



澳門城市大學
Universidade da Cidade de Macau
City University of Macau

設計計算工作室I



主講人 |

姓名 張琪

Name Zhang Qi

澳門城市大學

City University of Macau



考核要求

- 課後書面作業（30%）
- 出勤（10%）
- 書面報告（60%）
 - 書面報告應不超過十頁A4紙，參考給定的實驗報告格式和模板。
 - 單次的實驗書面報告不超過十頁，如兩次實驗合并的書面報告不應超過二十頁。





實驗目的

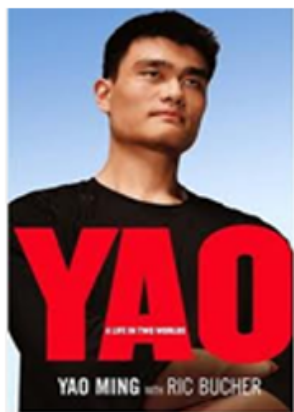
- 熟悉基本數據類型(int、char、float、double)占用的字節數和存儲形式，能夠定義基本數據類型的變量，使用這些類型的數據。
- 掌握字符數據的存儲形式及其與整數的運算，能正確理解“字符數據”與“字符串”的區別與聯繫，熟悉常用轉義字符的使用。
- 熟悉字符常量與一般常量的定義、使用。
- 掌握算術運算符(重點是/和%)、自增(++)和自減(--)運算符的使用，能夠構建正確的表達式，並輸出其結果。
- 理解不同類型數據之間運算時的類型轉換規則，掌握強制類型轉換的方法



實驗原理

1. 數據類型

- 在現實生活中,我們會接觸到許多數據,數據的性質可能存在著較大差異。程序設計要考慮數據的不同表現形式和操作方法。例如：



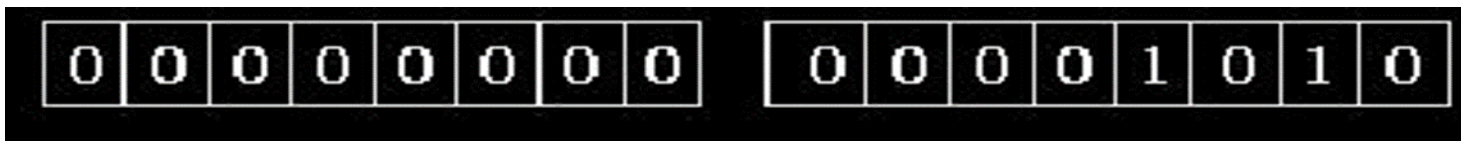
中文名：姚明	英文名：Yao Ming
出生地：上海	位置：中鋒
出生日期：1980.9.12	百米速度：15秒左右
身高：2.26m	體重：141kg

- 從上我們可以得到數據：姓名、出生地、位置，年齡：33歲，身高：2.26m,體重：141.0 kg ,...它們具有不同的性質:字符串有標識作用,數值型可計算,還有精度差異。編寫程序時，需區別對待

實驗原理

1. 數據類型

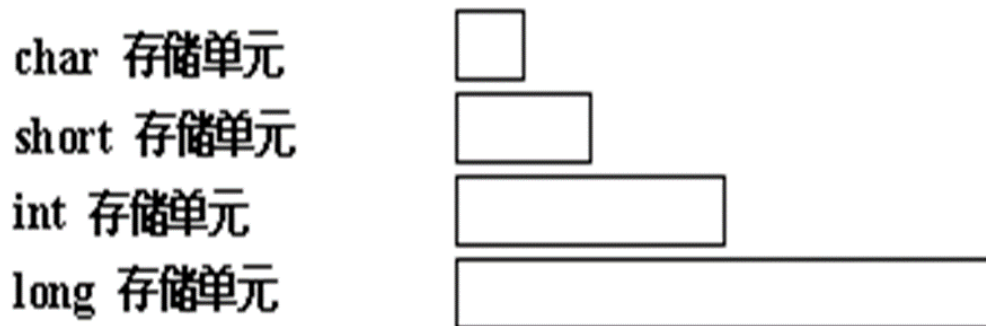
- 程序運行時所處理的數據需要存儲在內存中，不同性質的數據所占用存儲空間的多少、數值範圍及可進行的操作也不同；
- 所謂類型，就是對數據分配存儲單元的安排，包括存儲單元的長度(占多少字節)以及數據的存儲形式
- 不同的類型分配不同的長度和存儲形式
- 數據在計算機中的存儲是以字節為單位計算的，一個字節是8位。例如：整數10的用2字節表示的示圖



