

Bài 2.2 MỘT SỐ KIỂU DỮ LIỆU CƠ SỞ (BÀI 4)

Các Ngôn Ngữ Lập Trình thường cung cấp các kiểu dữ liệu cơ sở cùng với các phép toán để người lập trình có thể thực hiện các tính toán để giải quyết bài toán. Một số kiểu dữ liệu cơ sở dùng trong C++:

Kiểu nguyên

int chiếm **4 byte**, biểu diễn được 1 giá trị trong phạm vi - 2^{4*8-1} đến $2^{4*8-1}-1$

Kiểu thực

float chiếm **4 bytes**, biểu diễn được 1 giá trị trong phạm vi - $3.4*10^{38}$ đến $3.4*10^{38}$

double chiếm **8 bytes**, biểu diễn được 1 giá trị trong phạm vi - $1.79*10^{308}$ đến $1.79*10^{308}$

Kiểu ký tự

char chiếm **1 byte**, biểu diễn được 1 trong 256 kí tự trong **bảng mã ASCII**

Kiểu logic

bool chiếm **1 byte**, biểu diễn được 1 trong 2 giá trị **0 và 1**

.....
.....
.....
.....
.....

Bài 2.3 KHAI BÁO BIẾN (BÀI 5)***Khai báo không khởi tạo***

<.....> <.....>;

Danh sách biến: là một hoặc nhiều tên biến cách nhau bởi dấu phẩy*Kiểu dữ liệu:* Ở các chương trình đơn giản thường là kiểu dữ liệu chuẩn.**Chú ý:**

- Đặt tên biến sao cho gợi nhớ đến ý nghĩa của biến đó.
- Không nên đặt tên biến quá ngắn hay quá dài.
- Phạm vi giá trị của biến khi khai báo biến.

VD1:

```
float x ;    // khai báo biến x có kiểu thực
int i, j ;   // khai báo 2 biến i, j có kiểu nguyên
..... // khai báo biến a và biến b có kiểu thực 8 bytes
..... // khai báo biến Ngay chứa số ngày trong năm
..... // khai báo biến có kiểu logic
.....
.....
.....
```

Khai báo có khởi tạo

<Kiểu dữ liệu> <Tên biến 1> = <Giá trị 1>[, <Tên biến 2> = <Giá trị 2>...];

VD2:

```
int x = 2;    // khai báo biến x kiểu nguyên có giá trị 2
int a = 5, b = 10; // khai báo 2 biến a, b có kiểu nguyên a có giá trị 5, b có giá trị 10
..... // khai báo biến L kiểu logic có giá trị là sai
..... // khai báo biến NhiệtDo có giá trị 26.7
.....
.....
.....
.....
.....
.....
```

Bài 2.4 PHÉP TOÁN, BIỂU THỨC, CÂU LỆNH GÁN (BÀI 6)

Để, cũng như các ngôn ngữ lập trình khác, C++ xác định và sử dụng một số khái niệm cơ bản: phép toán, biểu thức, gán giá trị cho biến.

1.1.1 Phép toán

Phép toán	Trong toán học	Trong C++
Các phép toán số học với số nguyên	+ (cộng), - (trừ), × (nhân), div (chia lấy phần nguyên), mod (lấy phần dư)	
	Tăng thêm 1, giảm bớt 1	
Các phép toán số học với số thực	+ (cộng), - (trừ), × (nhân), : (chia)	
Các phép toán quan hệ	< (nhỏ hơn), ≤ (nhỏ hơn hoặc bằng), > (lớn hơn), ≥ (lớn hơn hoặc bằng), = (bằng), ≠ (khác)	
Các phép toán logic	¬ (phủ định), ∨ (hoặc), ∧ (và)	

Chú ý:

- Kết quả của các phép toán quan hệ cho giá trị
- Một trong những ứng dụng của phép toán logic là để

Trong lập trình, biểu thức số học là mộthoặchoặcliên kết với nhau bởi **một số hữu hạn phép toán số học**, các dấu ngoặc tròn.

VD: - Biểu thức $(a * 3 / b) > 5$ không là biểu thức số học.

- Biểu thứclà biểu thức số học

Biểu thức trong Toán học	Biểu thức trong C++
$3a+6b^2$	
$\frac{2xy}{z}$	
$\frac{x+y}{x-\frac{1}{2}} - \frac{x-z}{xy}$	

1.1.2 Hàm số học chuẩn:

Các chương trình trong thư viện dùng để tính giá trị những hàm toán học thường dùng gọi là các hàm số học chuẩn.

