8h15 điểm danh

3 nghỉ có phép = 1 nghỉ không phép

60h = 15b

Nghỉ > 3b: thi lần 2

Trượt lần 1: thi lần 2

Nghỉ > 4.5b (4 nghỉ + 1 đi muộn hoặc 1 có phép): học lại

Giải lao 15p lúc 10h

Nguyên Lý lập trình (Lập trình C++)

Cài Dev C++

Đuôi file .cpp

Biến (Variable): đại lượng được dùng để lưu trữ dữ liệu, dữ liệu có thể được trả về từ trong phần mềm hoặc được nhập từ bàn phím.

Đặc điểm:

* Biến là duy nhất
* Biến lúc này thì lưu dữ liệu này, lúc khác lại lưu dữ liệu khác
* Biến luôn luôn có 1 kiểu dữ liệu

Lưu ý: 1 biến muốn có thể sử dụng thì cần khai báo

Cấu trúc khai báo biến: biến chỉ được khai báo 1 lần

* Khai báo biến và không gán giá trị cho biến

Kieu\_du\_lieu\_cua\_bien ten\_bien;

* Khai báo biến và gán luôn giá trị cho biến

Kieu\_du\_lieu\_cua\_bien ten\_bien = gia\_tri\_cua\_bien;

Quy tắc đặt tên biến:

* Tên biến là duy nhất, không trùng nhau
* Tên biến viết liền không dấu hoặc viết bằng Tiếng Anh
* Tên biến chỉ chứa chữ cái (chữ hoa, chữ thường), số, \_
* Tên biến chỉ có thể bắt đầu bằng chữ cái
* Tên biến không trùng từ khóa
* Tên biến phân biệt hoa thường. Ví dụ: biến soNguyen khác biến SoNguyen

Kiểu dữ liệu (data type): kiểu giá trị của biến có thể lưu trữ được (quy định biến lưu được loại giá trị nào)

1 số kiểu dữ liệu cơ bản:

* Số nguyên: int (viết tắt của integer)
* Số thực: float, double
* Ký tự: char (viết tắt character). Lưu ý: giá trị của biến có kiểu dữ liệu char luôn nằm giữa 2 dấu ‘’
* Chuỗi: string
* True hoặc false (kiểu dữ liệu 2 trạng thái): bool (viết tắt của boolean)
* Kiểu dữ liệu không trả về giá trị: void

Ép kiểu: chuyển đổi dữ liệu của biến từ kiểu dữ liệu này sang kiểu dữ liệu khác

Cấu trúc: (kieu\_du\_lieu\_duoc\_chuyen\_sang)ten\_bien

Ví dụ: biến a có kiểu là int, chuyển biến a sang kiểu dữ liệu float: (float)a

|  |  |  |
| --- | --- | --- |
|  | int | float |
| int | int + int => int  int – int => int  int \* int => int  int / int => int | int + float => float  int – float => float  int \* float => float  int / float => float |
| float | float + int => float  float – int => float  float \* int => floar  float / int => float | float + float => float  float – float => float  float \* float => float  float / float => float |

Hằng số (constant): hằng số đại lượng không thay đổi trong suốt quá trình chương trình chạy

Khai báo hằng:

const kieu\_du\_lieu\_cua\_hang ten\_hang = gia\_tri;

Lưu ý: quy tắc đặt tên hằng số giống quy tắc đặt tên biến

Từ khóa (Keyword): là những từ có sẵn trong ngôn ngữ lập trình, mỗi từ có 1 chức năng riêng, không thể thay đổi chức năng của từ khóa

Ghi chú (comment): Những dòng ghi chú và không được chạy khi chạy chương trình

Có 2 kiểu comment:

* Comment trên 1 dòng: bắt đầu //
* Comment trên nhiều dòng: bắt đầu /\* kết thúc \*/

Nhập xuất dữ liệu

Nhập dữ liệu từ bàn phím: cin >> ten\_bien;

