

BÀI 1

CÁC KHÁI NIỆM CƠ BẢN

MỤC TIÊU

Học viên cần nắm được các khái niệm sau:

- Tập ký tự dùng trong ngôn ngữ C.
- Từ khóa và tên.
- Một số quy tắc khi viết chương trình.
- Sơ lược về khai báo biến, toán tử gán, câu lệnh vào và ra, toán tử include.

KIẾN THỨC CẦN CÓ

Để có thể nắm bắt và tiếp thu được môn học này, học viên cần có những kiến thức sau:

- Kiến thức về hệ điều hành: Chương trình C được xây dựng, lập trình và vận hành trên nền tảng windows 32 bit. Học viên cần có các khái niệm cơ bản về hệ điều hành windows như khởi động, sử dụng, điều chỉnh chương trình.
- Kiến thức về văn bản và gõ văn bản.
- Kỹ năng gõ bàn phím.
- Tư duy về lập trình, giải thuật.
- Học viên nên học môn Tin Học Cơ Bản trước khi bắt đầu với môn học này.

NỘI DUNG

1. Tập ký tự dùng trong ngôn ngữ.
2. Từ khóa (KEYWORD).
3. Tên (IDENTIFIER).
4. Ví dụ chương trình C.
5. Một số quy tắc cần nhớ khi viết chương trình.
6. Khai báo và toán tử gán.
7. Đưa kết quả lên màn hình.
8. Đưa kết quả ra máy in.
9. Vào số liệu từ bàn phím.
10. Một vài chương trình đơn giản.
11. Vận hành chương trình trên máy.

1. TẬP KÝ TỰ DÙNG TRONG NGÔN NGỮ

Ngôn ngữ lập trình C được xây dựng trên bộ ký tự sau:

- 26 chữ cái tiếng anh viết hoa: A,B,C,...,Z.
- 26 chữ cái tiếng anh viết thường: a,b,c,...,z.
- 10 chữ số hệ thập phân: 0,1,2,3,...,9.
- Các ký hiệu toán học như: +, -, *, /, =, ()
- Ký tự gạch dưới _.
- Các ký tự đặc biệt khác: ., : ; [] {} ! \ & % # \$...
- Dấu cách (space) dùng để tách các từ. Ví dụ chữ VIET NAM có 8 ký tự, còn VIETNAM chỉ có 7 ký tự.
- **Lưu ý:** Không được sử dụng bất kỳ ký tự nào khác ngoài các ký tự trên.
Ví dụ: Giải PT: $ax^2+bx+c = 0$, $\Delta = b^2 - 4ac$, Không dùng Δ .

2. TỪ KHÓA (KEYWORD)

- Từ khoá là những từ được sử dụng để khai báo các kiểu dữ liệu, để viết các toán tử và các câu lệnh. Các từ khóa này dành riêng cho ngôn ngữ lập trình C, người dùng không được sử dụng các từ khóa này cho việc đặt tên biến, hằng, mảng, hàm...
- Từ khóa đều phải viết bằng chữ thường:
Ví dụ: `int` (đúng) `INT` (sai)
- Danh sách các từ khóa chính của C:

`asm`

`break`

`case`

`char`

`int`

`const`

`union`

`float`

`Continous`

`default`

`do`

`far`

`if`

`void`

3. TÊN (IDENTIFIER)

- Tên được dùng để xác định các đối tượng khác nhau trong cùng 1 chương trình.
- Tên có thể là hằng, biến, mảng, hàm, con trỏ, tệp, cấu trúc, nhãn...
- 4 quy tắc đặt tên cơ bản:
 - Là 1 dãy các ký tự bao gồm chữ cái, số và gạch nối _.
 - Ký tự đầu tiên của tên phải là chữ hoặc gạch nối _.
 - Không trùng với từ khóa.
 - Dài tối đa 32 ký tự.
 - Ví dụ đúng : conmeo, a_1, _x1
 - Ví dụ sai: case (trùng từ khóa), 3XYZ(bắt đầu = số)

