



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# C PROGRAMMING INTRODUCTION

## TUẦN 4: BIẾN, HẲNG VÀ ĐẦU VÀO CHUẨN

# Nội dung

- Bién
  - Ôn tập lý thuyết
    - Bién
    - Các kiểu DL chuẩn
    - Hằng số
    - Đầu vào chuẩn
  - Bài tập thực hành

# Định danh

- Tên của các đối tượng (biến, hàm, v.v)

```
int nMyPresentIncome = 0;
```

```
int DownloadOrBuyCD();
```

# Quy tắc đặt tên

- Chữ cái, chữ số, dấu gạch dưới ‘\_’
  - i
  - CSE\_5a
  - a\_very\_long\_name\_that\_isnt\_very\_useful
  - fahrenheit
- Không bắt đầu bằng chữ số
  - 5a\_CSE không hợp lệ
- Phân biệt hoa thường
  - CSE\_5a khác cse\_5a

# Biến là gì?

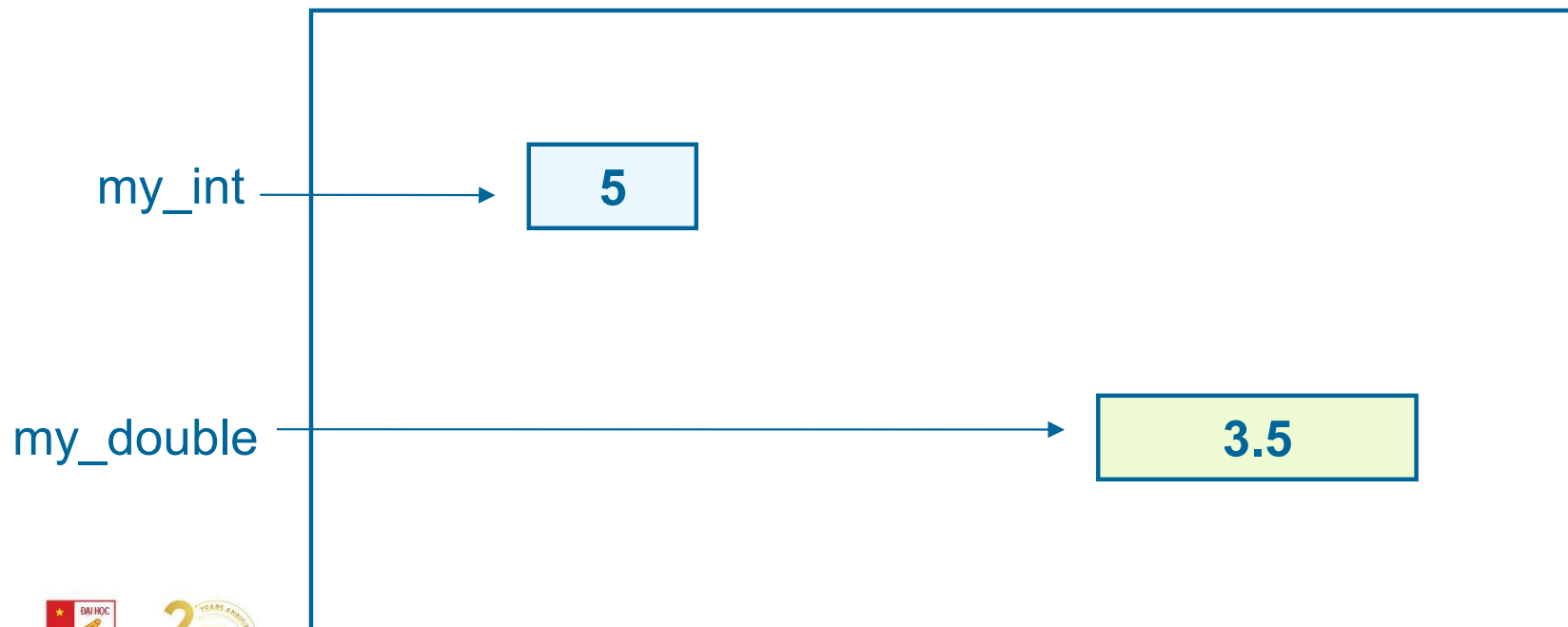
- Một vùng nhớ được đặt tên trong máy tính sử dụng để chứa DL ở một dạng nào đó (số nguyên, số thực, v.v)
- Chứa DL được chương trình sử dụng
- Tóm lại, biến là cách duy nhất để thao tác với DL

