STRING TRONG C++

1. **Khái niệm**

* Kiểu dữ liệu **string** dùng để lưu trữ các chuỗi kí tự, đoạn văn.
* Một chuỗi kí tự được đặt trong cặp dấu nháy kép “” .

Ví dụ:

string s = "hello";

string s1 = "Xin chao moi nguoi.";

string s2 = "Lap trinh C++";

* Include thư viện <string> trước khi sử dụng.

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s;

s = "Lap trinh C++";

cout << s;

}

1. **Nhập xuất chuỗi**
2. Nhập chuỗi

* Có thể sử dụng lệnh cin để cho phép nhập chuỗi từ bàn phím.

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string ten;

cout << "Nhap ten cua ban: ";

cin >> ten;

cout << "Chao, " << ten;

}

Tuy nhiên, lệnh cin xem các kí tự khoảng trắng (dấu cách, tab) là kí tự kết thúc nên chỉ có thể gõ được 1 từ đơn duy nhất. Hãy chạy thử ví dụ trên để kiểm tra thử .

(Gõ “Nguyen Van A” vào thì kết quả chương trình chỉ xuất ra “Nguyen” vì kí tự tiếp theo là khoảng trắng, lệnh cin sẽ không đọc tiếp phần chuỗi từ dấu khoảng trắng).

* Thông thường khi làm việc với string, để đọc 1 chuỗi từ bàn phím ta sử dụng hàm getline() để đọc 1 dòng dữ liệu.

Cú pháp:

getline(cin, tên\_biến);

Hàm getline nhận tham số thứ nhất là std::cin, tham số thứ 2 là một biến kiểu string mà bạn muốn cho phép người dùng nhập từ bàn phím.

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string ten;

cout << "Nhap ten cua ban: ";

getline(cin, ten);

cout << "Chao, " << ten;

}

* Lưu ý khi sử dụng hàm getline():

Xét ví dụ sau (nhập vào số nguyên n, sau đó nhập vào 1 chuỗi).

#include<iostream>

#include<string>

using namespace std;

int main()

{

int n;

cout << "Nhap n: ";

cin >> n;

string ten;

cout << "Nhap ten cua ban: ";

getline(cin, ten);

cout << "Chao, " << ten;

}

+ Chạy thử đoạn chương trình trên, nhập giá trị của n và nhấn enter. Ta thấy sau khi nhập n, chương trình không dừng lại để cho phép nhập giá trị cho chuỗi “ten” từ bàn phím, do sau khi nhập n và nhấn enter thì giá trị số nguyên sẽ được gán cho n, và dấu enter vẫn còn ở đó. Đến lệnh đọc một chuỗi từ bàn phím, chương trình đọc tiếp vào và gặp dấu enter nên sẽ kết thúc việc đọc chuỗi, biến “ten” sẽ mang giá trị rỗng.

+ Để khắc phục vấn đề trên, ta dùng lệnh cin.ignore(); trước khi nhập chuỗi để khử dấu enter.

#include<iostream>

#include<string>

using namespace std;

int main()

{

int n;

cout << "Nhap n: ";

cin >> n;

string ten;

cin.ignore();

cout << "Nhap ten cua ban: ";

getline(cin, ten);

cout << "Chao, " << ten;

}

1. Xuất chuỗi

Sử dụng lệnh cout để xuất chuỗi, tương tự các kiểu dữ liệu khác.

1. **Một số thao tác trên chuỗi**
2. **Truy xuất phần tử của chuỗi**

Tương tự với mảng, ta có thể truy xuất từng phần tử của chuỗi với cú pháp tương tự, với mỗi phần tử của chuỗi là một kí tự. Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao cac ban";

cout << "s[2] = " << s[2] << endl;

char KiThuThu0 = s[0];

cout << "s[0] = " << KiThuThu0 << endl;

s[3] = '.'; // thay doi gia tri ki tu thu 3

cout << s << endl; // xuat s sau khi thay doi

}

Kết quả chạy chương trình:



1. **Lấy chiều dài chuỗi**

Gọi đến phương thức (hàm) length() của để lấy chiều dài chuỗi (trong phần này sử dụng các cú pháp lập trình hướng đối tượng, các bạn chú ý cú pháp)

+ Cú pháp:

s.length()

Với s là chuỗi cần lấy chiều dài

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao cac ban";

cout << "Chieu dai chuoi = " << s.length();

}

Kết quả chạy chương trình:



1. **Nối chuỗi**

* Sử dụng toán tử cộng (+) để nối hai chuỗi với nhau.