4. VÍ DỤ CHƯƠNG TRÌNH C

Chương trình in ra màn hình dòng chữ “Xin chao cac ban”:

```
#include <stdio.h>
#include <conio.h>
int main (void)
{
    printf("Xin chao cac ban");
    getch();
}
```


4. VÍ DỤ CHƯƠNG TRÌNH C (tiếp theo)

Chương trình in ra màn hình 2 dòng chữ:

```
Chao mung ban den voi  
Ky Thuat Lap Trinh C
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main (void)
```

```
{
```

```
printf("\n Chao mung ban den voi \  
      \n Ky Thuat Lap Trinh C");
```

```
getch();
```

```
}
```

5. MỘT SỐ QUY TẮC CẦN NHỚ KHI VIẾT CHƯƠNG TRÌNH

Quy tắc 1:

- Mỗi lệnh có thể viết trên một hay nhiều dòng và Kết thúc lệnh là dấu “ ; ”

Ví dụ:

```
printf("\n Chao mung ban den voi  
      \n Ky Thuat Lap Trinh C");  
printf("Xin chao cac ban");
```

Cả 2 trường hợp trên đều cho kết quả hiển thị chính xác, tương đương với 1 lệnh, dù rằng chúng ở 1 dòng và 2 dòng.

5. MỘT SỐ QUY TẮC CẦN NHỚ KHI VIẾT CHƯƠNG TRÌNH (tiếp theo)

Quy tắc 2:

- Khi viết nội dung chú thích bắt đầu là " /* " và kết thúc là " */ "
- Chú thích có thể viết nhiều cách:
 - Trên 1 dòng.
 - Trên nhiều dòng.
 - Trên phần còn lại của dòng.

Ví dụ:

```
/* Tinh tong 2 so */  
/* tinh  
   tong  
   hai  
   so  
*/
```

Chú thích chỉ có ý nghĩa cho người lập trình hiểu chương trình, không có ý nghĩa đối với việc thực thi chương trình.

5. MỘT SỐ QUY TẮC CẦN NHỚ KHI VIẾT CHƯƠNG TRÌNH (tiếp theo)

Quy tắc 3:

- Nếu trong chương trình có sử dụng các hàm chuẩn thì phải khai báo quy tắc theo khai báo như sau:
 - Bắt đầu bằng `# include`.
 - Tên hàm nằm trong 2 dấu `<>`.
 - Kết thúc không có dấu `\`.

Ví dụ:

```
#include <stdio.h>
```

5. MỘT SỐ QUY TẮC CẦN NHỚ KHI VIẾT CHƯƠNG TRÌNH (tiếp theo)

Quy tắc 4:

- Chỉ có duy nhất 1 hàm main().

Ví dụ:

```
int main(void)
{
    /* Thân chương trình */
}
```

- Chương trình chính có thể có nhiều hàm, các hàm này có thể thực hiện nhiều việc khác nhau, tương đương nhau tùy theo mục đích lập trình.
- Nếu chương trình không có hàm main thì chương trình không thể thực hiện được.

6. KHAI BÁO VÀ TOÁN TỬ GÁN

- Mọi biến trước khi dùng phải được khai báo.
- Quy tắc khai báo biến:

```
int          a,b,c ;          /* Chú thích tên biến */
```

Kiểu dl biến Danh sách tên biến Kết thúc danh sách biến là dấu ;

- Có nhiều kiểu biến khác nhau như int, float....
- Sau khi khai báo thì các biến có thể được gán giá trị nhiều lần.

```
float x;          /* Khai báo x          */  
x= 10.5;          /* Gán x = 10.5          */  
x = 2*x + 3;      /* Gán x = 2*10.5 + 3 = 24 */
```

7. ĐƯA KẾT QUẢ LÊN MÀN HÌNH

- Lệnh để đưa kết quả ra màn hình là:

```
printf(chuỗi điều khiển, bt1, bt2... btk);
```

- Chuỗi điều khiển có thể là:
 - Ký tự xuống dòng.
 - Ký tự hiển thị.
 - Ký tự mô tả kiểu cách đưa ra của biến.
 - Ký tự chuyển dòng \n.
 - Ký hiệu đặc tả khác.