# Biến

- Vùng lưu trữ được đặt tên  
`int nRow = 0;`
- Kiểu của biến (kích thước và ý nghĩa của vùng lưu trữ)
- Phạm vi biến:
  - Biến địa phương bên trong khối lệnh
  - Tham số của hàm
  - Biến toàn cục
  - Chú ý không “giấu” biến
- Vòng đời (lớp lưu trữ)
  - Tự động/tạm thời (khối lệnh)
  - Toàn cục (chương trình)
  - Biến *static* (tĩnh) cục bộ (chương trình)

# Biến trong bộ nhớ

```
int my_int = 5;  
double my_double = 3.5;
```



# Khai báo, định nghĩa và khởi tạo

- Khai báo cung cấp vùng lưu trữ gọi là định nghĩa  
`int j;`
- Định nghĩa có thể gán giá trị cho biến (khởi tạo)  
`int j = 0;`
- Định nghĩa có thể mô tả ý nghĩa biến nhưng không cung cấp vùng lưu trữ (vd `extern`)  
`extern int j;`
- Chú ý: Các bản phát hành thường không khởi tạo biến bởi giá trị mặc định
- VD sử dụng biến:

```
printf(“%d + %d = %d\n“, a, b, c);
```



# VD: Khai báo biến

- `int i;`
- `char c;`
- `float f1, f2;`
- `float f1=7.0, f2 = 5.2;`
- `unsigned int ui = 0;`

# VD

```
1. #include <stdio.h>
2.
3. int main()
4. {
5.     int a, b, c;
6.     printf("The first number: ");
7.     scanf("%d", &a);
8.     printf("The second number: ");
9.     scanf("%d", &b);
10.    c = a + b;
11.    printf("%d + %d = %d\n", a, b, c);
12.    return 0;
13. }
```

12	c
7	b
5	a

**/ngonnguC/bin/tong**

**The first number: 5**

**The second number:7**

**5 + 7 = 12**

**/ngonnguC/bin/**

# Biến số và Hằng số (1)

- Biến số:

- Đặt tên cho một đối tượng bộ nhớ
- Sử dụng để lưu trữ giá trị và giá trị này có thể thay đổi
- Khai báo: Cho bộ biên dịch biết tên và kiểu biến <kiểu\_biến>  
tên\_biến;

VD:

int i;

float x, y, z;

char c;

- Phép gán: <tên\_biến> = <giá\_trị>;

VD:

i = 4;

x = 5.4;

y = z = 1.2;

# Biến số và Hằng số (2)

- Hằng số: có giá trị không đổi trong suốt chương trình

- Khai báo hằng:

- `#define <tên_hằng> <giá_trị>`

- VD:

- `#define TRUE 1`

- `#define FALSE 0`

# Hằng số (1)

- Hằng số nguyên

```
31          /* decimal */  
037        /* octal */  
0x1F       /* hexadecimal */  
31L        /* long */  
31LU       /* unsigned long */
```

- Hằng số thực

```
123.4       /* double */  
123.4F      /* float */  
123.        /* double */  
123.F       /* float */  
123.4L      /* long double */  
1e-2        /* double */  
123.4e-3    /* double */
```

- Hằng kí tự

```
'K'      /* normal ASCII - 'K' */  
'\113'   /* octal ASCII - 'K' */  
'\x48'   /* hexadecimal ASCII - 'K' */  
'\n'     /* normal ASCII - newline */  
'\t'     /* normal ASCII - tab */  
'\\'     /* normal ASCII - backslash */  
'\"'     /* normal ASCII - double quote */  
'\0'     /* normal ASCII - null (marks end of string) */
```

- Hằng chuỗi

"You have fifteen thousand new messages."

"I said, \"Crack, we're under attack!\"."

"hello," "world" trở thành → "hello, world"

- Kích thước và giới hạn (có thể thay đổi tùy theo máy) trên CUNIX:

<i>kiểu</i>	<i>kích thước (bit)</i>	<i>giới hạn</i>
char	8	-128...127
short	16	-32,768...32,767
int	32	-2,147,483,648...2,147,483,647
long	32	-2,147,483,648...2,147,483,647
float	32	$10^{-38}$ ... $3 \times 10^{38}$
double	64	$2 \times 10^{-308}$ ... $10^{308}$

- float có 6 bit cho độ chính xác
- double có 15 bit cho độ chính xác

# Các kiểu DL cơ sở (2)

- Bao gồm các các giá trị không dấu:

<i>kiểu</i>	<i>kích thước (bit)</i>	<i>giới hạn</i>
unsigned char	8	0...255
unsigned short	16	0...65,535
unsigned int	32	0...4,294,967,295
unsigned long	32	0...4,294,967,295

- Xem `/usr/include/limits.h`



# Định dạng đầu vào với `Scanf` (1)

- **`scanf`**

- Định dạng đầu vào
- Khả năng
  - Nhập tất cả các loại DL
  - Nhập các kí tự đặc biệt
  - Bỏ các kí tự đặc biệt

- Định dạng

**`scanf(chuỗi_điều_khiển_định_dạng [, các_tham_số]);`**

- chuỗi\_điều\_khiển\_định\_dạng - mô tả định dạng đầu vào
- các\_tham\_số - con trỏ tới các biến chứa giá trị DL
- có thể bao gồm các độ rộng trường để đọc một số lượng kí tự xác định từ luồng

# Định dạng đầu vào với Scanf (2)

Conversion specifier	Description
<i>Integers</i>	
<b>d</b>	Read an optionally signed decimal integer. The corresponding argument is a pointer to integer.
<b>i</b>	Read an optionally signed decimal, octal, or hexadecimal integer. The corresponding argument is a pointer to integer.
<b>o</b>	Read an octal integer. The corresponding argument is a pointer to unsigned integer.
<b>u</b>	Read an unsigned decimal integer. The corresponding argument is a pointer to unsigned integer.
<b>x</b> or <b>X</b>	Read a hexadecimal integer. The corresponding argument is a pointer to unsigned integer.
<b>h</b> or <b>l</b>	Place before any of the integer conversion specifiers to indicate that a <b>short</b> or <b>long</b> integer is to be input.
<i>Floating-point numbers</i>	
<b>e</b> , <b>E</b> , <b>f</b> , <b>g</b> or <b>G</b>	Read a floating-point value. The corresponding argument is a pointer to a floating-point variable.
<b>l</b> or <b>L</b>	Place before any of the floating-point conversion specifiers to indicate that a <b>double</b> or <b>long double</b> value is to be input.
<i>Characters and strings</i>	
<b>c</b>	Read a character. The corresponding argument is a pointer to <b>char</b> , no null ( <code>'\0'</code> ) is added.
<b>s</b>	Read a string. The corresponding argument is a pointer to an array of type <b>char</b> that is large enough to hold the string and a terminating null ( <code>'\0'</code> ) character—which is automatically added.
<i>Scan set</i>	
<i>[scan characters</i>	Scan a string for a set of characters that are stored in an array.
<i>Miscellaneous</i>	
<b>p</b>	Read an address of the same form produced when an address is output with <b>%p</b> in a <b>printf</b> statement.
<b>n</b>	Store the number of characters input so far in this <b>scanf</b> . The corresponding argument is a pointer to integer
<b>%</b>	Skip a percent sign ( <code>%</code> ) in the input.