Bản thân hàm chuẩn cũng được coi là một biểu thức số học và nó có thể tham gia vào biểu

thức số học như một toán hạng (giống như biến và hằng). Kết quả của hàm có thể là nguyên hoặc thực hay phụ thuộc vào kiểu của đối số.

MỘT SỐ HÀM SỐ HỌC CHUẨN THƯỜNG DÙNG

Hàm	Biểu diễn Toán học	Biểu diễn trong C++	Kiểu đối số	Kiểu kết quả
Lũy thừa	x^y	Pow(x,y)	Thực hoặc nguyên	Theo kiểu của đối số
Căn bậc hai	$\sqrt{x}$	sqrt(x)	Thực hoặc nguyên	Thực
Giá trị tuyệt đối	$ x $	abs(x)	Thực hoặc nguyên	Theo kiểu của đối số
Sin	$\sin x$	sin(x)	Thực hoặc nguyên	Thực
Cos	$\cos x$	cos(x)	Thực hoặc nguyên	Thực

VD:

Biểu thức trong Toán học	Biểu thức trong C++
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	
$\sin 3x = 3\sin x - 4\sin^3 x$	

1.1.3 Biểu thức quan hệ:

Hai biểu thức liên kết với nhau bởi phép toán quan hệ cho ta một biểu thức quan hệ.

Biểu thức quan hệ có dạng:

<biểu thức 1> <.....> <biểu thức 2>

Biểu thức 1 và *biểu thức 2* cùng là hoặc cùng là biểu thức

Biểu thức quan hệ được thực hiện theo trình tự:

- Tính giá trị các biểu thức.
- Thực hiện phép toán quan hệ.

Kết quả của biểu thức quan hệ là giá trị: 1 (đúng) hoặc 0 (sai).

VD1: Theo cú pháp, biểu thức quan hệ:

- $(x < 5+y)$ có *Biểu thức 1* và *biểu thức 2* cùng là:
- (“Truong” == “Marie Curie”) có *Biểu thức 1* và *biểu thức 2* cùng là:

1.1.4 Biểu thức logic:

Biểu thức *logic đơn giản* là biến logic hoặc hằng logic.

Biểu thức logic là các biểu thức logic đơn giản, các biểu thức quan hệ liên kết với nhau bởi phép toán logic, các biểu thức quan hệ thường được đặt trong cặp ngoặc (và).

Giá trị của biểu thức logic là: 1 (true) hoặc 0 (false).

VD1: $!(a+b>c)$ tương đương biểu thức quan hệ

VD2: Để thể hiện điều kiện $-1 \leq \sin(x) \leq 1$, trong C++ cần phải viết lại là:

.....

1.1.5 Câu lệnh gán

Cú pháp câu lệnh gán trong C++

$\langle \dots \rangle := \langle \dots \rangle ;$

VD:

Dem = 1;

x = -b/2/a;

Ketqua = 'A';

.....
.....
.....
.....
.....

Bài 2.5 CÁC THỦ TỤC CHUẨN VÀO / RA ĐƠN GIẢN (BÀI 7)

Các chương trình đưa dữ liệu vào và ra được gọi là các thủ tục chuẩn vào/ra đơn giản.

1.1.6 Nhập dữ liệu từ bàn phím:

Dùng đối tượng cin và toán tử >> như sau:

cin >> var1 [>> var2 >> var3 ...];

Trong đó: var1, var2, ... là các biến hợp lệ

VD1: Nhập dữ liệu cho biến C

cin >> C;

VD2: Nhập dữ liệu cho 2 biến D và R

cin >> D >> R;

1.1.7 Đưa dữ liệu ra màn hình:

Dùng đối tượng cout và toán tử << như sau:

cout << BT1 [<< BT2 << BT3 . . .];

Trong đó: BT1, BT2, ... là các biểu thức.

VD1: Đưa giá trị biến C ra màn hình

.....

VD2: Đưa giá trị chuỗi, biểu thức $D \cdot R$, ký tự xuống dòng ra màn hình

.....

1.1.8 Ví dụ chương trình đơn giản có sử dụng các thủ tục vào/ra

VD1: Sau đây là chương trình nhập cạnh của hình vuông, sau đó xuất ra màn hình diện tích của hình vuông đó.

1	#include<iostream>
2	using namespace std;
3	int main()
4	{
5	float E;
6	cout << “Hay nhap do dai canh hinh vuong: ”;
7	cin >> E;
8	cout << “Dien tich hinh vuong la: ” << E*E;
9	return 0;
10	}

VD2: Sau đây là chương trình nhập chiều dài và chiều rộng HCN, sau đó xuất ra màn hình diện tích và chu vi của HCN đó.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....