7. ĐƯA KẾT QUẢ LÊN MÀN HÌNH (tiếp theo)

- Các ký hiệu đặc tả được dùng thường xuyên nhằm đưa ra các thông tin hiển thị chuyên biệt hơn. Cú pháp của ký hiệu đặc tả là:

`%[fw]d`

- fw : Là số nguyên xác định độ rộng tối thiểu của trường thông tin ra.
- []: Thể hiện thông tin bên trong nó không bắt buộc, như vậy fw cũng không bắt buộc phải được khai báo.
- d: Dùng cho kiểu int với biến.
- Đối với kiểu số thực, d được thay bằng f.

`%[fw][.pp]f`

7. ĐƯA KẾT QUẢ LÊN MÀN HÌNH (tiếp theo)

Một số ví dụ về hiển thị khi in ra màn hình:

- Code :

```
printf("abc");  
printf("%d \n",456);  
printf("%6d \n",456);  
printf("%0.3f \n",-45.63);
```

- Hiển thị:

```
abc  
456  
  456  
-45.630
```

8. ĐƯA KẾT QUẢ RA MÁY IN

- Cách thức đưa kết quả ra máy in hoàn toàn giống như cách thức hiển thị thông tin ra màn hình:

```
fprintf("Thong tin can in",bt1,bt2);
```

- fprintf : Hàm thực hiện in ra máy in.
- "Thong tin can in" /* Thông báo ra máy in*/
- bt1,bt2 /* Các biểu thức được sử dụng để in ra máy in*/
- Số lượng các biểu thức có thể nhiều hơn 2.

9. VÀO SỐ LIỆU TỪ BÀN PHÍM

Để vào số liệu từ bàn phím ta **cần phải** chú ý 2 bước sau:

- Bước 1: Khai báo biến sử dụng.

```
int a,b;  
float c,d,e;
```

- Bước 2: Dùng lệnh scanf để đọc thông tin từ bàn phím.

```
scanf("%d",&a;  
scanf("%d",&b);  
scanf("%f",&c);  
scanf("%f",&d);  
scanf("%f",&e);
```

10. MỘT VÀI CHƯƠNG TRÌNH ĐƠN GIẢN

Nhập 2 số nguyên, tính tổng, hiệu, tích, thương của 2 số nguyên đó.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
int a,b,tong,tich,hieu,thuong;
```

```
printf("\nNhap 2 so a,b = ");
```

```
scanf("%d%d",&a,&b);
```

```
tong=a+b.tich=a*b.hieu=a-b.thuong=a/b;
```

```
printf("Tong cua 2 so %d va %d la %d",a,b,tong);
```

```
Hay in ra tich, hieu va thuong
```

```
getch();
```

10. MỘT VÀI CHƯƠNG TRÌNH ĐƠN GIẢN (tiếp theo)

Nhập 2 số thực, tính tổng, hiệu, tích, thương của 2 số thực đó.

```
#include <stdio.h>

#include <conio.h>

int main(void)

{
float a,b,tong,tich,hieu,thuong;
printf("\nNhap 2 so a,b = ");
scanf("%f%f",&a,&b);
tong=a+b.tich=a*b.hieu=a-b.thuong=a/b;

printf("Tong cua 2 so %4.1f va %4.1f la %4.1f",a,b,tong);
//Hay in ra tich, hieu va thuong
getch();
}
```

10. MỘT VÀI CHƯƠNG TRÌNH ĐƠN GIẢN (tiếp theo)

Nhập 3 số thực, tìm max của chúng.

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
float a,b,c,max;
printf("\nNhap 3 so a,b,c = ");
scanf("%f%f%f",&a,&b,&c);
max=a;
if(b>max) max=b;
if(c>max) max=c;
printf("Maxcua 3 so %4.1f, %4.1f va %4.1f la %4.1f",a,b,c,max);
getch();
}
```

