Khoa CNTT – Đại học Giao thông Vận tải

**LỚP BỔ TRỢ KIẾN THỨC SFIT**

**Môn: Kỹ Thuật Lập Trình**

*Phụ Trách: Lỗ Nguyễn Thành Vinh – CNTT5-K61*

**I. Tổng quan chương trình:**

**-** Gồm 9 phần gắn với các kiến thức liên quan đến học phần Tin học đại cương và Lập trình nâng cao, cũng như một số kiến thức bên ngoài, bao gồm:

* **Phần 1** Tổng quan về C.
* **Phần 2** Biến. Nhập xuất dữ liệu.
* **Phần 3** Các cấu trúc điều khiển
* **Phần 4** Đệ quy và Hàm.
* **Phần 5** Cấu trúc dữ liệu mảng.
* **Phần 6** Chuỗi trong C.
* **Phần 7** Làm việc với file.
* **Phần 8** Con trỏ trong C. Cấp phát bộ nhớ động.
* **Phần 9** Kiểu dữ liệu cấu trúc.
* **Phần 10** Danh sách liên kết.

- Thời gian bổ trợ: 24 buổi học. Trong đó:

* **Buổi 1**: Trình bày phần 1 và 2. Buổi 2 là bài tập tổng kết 2 phần.
* **Buổi 3 và 4**: Trình bày phần 3, bao gồm lý thuyết và bài tập.
* **Buổi 5, 6, 7**: Trình bày phần 4, bao gồm lý thuyết và bài tập.
* **Buổi 8**: Trình bày phần 5, bao gồm lý thuyết và bài tập.
* **Buổi 9**: Bài tập tổng hợp cho 5 phần ở trên.
* **Buổi 10 và 11**: Trình bày phần 6, bảo gồm lý thuyết và bài tập
* **Buổi 12**: Trình bày phần 7, bao gồm lý thuyết và bài tập.
* **Buổi 13**: Bài tập tổng hợp.
* **Buổi 14, 15, 16**: Trình bày phần 8. Bao gồm lý thuyết và bài tập.
* **Buổi 17, 18**: Trình bày phần 9. Bao gồm lý thuyết và bài tập
* **Buổi 19**: Bài tập tổng hợp.
* **Buổi 20, 21**: Trình bày phần 10. Bao gồm lý thuyết và bài tập.
* **Buổi 22, 23**: Bài tập tổng hợp và giải đáp thắc măc.
* **Buổi 24**: Tổng kết quá trình hướng dẫn.

- Thời gian học một buổi: **2 tiếng**, bắt đầu từ **18h đến 20h tối thứ hai hàng tuần**. Cách thức: **Kết hợp cả online và offline**. Địa điểm: Đối với hình thức offline sẽ học tại **Samsung Lab – A4 ĐH GTVT**.

**Phần 1. Tổng quan về ngôn ngữ lập trình C:**

**1.1. Các khái niệm về lập trình:**

- Chương trình là một tập hợp các câu lệnh nhằm hướng dẫn máy tính thực hiện một tác vụ nào đó.

- Lập trình là việc xây dựng, thiết kế ra các chương trình máy tính, sử dụng các ngôn ngữ lập trình.

- Ngôn ngữ lập trình là các công cụ sử dụng để xây dựng lên chương trình máy tính, được chia làm 3 loại là ngôn ngữ máy, hợp ngữ và ngôn ngữ lập trình bậc cao. Để máy tính có thể hiểu được, chương trình máy tính được viết bằng hợp ngữ và ngôn ngữ lập trình bậc cao phải được dịch sang ngôn ngữ máy.

- C thuộc ngôn ngữ lập trình bậc cao.

**1.2. Lịch sử ngôn ngữ C:**

- C được phát triển bởi Dennis Ritchie vào năm 1972, với mục đích chính là tạo ra một ngôn ngữ lập trình hệ thống để xây dựng các hệ điều hành (Unix,…).

- Cái tên C được sử dụng để biểu thị đây là một ngôn ngữ lập trình mới (sau B – Một ngôn ngữ lập trình khác được Ken Thompson phát triển).

- Ngày nay, phần lớn các hệ điều hành quan trọng đều được viết bởi C, bao gồm cả Windows.

**1.3. Một số đặc điểm của ngôn ngữ C:**

- C là một ngôn ngữ lập trình thủ tục, hỗ trợ lập trình có cấu trúc, phạm vi biến, đệ quy,…

- Các lý do khiến C phổ biến:

*i. Dễ học*

*ii. Là một ngôn ngữ lập trình cấu trúc.*

*iii. Cho ra các chương trình hiệu suất cao.*

*iv. Có thể xử lý các hoạt động mức low-level, rất gần với ngôn ngữ máy.*

*v. Có thể được biên dịch sang nhiều nền tảng. Tuy nhiên C lại độc lập nền tảng, phiên bản nền tảng này không thể chạy ở nền tảng kia.*

- Một số tính năng của C:

* Con trỏ: Cho phép tham chiếu đến một vị trí trong bộ nhớ bằng tên được gán cho nó trong chương trình.
* Cấp phát bộ nhớ động.
* Đệ quy
* Thao tác với bit.
* …

- Một số ví dụ hệ thống, chương trình được viết bằng C:

* Hệ điều hành (UNIX OS, Windows,…)
* Các trình biên dịch, thông dịch
* Text editor.
* Driver.
* Cơ sở dữ liệu.

