**Môn: Nhập môn lập trình**

**Bài tập LT tuần 2**

***\* Phần trắc nghiệm cơ bản:***

**1.** Biến là gì?

A. Nơi chứa số trong quá trình thực thi chương trình.

B. Nơi chứa chữ trong quá trình thực thi chương trình.

C. Nơi chứa dữ liệu trong quá trình thực thi chương trình.

D. Nơi chứa chữ và số trong quá trình thực thi chương trình.

**2.** Miền giá trị kiểu long?

A. –2.157.483.348 … +2.157.483.347

B. –2.147.213.649 … +2.147.123.649

C. –2.117.483.649 … +2.197.483.649

D. –2.147.483.648 … +2.147.483.647

**3.** Tên biến nào dưới đây được viết đúng theo quy tắc đặt tên của ngôn ngữ lập trình  C?

A. diem toan

B. 3diemly

C. \_diemhoa

D.–diemsinh

**4.** Biến toàn cục là?

A. Khai báo cho tất cả các hàm, trừ main.

B. Khai báo cho tất cả các hàm, kể cả main.

C. Được khai báo trong hàm main.

D. Được khai báo ngoài hàm main.

**5.** Cho biết giá trị của biểu thức 23 < 3:

A. 0

B. 1

C. 2

D. Không câu nào đúng.

***\*Phần câu hỏi tự luận:***

**1.** C++ khác C ở điểm nào?

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Ngôn ngữ lập trình C** | **Ngôn ngữ lập trình C++** |
|  | Lập trình hướng thủ tục | **[Lập trình hướng đối tượng](https://topviecit.vn/blog/lap-trinh-huong-doi-tuong/" \t "_blank)** |
| Cách tiếp cận | Tiếp cận theo hướng top – down: Chia nhỏ vấn đề lớn thành những thành phần nhỏ hơn. | Tiếp cận theo hướng bottom – up: Tập trung giải quyết vấn đề nhỏ hơn ở mức độ cơ bản, sau đó mới tích hợp thành một giải pháp toàn diện, hoàn chỉnh. |
| Đuôi mở rộng | Đuôi mở rộng là *.c* | Đuôi mở rộng là *.cpp* |
| Tổ chức chương trình | Chia chương trình lớn thành nhỏ, được gọi là các function (hàm) | Chia chương trình lớn thành các Class (lớp) và Object (đối tượng). |
| Inline function | Không hỗ trợ | Có hỗ trợ |
| Nhập xuất chuẩn | Sử dụng *scanf* và *printf* (trong *stdio.h*) | Sử dụng *cin* và*cout* trong *std* |
| Con trỏ | Hỗ trợ tham trị, con trỏ | Hỗ trợ tham trị, tham chiếu, con trỏ |
| Quản lý ngoại lệ | Không có quản lý ngoại tệ. Nhưng vẫn có một số giải pháp khác. | Có hỗ trợ, nhưng bạn cũng có thể sử dụng try catch để tìm lỗi. |
| Nạp chồng hàm | Không hỗ trợ | Có hỗ trợ |
| Hàm | Không cho phép giá trị mặc định của tham số | Cho phép giá trị mặc định của tham số. |
| Quan hệ | C không chạy được code C++ | C++ chạy được code của C |
| Namespace | Không có | Có |
| Quản lý bộ nhớ | *malloc()* và *calloc ()* cho phép cấp phát động | Toán tử *new* cho cấp phát động |
| Lập trình giao diện | Sử dụng công cụ GTK | Qt hỗ trợ lập trình GUI |
| Hàm ảo | Không tồn tại | Có tồn tại |
| Hiệu suất | Hiệu suất của C trong một số trường hợp sẽ nhanh hơn | Mã C++ được viết tốt hơn có thể sẽ hoạt động tốt hơn. |

**2.** Các chú thích (comment) được dùng để làm gì?

Trong lập trình máy tính, comment là phần giải thích, chú thích hay chú giải mà lập trình viên có thể đọc được trong mã nguồn của một chương trình máy tính. Chúng được thêm vào với mục đích là làm cho mã nguồn dễ hiểu hơn cho con người, và nói chung trình biên dịch và trình thông dịch đều bỏ qua chúng.

**3.** Khối lệnh là gì?

Trong C++, một khối lệnh là tập hợp những câu lệnh được đặt trong cặp dấu ngoặc nhọn **{** và **}**.

**4.** Tại sao phải luôn khởi tạo giá trị cho một biến mới?

Khởi tạo biến không chỉ đảm bảo rằng biến chứa giá trị có ý nghĩa ban đầu mà còn giúp tránh các giá trị ngẫu nhiên không mong muốn. Nó còn giúp trong việc debug và kiểm tra lỗi, vì bạn biết chính xác giá trị nào đang được sử dụng từ đầu.

**5.** Giải thích sự khác nhau của các câu lệnh sau:

cout << "Seven divided by three is " << 7 / 3 << endl;

7 và 3 là số nguyên

cout << "Seven divided by three is " << 7.0 / 3 << endl;

7 là số thực, 3 là số nguyên

cout << "Seven divided by three is " << 7.0 / 3.0 << endl;

7 và 3 đều là số thực

**6.** Kiểu dữ liệu bool xuất phát từ đâu?

Từ nhà toán học George Boole chỉ ra đúng sai thường được kết hợp với câu điều kiện

***\* Phần lập trình:***

*Nguồn tham khảo: sách Programming Principles and Pratice using C++*

**7.** Write a program in C++ that converts from miles to kilometers. Your program  should have a reasonable prompt for the user to enter a number of miles. Hint:  There are 1.609 kilometers to the mile.

**8.** Write a program that doesn’t do anything, but declares a number of variables with  legal and illegal names (such as **int double = 0;**), so that you can see how the  compiler reacts.

**9.** Write a program that prompts the user to enter two integer values. Store these  values in **int** variables named **val1** and **val2**. Write your program to determine the  sum, difference, product, and ratio of these values and report them to the user.

Modify the program above to ask the user to enter floating-point values and store  them in **double** variables. Compare the outputs of the two programs for some  inputs of your choice.

Are the results the same? Should they be? What’s the difference?

**10.** Write a program that prompts the user to enter some number of pennies (1-cent  coins), nickels (5-cent coins), dimes (10-cent coins), quarters (25-cent coins), half  dollars (50-cent coins), and one-dollar coins (100-cent coins). Query the user  separately for the number of each size coin,

e.g., “How many pennies do you have?” Then your program should print out  something like this:

**You have 23 pennies.**

**You have 17 nickels.**

**You have 14 dimes.**

**You have 7 quarters.**

**You have 3 half dollars.**

**The value of all of your coins is 573 cents.**

Make some improvements: if only one of a coin is reported, make the output

grammatically correct, e.g., **14 dimes** and **1 dime** (not **1 dimes**). Also, report the  sum in dollars and cents, i.e., **$5.73** instead of **573 cents**.