實驗原理

1. 數據類型

- 同一數據類型在不同計算機中占用的空間可能不同，如：
int型，在16位計算機中占2字節，在32位計算機中占4字節
- 數據占用內存字節的多少決定了其能表達數據的範圍

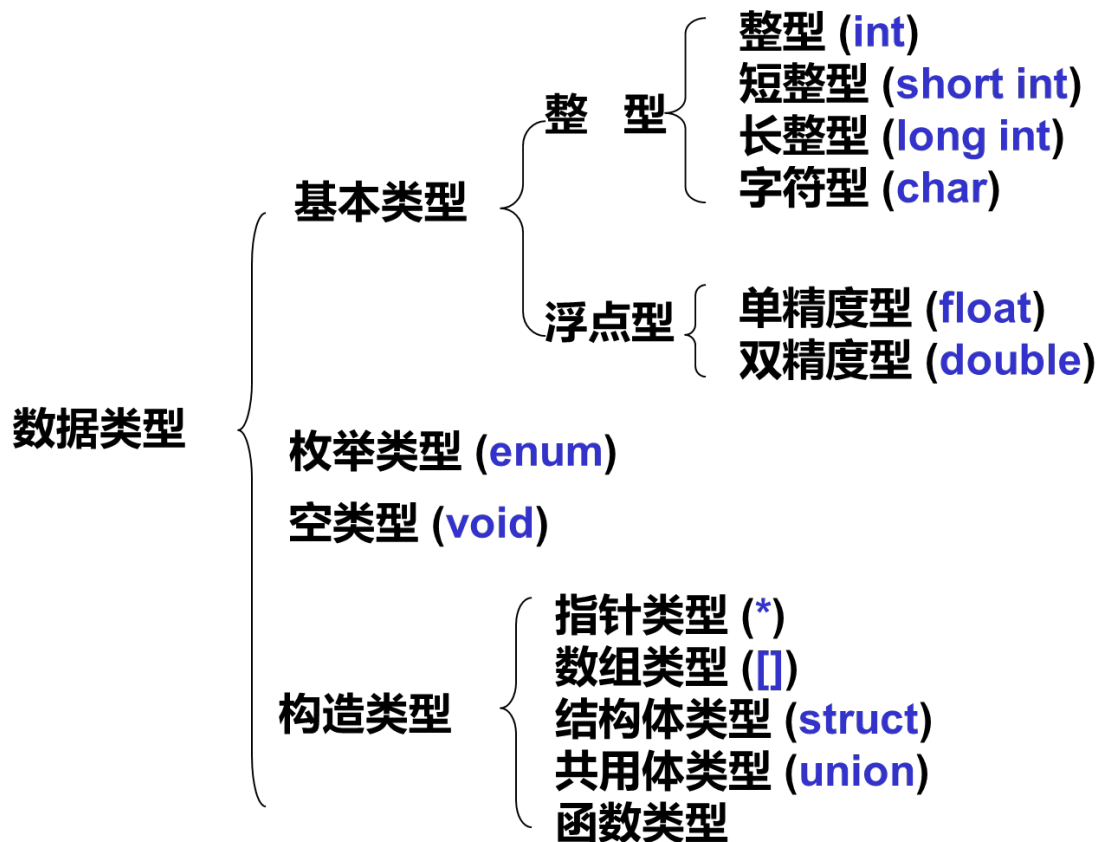


- 計算機內存中的同一數據當做不同類型來解釋時，所得結果是不同的；
- C語言包含的數據類型有：

實驗原理

1. 數據類型

- C語言包含的數據類型有：



實驗原理

1. 數據類型

- 數據類型情況重點掌握類型關鍵字、所占字節數、表示的數值範圍。下表是VC的基本數據類型：

数据类型	关键字	字节数	数值范围
字符型	char	1	-128 ~ 127
整型	int	4	-2147483648 ~ 2147483647
单精度浮点型	float	4	$\pm(3.4E-38 \sim 3.4E38)$
双精度浮点型	double	8	$\pm(1.7E-308 \sim 1.7E308)$
空值型	void	0	

- C語言還允許在基本數據類型（除void類型外）前加上類型修飾符，更具體地表示數據類型。C語言的類型修飾符包括：
 - signed(有符號) unsigned(無符號)
 - short(短型) long(長型)

實驗原理

1. 數據類型

- 基本整型(int)：不同編譯系統或計算機系統中所占用的字節不同，通常為2字節(TC)或4字節(VC)
- 在VC中，占4字節(32位)，
 - 範圍：-2147483648 ~ 2147483647
- 存儲方式：以二進制補碼形式存放數據(以VC系統為例)

12

00000000	00000000	00000000	00001100
----------	----------	----------	----------

-12

11111111	11111111	11111111	11110100
----------	----------	----------	----------

$2^{31}-1$ (最大整數)

01111111	11111111	11111111	11111111
----------	----------	----------	----------

實驗原理

1. 數據類型

- 短整型(short int)：為2字節，範圍：-32768 ~ 32767(即 $2^{15}-1$)
- 長整型(long int)：為4字節，範圍：-2147483648 ~ 2147483647(即 $-2^{31} \sim 2^{31}-1$)
- 雙長整型(long long int)：一般分配8字節，是C99新增類型，許多編譯系統尚未實現