…

**1.4. Cài đặt môi trường cho C:**

- Một chương trình viết bằng C được soạn thảo và lưu trữ trong một file đuôi “.c”. Trong việc lập trình C ta có thể sử dụng rất nhiều trình soạn thảo, như Notepad, Vim, VS Code, Sublime Text,…

- Để chạy chương trình, ta cần dịch file “.c” sang file thực thi “.exe” thông qua một trình biên dịch, từ đó CPU có thể thực thi. Một trong những trình biên dịch phổ biến nhất chính là GNU C/C++.

- Trong học phần này, chúng ta sẽ sử dụng DevC++ hoặc VS Code để thực hành.

**1.4.1. Cài đặt DevC++:**

- Do DevC++ đã tích hợp sẵn cả Text Editor và Trình biên dịch, nên ta không cần cài thêm trình biên dịch nữa.

**1.4.2. Cài đặt môi trường cho VS Code:**

**1.5. Chương trình đầu tiên:**

**-** Ta sẽ viết một chương trình đầu tiên là in ra “Hello, World!” ra ngoài màn hình. Tất nhiên ta có thể in bất kỳ thứ gì mà ta muốn.

#include <stdio.h>

**int** main()

{

printf(“Hello, World!”);

return 0;

}

**-** Việc in ra một dòng chữ ra màn hình có ý nghĩa rất quan trọng: Đầu tiên giúp kiểm tra xem quá trình cài đặt môi trường lập trình đã ổn định chưa, có phát sinh lỗi nào không; Thứ hai là giúp ta hình dung được cấu trúc của một chương trình C.

- Nhìn chung, một chương trình C sẽ gồm các phần:

***i. Documentation:*** *Đây là phần comment của chương trình và không được biên dịch.*

***ii. Linking:*** *Nạp các thư viện vào chương trình hoặc liên kết đến một chương trình khác.*

***iii. Define:*** *Định nghĩa các giá trị hằng số.*

***iv. Global Declaration:*** *Định nghĩa các biến toàn cục sử dụng trong toàn bộ chương trình.*

***v. Main****: Hàm đầu tiên được gọi để thực thi, chứa toàn bộ các câu lệnh chính được chạy.*

***vi. User Defined Function:*** *Chứa các hàm được người lập trình tạo ra nhằm bổ sung và phục vụ cho chương trình.*

- Sau khi soạn thảo chương trình, để chạy ta cần biên dịch chương trình thành mã máy thông qua một trình biên dịch. Quy trình sẽ như sau:

1. Thực hiện các câu lệnh preprocesser (Nạp thư viện, định nghĩa các giá trị hằng số, loại bỏ các câu comment).
2. Trình biên dịch dịch từng dòng code thành các chỉ dẫn máy. Tất cả các lỗi cú pháp nếu được phát hiện sẽ được báo lên trên command line. Cuối cùng nêu không có lỗi, trình biên dịch tạo một file object “.o” chứa toàn bộ mã máy của chương trình chúng ta.
3. Một trình liên kết được sử dụng để liên kết các object file được tạo ra sau quá trình dịch và tổng hợp thành một file thực thi “.exe” hoàn chỉnh.

**1.6. Kiểu dữ liệu, toán tử và biểu thức**

**1.6.1. Các kiểu dữ liệu trong C:**

- C là một ngôn ngữ dạng *kiểu*. Mỗi biến đều được gán một kiểu xác định giá trị nó có thể biểu diễn, cách dữ liệu được lưu trữ trong bộ nhớ và cách các toán tử có thể biểu diễn được trên nó.

- Trong C, kiểu dữ liệu gồm 3 loại chính: Ký tự (Character), Số nguyên (Integer), Số chấm phẩy động (Số thực – float). Mỗi kiểu lại có một kích thước xác định:

|  |  |  |  |
| --- | --- | --- | --- |
| Tên kiểu | Ký hiệu | Kích thước | Đặc tả |
| Ký tự | char | 0 – 256 (1 byte) | %c |
| Số nguyên | int | -2147483648 – 2147483647 (4 byte) | %d |
|  | unsigned Int | 0 – 4294967296 (4 byte) | %u |
|  | long Long Int | -(2^63) – (2^63)-1 (8 byte) | %lld |
| Số thực | float | 8 byte | %f |
|  | double | 16 byte | %lf |