



# Chương 1

## ÔN TẬP

1. ThS. Nguyễn Hữu Lợi
2. ThS. Nguyễn Văn Toàn
3. TS. Nguyễn Duy Khánh
4. 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.
- 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)

# 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.



# 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 1;  
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 Trung binh:" << DiemTrungBinh;
31.     return 1;
32. }
```

# 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)



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)



# 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.

# 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 1;
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 Trung binh:" << DiemTrungBinh;
31.     return 1;
32. }
```



Function and global variable

## **5. HÀM VÀ BIẾN TOÀN CỤC**

# Hàm và biến toàn cục

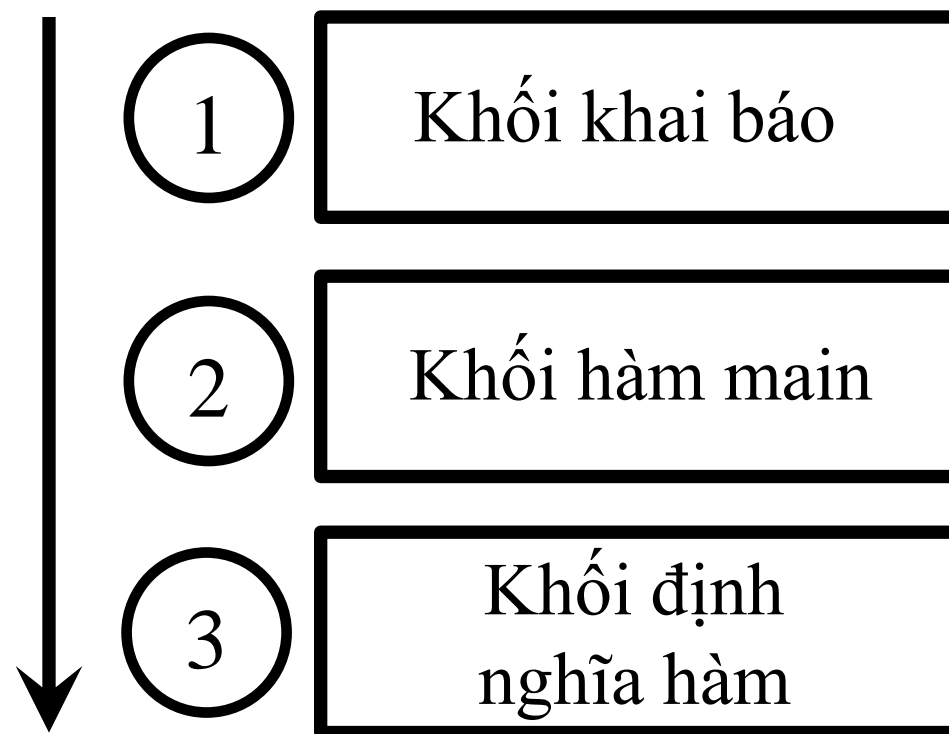


- Kiến trúc chương trình C.
- Khởi khai báo.
- Khởi hàm main.
- Khởi định nghĩa hàm.

# 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 1;
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. }
```



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.
- 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



```
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



```
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 1;  
26.}
```

# 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 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 trong thế giới thực.



# Kiểu cấu trúc



## — Cú pháp:

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

# 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. 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 1;  
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

# 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 1;  
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. }
```

## 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 1;
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. }
```



## 9. BÀI TẬP

## 9. 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 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.

## 9. Bài tập



- Bài 6: Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.
- 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**