10. MỘT VÀI CHƯƠNG TRÌNH ĐƠN GIẢN (tiếp theo)

Tìm ước số chung lớn nhất của 2 số bất kỳ nhập vào từ bàn phím.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
int main(void)
```

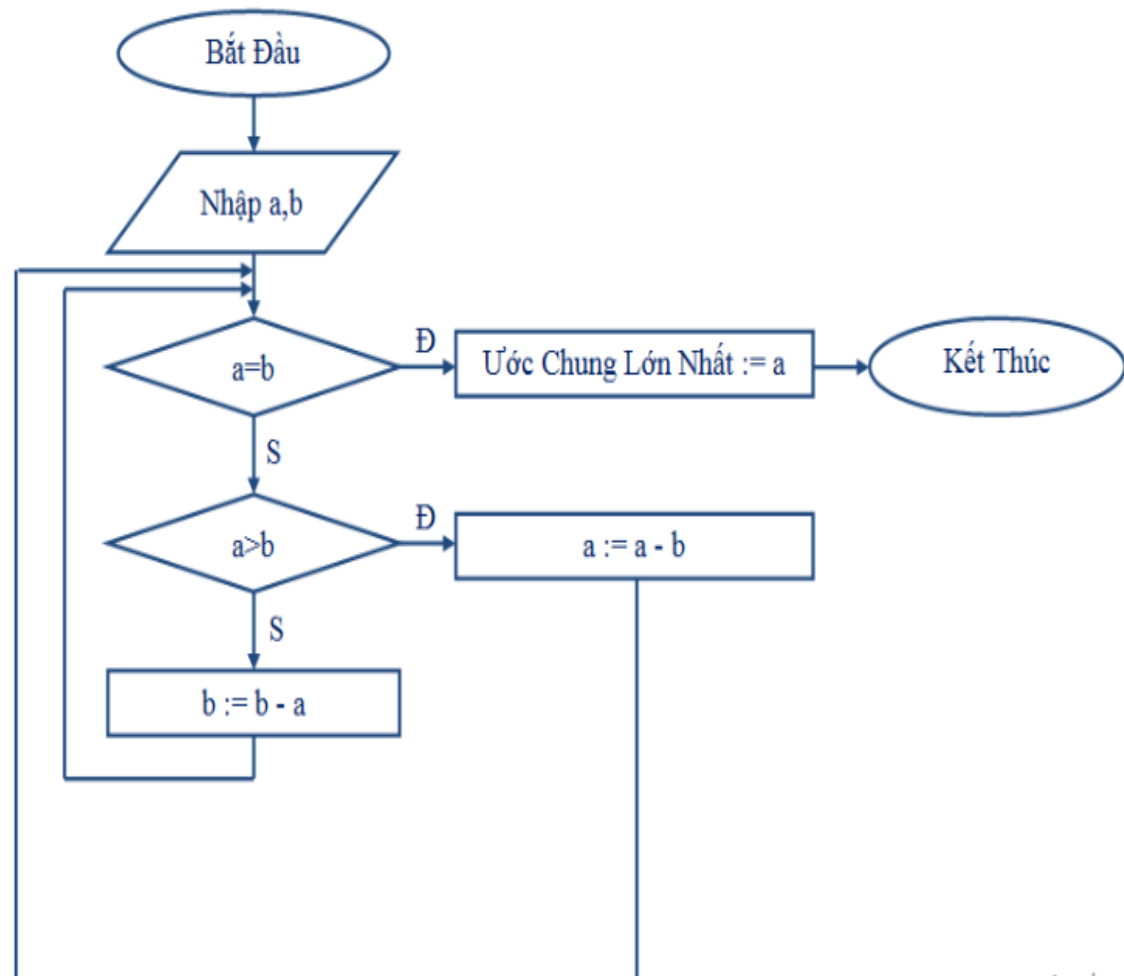
```
{ long a,b,t1,t2,ucln;
```

```
printf("\na, b =");
```

```
scanf("%ld%ld",&a,&b);
```

```
t1=a;
```

```
t2=b;
```



10. MỘT VÀI CHƯƠNG TRÌNH ĐƠN GIẢN (tiếp theo)

```
if (a == 0 || b == 0){
    ucln= a + b;
}
while (a != b){
    if (a > b){
        a -= b; // a = a - b
    }else{b -= a;}}
ucln= a; /* return a or b vì lúc này a=b
printf("\nUSCLN của %ld va %ld la: %ld ",t1,t2,ucln);
getch();
}
```


11. VẬN HÀNH CHƯƠNG TRÌNH TRÊN MÁY

- Các bước vận hành chương trình trên máy:
 - Soạn thảo 1 chương trình đuôi .c.
 - Dịch chương trình, tạo file có đuôi là .exe.
 - Chạy chương trình.
- Các bước thực hiện sau đây được sử dụng với chương trình Dev-C

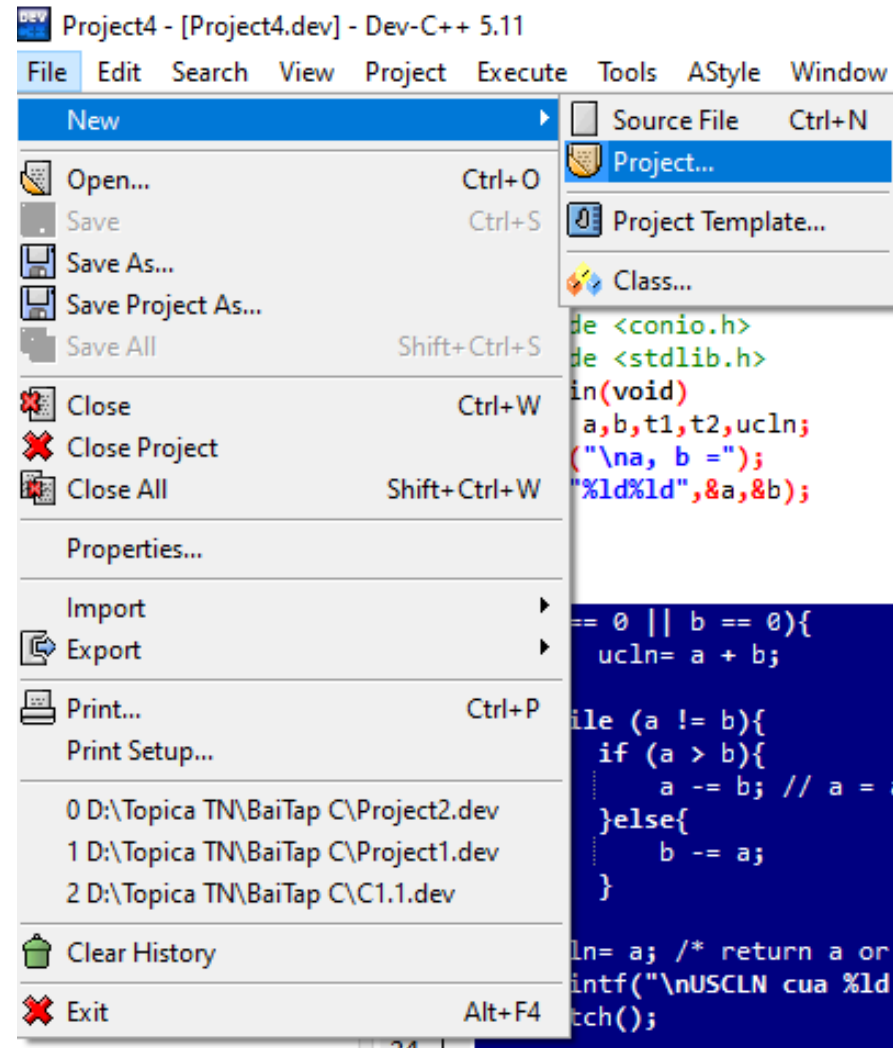
11. VẬN HÀNH CHƯƠNG TRÌNH TRÊN MÁY (tiếp theo)

Soạn thảo 1 chương trình đuôi .c có thể có nhiều cách:

- Cách 1: Dùng 1 phần mềm soạn thảo bất kỳ như word, notepad... Để soạn thảo rồi lưu dưới định dạng .c.
- Cách 2: Dùng phần mềm Dev-C để soạn thảo.
- Cách 3: Mở các file .c có sẵn và chỉnh sửa lại.
- Cách 4: Dùng các file .c có sẵn từ thư viện hoặc trong chương trình.

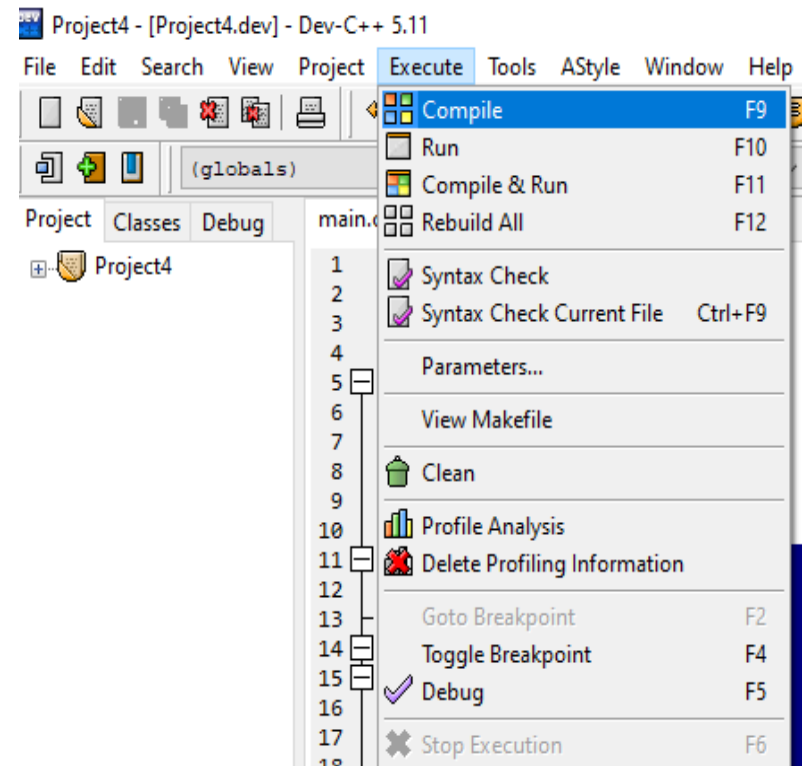
11. VẬN HÀNH CHƯƠNG TRÌNH TRÊN MÁY (Bước 1)

- Khởi động Dev-C.
- Chương trình sẽ chạy và hiển thị giao diện như hình.
- Vào File->New->Project để soạn thảo 1 chương trình mới.
- Thực hiện soạn thảo chương trình.



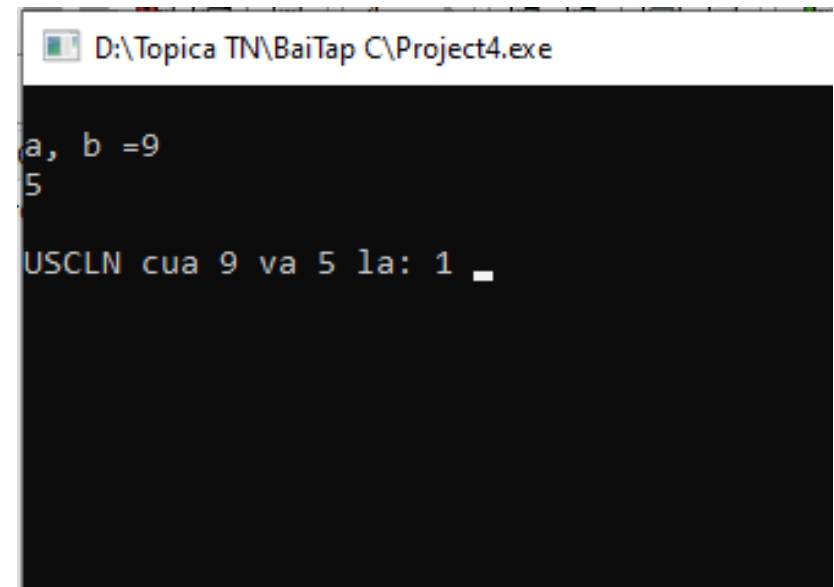
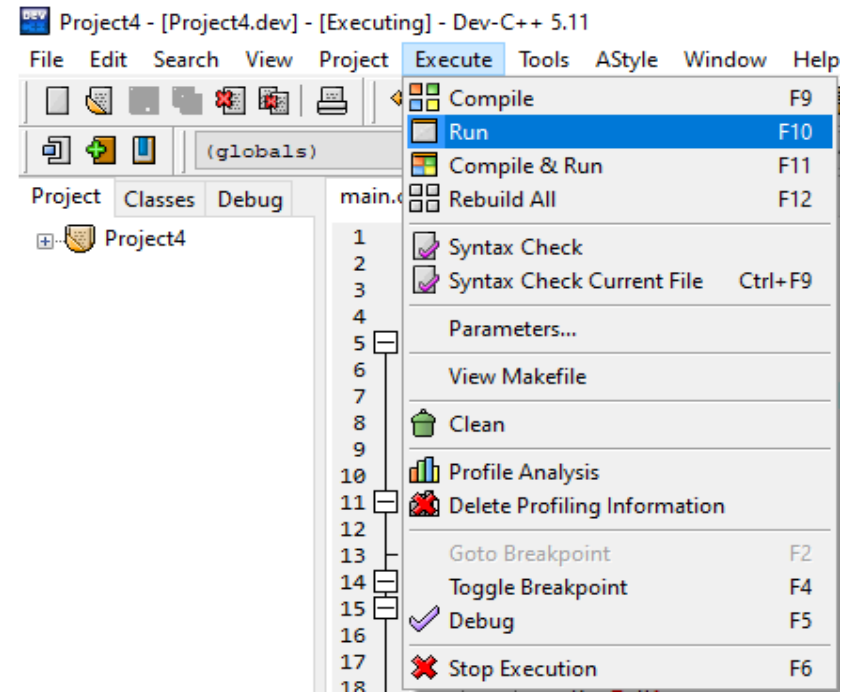
11. VẬN HÀNH CHƯƠNG TRÌNH TRÊN MÁY (Bước 2)

- Sau khi soạn thảo xong chương trình, ấn lệnh F9 hoặc vào Execute->Compile để biên dịch chương trình.
- Quá trình biên dịch sẽ cho ra 3 hướng kết quả khác nhau:
 - Chương trình có lỗi, sẽ ko thể chạy được.
 - Chương trình có cảnh báo, có thể chạy được.
 - Chương trình không có lỗi, chạy bình thường.



11. VẬN HÀNH CHƯƠNG TRÌNH TRÊN MÁY (Bước 3)

- Ấn F10 hoặc vào Execute->Run để chạy chương trình .
- Chương trình sẽ hiển thị ra màn hình DOS với nội dung thực hiện.



TÓM LƯỢC CUỐI BÀI

- Bài 1 đã trình bày các khái niệm cơ bản về ngôn ngữ lập trình C như khai báo biến, các thao tác vào và ra dữ liệu, các chương trình C đơn giản và cách thực hiện các chương trình C trên máy.
- Người học cần chú ý các bước thực hiện chương trình Dev-C. Cài đặt và chạy thử thành công các đoạn chương trình có trình bày trong slide.
- Cần phân biệt các quy tắc đặt tên biến, các từ khóa của chương trình để có thể đặt tên biến chính xác.
- Cần nhớ các quy tắc cơ bản khi viết chương trình.