
14.         a = a % b  
15.         a, b = b, a  
16.     return a
```

# Ứng dụng 1

```
17. def RutGon(x):  
18.     uc = ucIn(x.Tu, x.Mau)  
19.     tutu = x.Tu // uc  
20.     maumau = x.Mau // uc  
21.     x = x._replace(Tu = tutu, Mau = maumau)  
22.     return x
```

# Ứng dụng 1

```
23. def main():
24.     ps = Nhap()
25.     print("Phan so truooc khi rut gon:", end = ' ')
26.     Xuat(ps)
27.     ps = RutGon(ps)
28.     print("Phan so sau khi rut gon:", end = ' ')
29.     Xuat(ps)
30. if __name__ == "__main__":
31.     main()
```



## 9. ỨNG DỤNG 2 – PYTHON

## Ứng dụng 2

—Ứng dụng 2: Viết chương trình nhập vào tọa độ 2 điểm trong mặt phẳng Oxy. Tính khoảng cách giữa chúng và xuất kết quả

# Ứng dụng 2

```
1. import math
2. from collections import namedtuple
3. Diem = namedtuple('D', 'x, y')

4. def Nhap():
5.     xx = float(input("Nhap hoành do: "))
6.     yy = float(input("Nhap tung do: "))
7.     P = Diem(xx, yy)
8.     return P
```

## Ứng dụng 2

```
9. def Xuat(P):  
10.     print('(' , P.x, ', ' , P.y, ')')  
  
11. def KhoangCach(P, Q):  
12.     return math.sqrt((P.x-Q.x)*(P.x-Q.x) +  
13.                      (P.y-Q.y)*(P.y-Q.y))
```

# Ứng dụng 2

```
14. def main():
15.     print("Nhap toa do diem A:")
16.     A = Nhap()
17.     print("Nhap toa do diem B:")
18.     B = Nhap()
19.     print("Diem A:", end = ' ')
20.     Xuat(A)
21.     print("Diem B:", end = ' ')
22.     Xuat(B)
23.     print("Khoang cach giua 2 diem la:", KhoangCach(A,B))
```

# Ứng dụng 2

```
24. if __name__ == "__main__":  
25.     main()
```

# 10. BÀI TẬP



# Bài tập

- Bài 1: Viết chương trình nhập vào một phân số. Hãy cho biết phân số đó là phân số âm hay dương hay bằng không.
- Bài 2: Viết chương trình nhập tọa độ hai điểm trong không gian. Tính khoảng cách giữa chúng và xuất kết quả.
- Bài 3: Viết chương trình nhập vào 2 phân số. Tìm phân số lớn nhất và xuất kết quả.

# Bài tập

- Bài 4: Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.
- Bài 5: Viết chương trình nhập vào 2 số phức. Tính tổng, hiệu, tích và xuất kết quả.
- Bài 6: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.

# Bài tập

- Bài 7: Viết chương trình nhập vào một ngày. Tìm ngày hôm qua và xuất kết quả.
- Bài 8: Viết chương trình nhập tọa độ 3 đỉnh A,B,C của 1 tam giác trong mặt phẳng Oxy. Tính chu vi, diện tích và tìm tọa độ trọng tâm.
- Bài 9: Viết chương trình nhập tọa tâm và bán kính của một đường tròn. Tính diện tích và chu vi của đường tròn.

**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**