- 可用：sizeof(類型名或數據) 來獲取該類型數據所占字節數。(注：Sizeof是運算符，不是函數)
- 雖然在不同系統中，同一數據類型所占用字節存在著差異，但同一系統不同數據類型的長度有明確的大小關係，即：
 $\text{sizeof}(\text{short}) \leq \text{sizeof}(\text{int}) \leq \text{sizeof}(\text{long}) \leq \text{sizeof}(\text{long long})$

實驗原理

1. 數據類型

- 例1：使用sizeof運算符獲取數據、類型的所占字節數

```
#include <stdio.h>

int main()
{
    printf("int      型数据所占字节数: %d\n",sizeof(int));
    printf("short int  型数据所占字节数: %d\n",sizeof(short int));
    printf("long int   型数据所占字节数: %d\n",sizeof(long int));
    printf("long long int型数据所占字节数: %d\n",sizeof(long long));
    printf("整数 2345 所占字节数: %d\n",sizeof(2345));
    return 0;
}
```

```
int      型数据所占字节数: 4
short int  型数据所占字节数: 2
long int   型数据所占字节数: 4
long long int型数据所占字节数: 8
整数 2345 所占字节数: 4
```



實驗原理

1. 數據類型

- 可在整型前加unsigned來定義無符號整數
- 例如：unsigned int、unsigned short int、unsigned long int、unsigned long long int，由于最高位不再用來存放整數的符號，因此，可表示整數的範圍擴大一倍
- 說明：
 - 只有整型(包括字符型)可加unsigned或signed，實數不允許



實驗原理

1. 數據類型

(2) 字符型數據(由一個字符組成)

- 字符是按其代碼(整數)形式存儲的
- C99把字符型數據作為整數類型的一種
- 字符型數據在使用上有自己的特點
- 字符與字符代碼
 - 大多數系統採用ASCII字符集
 - 字母：A ~ Z，a ~ z
 - 數字：0 ~ 9
 - 專門符號：29個：! " # & ' () * 等
 - 空格符：空格、水平制表符、換行等
 - 不能顯示的字符：空(null)字符(以 '\0' 表示)、警告(以 '\a' 表示)、退格(以 '\b' 表示)、回車(以 '\r' 表示)等