Hiển thị dữ liệu ra ngoài màn hình: cout << du\_lieu\_can\_hien\_thi;

Nếu hiển thị 1 chuỗi thì chuỗi đó phải nằm trong cặp “ ”

Toán tử (Operator): ký hiệu thể hiện các phép toán

1 số loại toán tử

* **Toán tử số học:** +, -, \*, /, %
  + Toán tử %: chia lấy dư. Ví dụ: 5 chia 3 dư 2 => 5 % 3 = 2
* **Toán tử quan hệ (toán tử quan hệ luôn trả về True hoặc False):** >, <, >=, <=, ==, !=
  + Toán tử ==: so sánh 2 giá trị có bằng nhau hay không. Ví dụ: so sánh biến a có bằng 3 hay không: a == 3
  + Toán !=: so sánh 2 giá trị có khác nhau hay không. Ví dụ: so sánh biến a có khác 5 hay không: a != 5
* **Toán tử logic: && (AND), || (OR), ! (NOT)**
  + Toán tử &&: Trả về true khi tất cả các biểu thức trả về True, chỉ cần 1 biểu thức trả về False => trả về False
    - 5 > 3 => true, 5 != 7 => true, (5 > 3) && (5 != 7) => true
    - 5 > 3 => true, 5 == 7 => false, (5 > 3) && (5 == 7) => false
  + Toán tử ||: Trả về False khi tất cả các biểu thức False, chỉ cần có 1 biểu thức True => trả về True
    - 5 < 3 => false, 5 == 7 => false, (5 < 3) || ( 5 == 7) => false
    - 5 < 3 => false, 5 != 7 => true, (5 < 3) || (5 != 7) => true
  + Toán tử !: chuyển true => false và ngược lại
    - 5 > 3 => true, !(5 > 3) => false
    - 5 == 7 => false, !(5 == 7) => true
* **Toán tử gán:** =, +=, -=, \*=, /=, %=
  + Toán tử +=
    - a += 5: a = a + 5: Lấy giá trị của a cộng thêm 5, sau đó gán giá trị vừa nhận được vào a
  + Toán tử -=
    - a -= 5: a = a - 5: Lấy giá trị của a trừ đi 5, sau đó gán giá trị vừa nhận được vào a
  + Toán tử \*=
    - a \*= 5: a = a \* 5: Lấy giá trị của a nhân thêm 5, sau đó gán giá trị vừa nhận được vào a
  + Toán tử /=
    - a /= 5: a = a / 5: Lấy giá trị của a chia cho 5, sau đó gán giá trị vừa nhận được vào a
  + Toán tử %=
    - a %= 5: a = a % 5: Lấy giá trị của a chia lấy dư cho 5, sau đó gán giá trị vừa nhận được vào a
* **Toán tử bit:** ^, &. |
* **Toán tử khác:** ? :
* **++, --**
  + **++:** a++ => a += 1 => a = a + 1
    - a++ (hậu tố): gán trước, cộng sau
      * a = 5, b = a++
        + a = 6
        + b = 5
    - ++a (tiền tố): cộng trước, gán sau
      * a = 5, b = ++a
        + a = 6
        + b = 6
  + **-**-: a-- => a -= 1 => a = a – 1
    - a-- (hậu tố): gán trước, trừ sau
      * a = 5, b = a --
        + a = 4
        + b = 5
    - --a (tiền tố): trừ trước, gán sau
      * a = 5, b = --a
        + a = 4
        + b = 4
* **Độ ưu tiên của toán tử**

1. Toán tử nằm trong ()
2. Toán tử số học
3. Toán tử quan hệ
4. Toán tử logic
5. Toán tử gán
6. Thực hiện từ trái sang phải

* Ví dụ: a = (5 \* 3 / 9) == 6 && 6 % 2 != 1
  + Thứ tự thực hiện:
    - 5 \* 3
    - / 9
    - == 6
    - 6 % 2
    - != 1
    - &&
    - a =