Ví dụ 1:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s1 = "Xin chao ";

string s2 = "the gioi";

string s3 = s1 + s2;

cout << "Ket qua noi chuoi: " << s3;

}

Kết quả chạy chương trình:



Ví dụ 2:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s1 = "10";

string s2 = "20";

string s3 = s1 + s2;

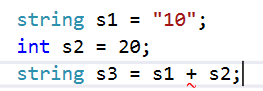
cout << "Ket qua noi chuoi: " << s3;

}

Kết quả chạy chương trình:



* Lưu ý: có thể nối chuỗi với kí tự, nhưng không thể nối chuỗi với các giá trị số nguyên / số thực. Đoạn chương trình sau sẽ bị lỗi:



1. **Chèn và xóa chuỗi**

* Sử dụng phương thức insert() để chèn một chuỗi vào giữa chuỗi hiện hành.

Cú pháp:

s.insert(x, s1);

Lệnh này sẽ chèn chuỗi s1 vào vị trí x của chuỗi s.

Ví dụ 1: cho chuỗi s = “xin ao”, chèn chuỗi s1 = “ch” vào vị trí thứ 4 trong chuỗi s:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "xin ao";

string s1 = "ch";

s.insert(4, s1);

cout << s;

}

* Sử dụng phương thức erase() để xóa một phần của chuỗi.

+ Cú pháp:

s.erase(x, n);

Lệnh này sẽ xóa n kí tự của chuỗi s, bắt đầu từ vị trí x.

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao cac ban";

s.erase(8, 4);

cout << s;

}

Kết quả chạy chương trình:



1. **So sánh chuỗi**

* Gọi phương thức compare() để so sánh hai chuỗi với nhau.

+ Cú pháp

s.compare(s1)

Lệnh này sẽ so sánh chuỗi s với chuỗi s1 và trả về một số nguyên có ý nghĩa như sau:

|  |  |
| --- | --- |
| **Giá trị trả về** | **Ý nghĩa** |
| = 0 | Hai chuỗi s và s1 bằng nhau |
| < 0 | Chuỗi s1 bé hơn s2 |
| > 0 | Chuỗi s1 lớn hơn s2 |

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s1 = "xin chao";

string s2 = "xin chao";

string s3 = "xin";

string s4 = "cac ban";

cout << s1.compare(s2) << endl;

cout << s1.compare(s3) << endl;

cout << s4.compare(s1) << endl;

}

Kết quả chạy chương trình:



1. **Lấy chuỗi con**

* Sử dụng phương thức substr() để lấy một phần của một chuỗi.

+ Cú pháp

s.substr(x, n)

Lệnh này sẽ trả về một chuỗi là chuỗi con của s, lấy từ vị trí x và lấy n kí tự.

Ví dụ: cho chuỗi s = “Xin chao moi nguoi”, lấy ra chuỗi con của s từ vị trí x = 4 và lấy 8 kí tự:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao moi nguoi.";

string s1 = s.substr(4, 8);

cout << s1;

}

Kết quả chạy chương trình:



1. **Tìm vị trí xuất hiện của một chuỗi con**

* Sử dụng phương thức find() để lấy vị trí xuất hiện đầu tiên của một chuỗi con trong chuỗi hiện hành.

+ Cú pháp:

s.find(s1)

Lệnh này trả về vị trí xuất hiện đầu tiên của chuỗi s1 trong chuỗi s.

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao cac ban.";

cout << s.find("chao");

}

Kết quả chạy chương trình:



* Tương tự, phương thức rfind() trả về vị trí xuất hiện cuối cùng của một chuỗi con trong chuỗi hiện hành. Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao cac ban, chao the gioi";

cout << s.rfind("chao");

}

Kết quả chạy chương trình:



1. **Thay thế**

* Sử dụng phương thức replace() để thay thế một phần của chuỗi bằng một chuỗi khác.

+ Cú pháp:

s.replace(pos, len, s1)

Lệnh này sẽ thay thế chuỗi s, bắt đầu từ vị trí pos và lấy len kí tự, thay bằng chuỗi s1.

Ví dụ:

#include<iostream>

#include<string>

using namespace std;

int main()

{

string s = "Xin chao cac ban, chao the gioi";

s.replace(0, 8, "Chao");

cout << s;

}

Kết quả chạy chương trình:

