

25 YEARS ANNIVERSARY  
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# Tuần 4. Kiểu dữ liệu, biến, hằng

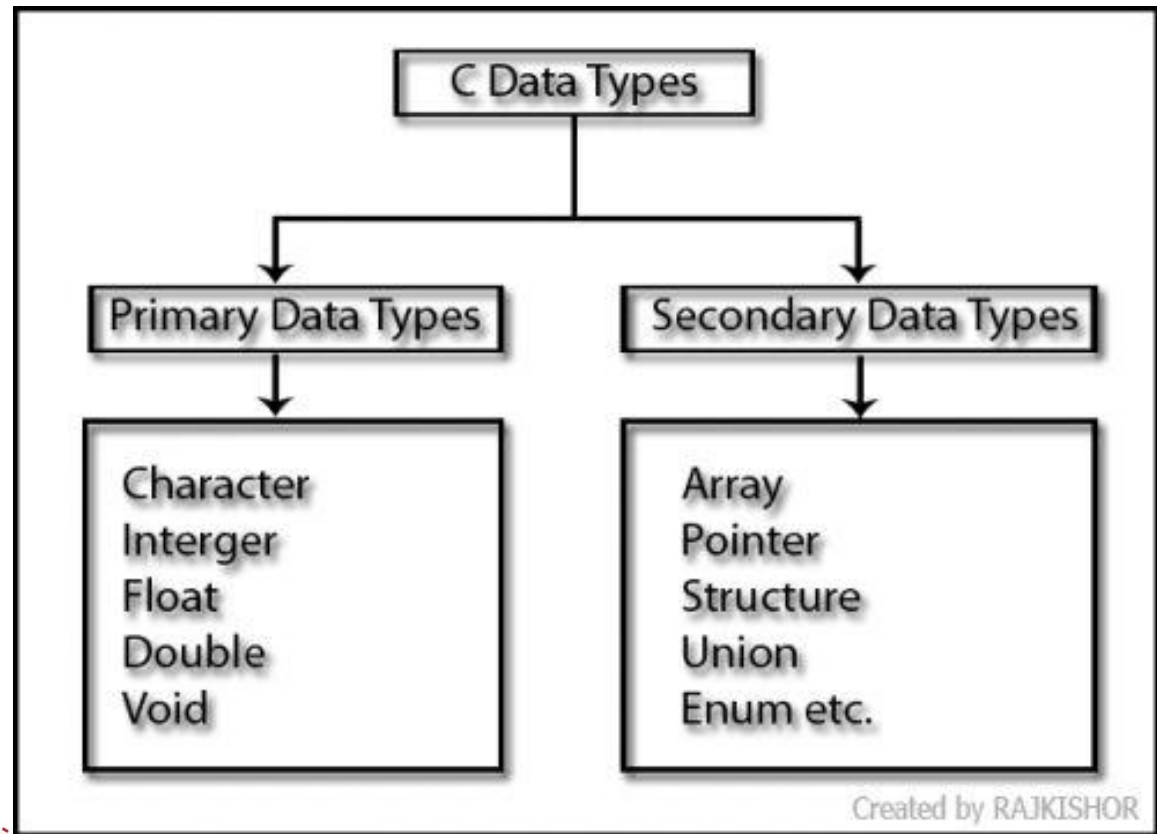
Data type, variable, const, scanf

# Nội dung

- Nội dung
  - Kiểu dữ liệu
  - Variables
  - Constants
  - Standard input.
- Bài tập

# Kiểu dữ liệu

- Các kiểu dữ liệu cơ bản trong C : **int**, **float**, **double**, **char**



# Kiểu dữ liệu

- Kiểu số nguyên cơ bản : **int**
  - 10, -23, 0 là các hằng số nguyên
  - Mặc định trong C các hằng số nguyên biểu diễn bằng hệ cơ số 10.
  - Ngoài hệ cơ số 10, hằng số nguyên còn được biểu diễn bằng hệ cơ số 8 (**octal**) và hệ cơ số 16 (**hexa**)
    - Hệ cơ số 8 : bắt đầu bằng số **0**, vd 050, 045
    - Hệ cơ số 16: bắt đầu bằng **0x**, vd 0x5F, 0xE5
  - In ra dùng **%d, %i , %o, %#o, %x, %#x**

```
printf("%d %d %d\n", 50, 050, 0x50);
```

In ra : 50 40 80

# Kiểu dữ liệu

- Các kiểu số nguyên khác: **char**, **short**, **long**, **long long**
- Kích thước lưu trữ

Type	Macintosh Metrowerks CW (Default)	Linux on a PC	IBM PC Windows XP Windows NT	ANSI C Minimum
<b>char</b>	8	8	8	8
<b>int</b>	32	32	32	16
<b>short</b>	16	16	16	16
<b>long</b>	32	32	32	32
<b>long long</b>	64	64	64	64

- Để chỉ rõ hằng kiểu **long** ta dùng thêm ký hiệu **l** hoặc **L** ở sau

# Kiểu dữ liệu

- Số nguyên có dấu và số nguyên không dấu: **signed** và **unsigned**
  - Mặc định các kiểu số nguyên là **signed**
  - Khai báo số nguyên không dấu: **unsigned**  
`unsigned int`, `unsigned long`, ...
  - Để chỉ ra rõ một hằng số nguyên là không dấu, dùng **u** hoặc **U** ở cuối  
`243U`, `34u`, `343454UL`

# Kiểu dữ liệu

- Kiểu số thực : **float**

- **3.**, **125.8**, và **-.0001** là các hằng số thực
- **3.5e+2**, **.12e3** là các hằng số thực dưới dạng ký pháp khoa học
- Kích thước biểu diễn : 32 bit

- In ra số thực

- **%f** dưới dạng dấu phẩy thập
- **%e** dưới dạng ký pháp khoa học
- **%g** tự điều chỉnh cho dễ nhìn

```
printf("%f %e %g", .00000012, 5.12, 50000.12);
```



# Kiểu dữ liệu

- Kiểu số thực mở rộng : **double**
  - Sử dụng 64 bit
  - Độ chính xác gấp đôi so với float
  - Để phân biệt 1 hằng số thực là float thì thêm ký hiệu f hoặc F ở cuối .34f 45.56F

# Kiểu dữ liệu

- Kiểu ký tự : **char**
  - ‘A’, ‘v’, ‘0’, ‘\n’ là các hằng ký tự
  - In ra bằng **%c**
  - Có thể dùng như một giá trị nguyên (chính là mã ASCII của ký tự đó)

```
printf ("%c %d", 'A', 'A') ;
```

# Kiểu dữ liệu

- Một số hằng ký tự

Ký hiệu	Ý nghĩa
\a	Audible alert
\b	Backspace
\f	Form feed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
\”	Double quote
\’	Single quote
\?	Question mark

# Kiểu dữ liệu

Type	Constant Examples	Printf char
char	'a', '\n'	%c
short int		%hi, %hx, %ho
unsigned short int		%hi, %hx, %ho
int	12, -97, 0xFFE0, 0177	%i, %x, %o
unsigned int	12u, 100U, 0xFFu	%u, %x, %o
long int	12L, -2001, 0xfffffL	%li, %lx, %lo
unsigned long int	12UL, 100ul, 0xffeeUL	%lu, %lx, %lo
long long int	0xe5e5e5e5LL, 5001l	%lli, %llx, %llo
unsigned long long int	12ull, 0xffeeULL	%llu, %llx, %llo
float	12.34f, 3.1e-5f	%f, %e, %g
double	12.34, 3.1e-5	%f, %e, %g
long double	12.341, 3.1e-51	%Lf, %Le, %Lg

# Các định danh (Identifier)

- Tên gọi của các biến, hàm ...

```
int nMyPresentIncome = 0;
```

```
int DownloadOrBuyCD();
```

# Quy tắc đặt tên định danh

- Chữ cái, chữ số, gạch chân
  - i
  - CSE\_5a
  - a\_very\_long\_name\_that\_isnt\_very\_useful
  - fahrenheit
- Ký tự đầu tiên không thể là chữ số
  - 5a\_CSE là **không hợp lệ!**
- Phân biệt chữ hoa chữ thường
  - CSE\_5a khác với cse\_5a

# Các *biến* là gì?

- Một vùng bộ nhớ được đặt tên, dùng để chứa các giá trị của một kiểu cụ thể (số nguyên, số thực....)
- Chứa dữ liệu mà chương trình làm/sẽ làm việc
- Cách duy nhất để thao tác dữ liệu

# Variables

- Vùng lưu trữ được đặt tên

```
int nRow = 0;
```

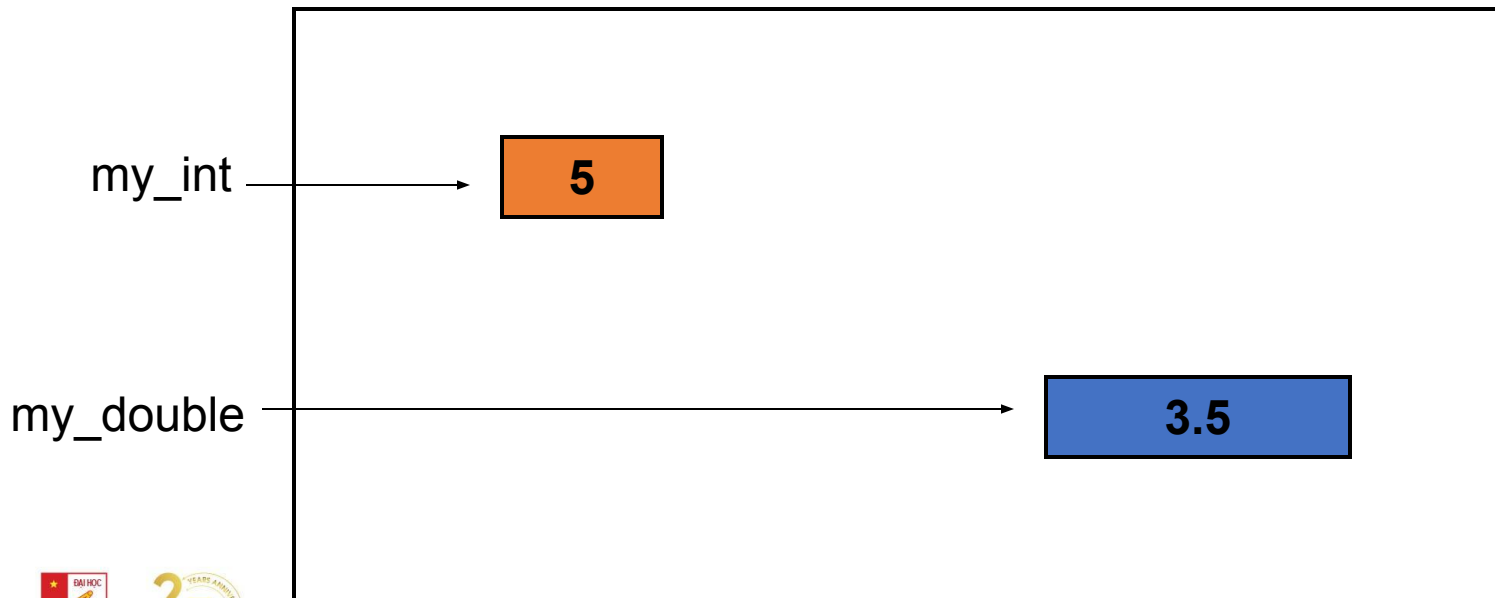
- Kiểu (kích thước và ý nghĩa của vùng lưu trữ)
- Phạm vi
  - Khối câu lệnh (Block)
  - Tham số cho các hàm (Function args)
  - Toàn cục (Global)
  - Cần thận để không “dấu” một biến
- Lifetime (storage class)
  - Tự động/ tạm thời (block's lifetime)
  - Globals (program's lifetime)
  - Local *static* (program's lifetime)



# Lưu trữ biến trong bộ nhớ

```
int my_int = 5;
```

```
double my_double = 3.5;
```



# Khai báo, định nghĩa, khởi tạo

- Các khai báo để dành trước vùng lưu trữ được gọi là các định nghĩa

```
int j;
```

- Các định nghĩa có thể gán một giá trị cho biến (khởi tạo)

```
int j = 0;
```

- Các khai báo chỉ ra ý nghĩa nhưng có thể không dành trước vùng lưu trữ (ví dụ *extern*)

```
extern int j;
```

- Sử dụng các biến:

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

# Ví dụ: khai báo dữ liệu

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

# Ví dụ

```
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/**

# Các biến và hằng số

- **Biến:**

- Tên một đối tượng bộ nhớ.
- Được dùng để lưu các giá trị và ta có thể thay đổi các giá trị này.
- Khai báo: Báo cho trình biên dịch (compiler) biết các biến và kiểu của biến

<typename> varname;

ví dụ:

int i;

float x, y, z;

char c;

- Gán giá trị: <varname> = <value>;

vd:

i = 4;

x = 5.4;

y = z = 1.2;

# Các biến và hằng số(2)

- Hằng số: giá trị không thay đổi trong toàn bộ chương trình.
  - Khai báo hằng số:  
`#define <tên hằng> <giá trị>`  
ví dụ:  
`#define TRUE 1`  
`#define FALSE 0`

# Hằng số (1)

- Integer constants

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

- Float constants

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

# Constants (2)

- Character constants

```
'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) */
```

- String literals

"You have fifteen thousand new messages."

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

"hello," "world" becomes "hello, world"



# Thiết lập khuôn dạng nhập dữ liệu với **Scanf**

- **scanf**

- **Thiết lập khuôn dạng nhập dữ liệu**

- **Khả năng**

- Nhập tất cả các kiểu dữ liệu
    - Nhập các ký tự cụ thể
    - Bỏ qua các ký tự cụ thể

- **Format**

**`scanf(format-control-string, other-arguments);`**

- **format-control-string** – miêu tả khuôn dạng của đầu vào
  - **other-arguments** – con trỏ tới các biến nơi dữ liệu sẽ được lưu vào
  - Có thể bổ sung độ rộng trường để đọc một số lượng cụ thể các ký tự từ luồng vào

# Thiết lập khuôn dạng nhập dữ liệu với `Scanf` (II)

Chỉ định	
<code>d</code>	Nhập số nguyên thập phân có dấu
<code>i</code>	Nhập số nguyên thập phân, bát phân, thập lục phân có dấu
<code>o</code>	Nhập số nguyên hệ 8 không dấu
<code>u</code>	Nhập số nguyên hệ 10 không dấu
<code>X</code> hoặc <code>x</code>	Nhập số nguyên hệ 16 không dấu. <b>X</b> hiển thị các chữ cái A-F và <b>x</b> hiển thị chữ cái <b>a-f</b>
<code>l</code> hoặc <code>L</code>	Đứng trước các chỉ định để chỉ ra số nguyên <b>short</b> hay <b>long</b> sẽ được hiển thị.

# Thiết lập khuôn dạng nhập dữ liệu với `Scanf` (III)

Chỉ định	
e, E, f, g, G	Nhập số thực
l hoặc L	Đứng trước các chỉ định để chỉ ra số thực <b>double</b> hay <b>long double</b> sẽ được nhập.
c	Nhập một ký tự
s	Nhập chuỗi ký tự

# Thiết lập khuôn dạng nhập dữ liệu với `scanf` (IV)

Chỉ định	
<code>scanset</code>	Dò tìm một xâu theo một tập ký tự cho trước
<code>p</code>	Nhập một địa chỉ, giống như khuôn dạng địa chỉ được in ra bởi <code>printf</code>
<code>n</code>	Lưu số ký tự được nhập vào bởi <code>scanf</code>
<code>%</code>	Bỏ qua dấu <code>%</code>

- Ví dụ về scanf()