# VD scanf()

```
int d,m,y,x;
char ch1,ch2;
float f;
scanf("%d", &x);           4
                        // x=4
scanf("%2d%2d%4d", &d,&m,&y); 22062007
                        // d=22, m=6, y=2007
scanf("%d/%d/%d", &d,&m,&y);  22/06/2007
                        // d=22, m=6, y=2007
scanf("%c%c", &ch1,&ch2);    Ab
                        // ch1='A', ch2='b'
scanf("%f", &f);           2.3
                        // f=2.300000
```

Result

# Định dạng đầu vào với `Scanf` (3)

- Đọc tập hợp
  - Tập hợp các kí tự nằm trong cặp ngoặc vuông `[]`
    - Theo sau dấu hiệu `'%'`
  - Đọc luồng đầu vào, chỉ quan tâm tới các kí tự trong tập hợp
    - Mỗi khi gặp kí tự phù hợp, lưu vào một mảng
    - Dừng khi gặp kí tự không phù hợp
  - Đọc kí tự không chứa trong tập hợp
    - Thêm dấu mũ `'^'`: `[^aeiou]`
    - Chứa các kí tự không nằm trong tập hợp

# Định dạng đầu vào với Scanf (4)

- Bỏ qua kí tự
  - Mô tả các kí tự bỏ qua khi đọc
  - hoặc sử dụng \*
  - Bỏ qua tất cả các kí tự

# VD 2

## •Đọc kí tự và chuỗi

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char x, y[ 9 ];
6
7      printf( "Enter a string: " );
8      scanf( "%c%s", &x, y );
9
10     printf( "The input was:\n" );
11     printf( "the character \"%c\" ", x );
12     printf( "and the string \"%s\"\n", y );
13
14     return 0;
15 }
```

Enter a string: Sunday  
The input was:  
the character "S" and the string "unday"

# VD 3

- Bỏ qua các kí tự trong tập

```
2 #include <stdio.h>
3
4 int main()
5 {
6     char z[ 9 ] = { '\0' };
7
8     printf( "Enter a string: " );
9     scanf( "%[^aeiou]", z );
10    printf( "The input was \"%s\"\n", z );
11
12    return 0;
13}
```

Enter a string: String  
The input was "Str"

# VD 4

## •Đọc và bỏ qua các kí tự từ luồng đầu vào

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int month1, day1, year1, month2, day2, year2;
6
7     printf( "Enter a date in the form mm-dd-yyyy: " );
8     scanf( "%d%c%d%c%d", &month1, &day1, &year1 );
9     printf( "month = %d  day = %d  year = %d\n\n",
10         month1, day1, year1 );
10    printf( "Enter a date in the form mm/dd/yyyy: " );
14    scanf( "%d%c%d%c%d", &month2, &day2, &year2 );
15    printf( "month = %d  day = %d  year = %d\n",
16        month2, day2, year2 );
17
18    return 0;
19}
```

```
Enter a date in the form mm-dd-yyyy: 11-18-2000
month = 11  day = 18  year = 2000
```

```
Enter a date in the form mm/dd/yyyy: 11/18/2000
month = 11  day = 18  year = 2000
```



# Bài tập 4.1

- Viết chương trình đọc một số nguyên và một số thực từ người dùng, sử dụng một biến số nguyên và một biến số thực dấu phẩy động để lưu và in ra màn hình giá trị các biến đã lưu.

# Lời giải (1)

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

```
void main (void) {
```

```
    /* We will use a floating-point and an integer variable. */
```

```
    double x;
```

```
    int n;
```

```
    /* Read in an integer. */
```

```
    printf("Please enter an integer: ");
```

```
    scanf("%d", &n);
```

```
    printf("The integer was %d\n\n", n);
```

# Lời giải (2)

```
/* Read in a double. */
```

```
printf("Please enter a double: ");  
scanf("%lf", &x);  
printf("The double was %g\n\n", x);
```

```
/* Read in an integer and a double. */
```

```
printf("Please enter an integer and a floating-point number: ");  
scanf("%d%lf", &n, &x);  
printf("The numbers were %d %g\n", n, x);
```

```
}
```

# Bài tập 4.2

- Viết và chạy chương trình để xem giới hạn của các kiểu DL int và long.
- Mở rộng chương trình với các kiểu DL khác
- Sử dụng thư viện limits.h để viết chương trình

# Lời giải (1)

```
#include <stdio.h>
#include <limits.h> /* defines INT_MIN, INT_MAX, LONG_MIN, LONG_MAX
*/

main()
{
    int j;
    long int k;
    float x;
    double z;

    printf("Enter an integer (between %i and %i): ", INT_MIN, INT_MAX);
    scanf("%i", &j);
    printf("You entered %d\n\n", j);
```

# Lời giải (2)

```
printf("Enter a long integer (between %ld and %ld): ", LONG_MIN,  
      LONG_MAX);  
scanf("%ld", &k);  
printf("You entered %li\n\n", k);  
  
printf("Enter a floating point number: ");  
scanf("%f", &x);  
printf("You entered %20.10E\n\n", x);  
  
printf("Enter a double precision floating point number: ");  
scanf("%lf", &z);  
printf("You entered %20.10E\n\n", z);  
  
puts("\n\nTry again: enter invalid data and see what happens!");  
}
```

# Bài tập 4.3

- Viết chương trình đọc một chuỗi đọc từ bàn phím có sử dụng tập hợp kí tự

# Lời giải

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

```
/* function main begins program execution */
```

```
int main( void )
```

```
{
```

```
    char z[ 9 ];
```

```
    printf( "Enter string: " );
```

```
    scanf( "%[aeiou]", z ); /* search for set of characters */
```

```
    printf( "The input was \"%s\"\n", z );
```

```
    return 0;
```

```
}
```



# Bài tập 4.4

- Viết chương trình nhập vào DL với độ rộng trường
- Mở rộng với tất cả kiểu DL cơ sở

# Lời giải

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

```
int main( void )
```

```
{
```

```
    int x;
```

```
    int y;
```

```
    printf( "Enter a six digit integer: " );
```

```
    scanf( "%2d%d", &x, &y );
```

```
    printf( "The integers input were %d and %d\n", x, y );
```

```
    return 0; /* indicates successful termination */
```

```
}
```

# Bài tập 4.5

- Viết chương trình yêu cầu người dùng nhập vào bán kính đường tròn. Sử dụng hằng số để lưu số  $PI$ .
  - a) In ra diện tích và chu vi đường tròn
  - b) In ra diện tích và thể tích khối tròn tương ứng

# Lời giải

```
#include <stdio.h>
#define PI 3.142

main()
{
    double r, c, ac, as, v;
    r = 5.678;
    printf("Radius = %f\n", r);

    c = 2.0 * PI * r;
    printf("Circle's circumference = %f\n", c);

    ac = PI * r * r;
    printf("Circle's area = %f\n", ac);

    as = 4.0 * PI * r * r;
    printf("Sphere's area = %f\n", as);

    v = 4.0/3.0 * PI * r * r * r;
    printf("Sphere's volume = %f\n", v);

}
```

# Bài tập 4.6

- Viết chương trình tính và hiển thị tiền lương của công nhân trong tuần. Số giờ làm việc trong tuần là 40 giờ + thời gian làm ngoài giờ. Thù lao một giờ làm việc là 25000VND và 40000 VND cho một giờ làm ngoài giờ. Trong tuần công nhân đã làm 50 giờ.

# Lời giải

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

```
int main()
```

```
{
```

```
    double regWages,    // calculated regular wages
```

```
        basePay = 25000, // base pay rate
```

```
        regHours = 40.0, // hours worked less overtime
```

```
        otWages,        // overtime wages
```

```
        otPay = 40000,  // overtime pay rate
```

```
        otHours = 10,   // overtime hours worked
```

```
        totalWages;     // total wages
```

```
    regWages = basePay * regHours;
```

```
    otWages = otPay * otHours;
```

```
    totalWages = regWages + otWages;
```

```
    printf("\n Wages for this week are %0.1f $", totalWages);
```

```
    return 0;
```

```
}
```

# Bài tập 4.7

- Viết chương trình hỏi người dùng về thông tin sách: ISBN, Tiêu đề, Giá tiền, Số lượng. Thuế VAT là 4%. Chương trình cần in ra các thông tin sau theo đúng giao diện:

BK Bookseller

Qty	ISBN	Title	Price	Total
-----	------	-------	-------	-------

VAT

You pay:





25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