```
int d,m,y,x;  
char ch1,ch2;  
float f;  
scanf("%d", &x);  
  
scanf("%2d%2d%4d", &d,&m,&y);  
  
scanf("%d/%d/%d", &d,&m,&y);  
  
scanf("%c%c", &ch1,&ch2);  
  
scanf("%f", &f);
```

### Result

```
4  
// x=4  
22062007  
// d=22, m=6, y=2007  
22/06/2007  
// d=22, m=6, y=2007  
Ab  
// ch1='A', ch2='b'  
2.3  
// f=2.300000
```

# Thiết lập khuôn dạng nhập dữ liệu với `Scanf` (III)

- Scan sets
  - Tập các ký tự nằm trong ngoặc vuông `[]`
    - Đứng trước bởi dấu `%`
  - Scans luồng vào , chỉ tìm các ký tự trong tập scan
    - Khi tìm thấy một ký tự, lưu ký tự vào mảng đã được chỉ định
    - Dừng việc scan khi thấy một ký tự không hợp
  - Đảo tập scan
    - Dùng dấu mũ `^`: `[^aeiou]`
    - Làm cho các ký tự không trong tập scan sẽ được lưu lại

# Thiết lập khuôn dạng nhập dữ liệu với `Scanf` (IV)

- Bỏ qua các ký tự
  - Bỏ qua ký tự trong điều khiển khuôn dạng
  - Hoặc dùng dấu \*
  - Bỏ qua bất kỳ kiểu dữ liệu mà không lưu nó

# Ví dụ 2

## •Đọc các ký tự và xâu

```
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"
```



# Example 3

## •Sử dụng một tập scan đã nghịch đảo

```
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"

# Example 4

## •Đọc và loại bỏ các ký tự từ luồng dữ liệu nhập 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
```

# Exercises 4.1

- Viết một chương trình đọc một số nguyên và số thực (double) từ người dung, sử dụng một biến kiểu dấu phẩy động(floating-point) và một biến kiểu integer để lưu trữ và sau đó hiển thị lại các giá trị đã nhập lên màn hình.

# Solution

```
#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);
```

# Solution

```
/* 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);
```

```
}
```

# Exercises 4.2

- Viết và chạy chương trình sau để thấy giới hạn của các kiểu dữ liệu cơ bản: int, long.
- Mở rộng chương trình cho các kiểu dữ liệu khác.
- Sử dụng thư viện limits.h để xây dựng chương trình.

# Solution

```
#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);
```

# Solution

```
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!");
}
```



# Exercises 4.3

- Viết một chương trình đọc vào một chuỗi từ bàn phím bằng cách dùng tập scan set.

# Solution

```
#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;

}
```

# Exercises 4.4

- Viết chương trình nhập dữ liệu vào với độ rộng trường.
- Mở rộng với tất cả các kiểu dữ liệu cơ bản.

# Solution

```
#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 */

}
```

# Exercise 4.5

- Viết một chương trình yêu cầu người dùng nhập vào bán kính vòng tròn Dùng hằng số cho số PI.
  - a) Hiển thị diện tích và chu vi đường tròn
  - b) Nếu coi dữ liệu vào là bán kính hình cầu. Tính diện tích diện tích bề mặt và thể tích hình cầu

# Solution

```
#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);
```

# Exercise 4.6

- Viết một chương trình tính và hiển thị lương một nhân viên trong tuần. Giờ làm việc quy định là 40 và số giờ quá 40 sẽ được xem là làm thêm.
- Nhân viên được trả 25000 VND một giờ và 40000 VND một giờ làm thêm.
- Tuần này nhân viên đã làm việc 50 giờ.

# Solution

```
#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;
```

```
}
```



# Exercise 4.7

- Viết một chương trình yêu cầu nhập thông tin về một cuốn sách bạn mua ở cửa hàng: ISBN, Title, Price, Quantity. Thuế VAT là 4%. Chương trình nên hiển thị các thông tin như giao diện dưới đây:

BK Bookseller

Qty	ISBN	Title	Price	Total
<hr/>				

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)





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