實驗原理

1. 數據類型

(2) 字符型數據(由一個字符組成)

- 字符' 1' 和整數1是不同的概念：
- 字符' 1' 只是代表一個形狀為' 1' 的符號，在需要時按原樣輸出，在內存中以ASCII碼形式存儲，占1個字節(在所有系統中)
- 整數1是以整數存儲方式(二進制補碼方式)存儲的，TC系統占2個字節，VC系統4個字節，例如：

00110001

00000000

00000000

00000000

00000001

- 字符型的定義與輸出：
 - 用char定義字符變量，例如：
 - `char c = ' ?' ;` //系統把 " ? " 的ASCII代碼63賦給變量c
 - 用%c輸出字符，用%d輸出整數，例如：
 - `printf(" %d %c\n" ,c,c);` //輸出結果是：63 ?

實驗原理

1. 數據類型

(2) 字符型數據(由一個字符組成)

- 字符型數據的存儲空間和值的範圍：

类型	字节数	取值范围
<code>signed char</code>	1	-128~127, 即 $-2^7 \sim (2^7-1)$
<code>unsigned char</code>	1	0~255, 即 $0 \sim 2^8-1$

- 說明：

- 將一個負整數賦給有符號字符變量是合法的
- 例如：`signed char c = -6;` //不代表一個字符，用1字節整型變量存儲負數
- 如果char前面未加signed或unsigned,具體含義由各系統決定
- 例如：`char c=255; printf(" %d\n" ,c);` //在VC中輸出結果是：-1，即默認為signed char

實驗原理

1. 數據類型

(3) 浮點型數據

- 浮點型數據是用來表示具有小數點的實數
- 在C語言中，實數是以指數形式存放在內存存儲單位中的

數符	小数部分	指數
----	------	----

- 以圓周率PI的近似值3.14159為例，可寫成： 3.14159×10^0 、 0.314159×10^1 、 0.0314159×10^2 、 31.4159×10^{-1} 等形式。由於小數點位置可以浮動，故稱為浮點數
- 規格化指數形式：小數點前數字為0、小數點的第1位數字非0對於一個實數來說，其規格化指數形式只有一種，例如：
 0.314159×10^1 符合要求， 0.0314159×10^2 、 31.4159×10^{-1} 等則不符合要求
- 指數表示了數據的大小，尾數表示了數據的精度。通常用e或E後面帶整數表示10的多少次方



實驗原理

1. 數據類型

(3) 浮點型數據

- float型(單精度浮點型)
 - 編譯系統為float型變量分配4個字節，精確到6位有效數字，範圍為 $-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
 - 數值以規範化的二進制數指數形式存放
- double型(雙精度浮點型)
 - 編譯系統為double型變量分配8個字節，精確到15位有效數字，範圍為 $-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$
 - 數值以規範化的二進制數指數形式存放



實驗原理

2. 常量和變量

(1) 標識符

- 在C語言中，有許多符號需要命名，如：變量名、函數名、數組名等，這些符號稱為標識符
- 標識符的命名規則：由字母、數字、下劃綫組成，並且第一個字符必須為字母或下劃綫。
- 合法標識符：area、PI、a_array、price等
- 非法標識符：456p、cade-y、x.y、a&b等
- 不同的C語言編譯系統所規定的標識符長度可能不同
- 建議：命名時最好“見名知意”，多採用單詞、拼音、下劃綫等

實驗原理

2. 常量和變量

(1) 標識符

- 標識符有三類：
- **關鍵字**：C語言預先規定的一批有固定含義的標識符，不能另作他用

auto	double	int	struct
case	else	long	switch
break	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while 等

- 預定義標識符：C語言預先定義并有特定含義的標識符，如：庫函數名(`printf`)、預處理命令(`define`)。不建議另作他用
- 用戶標識符：用戶定義的標識符，如：變量名、函數名、數組名等，最好“見名知意”

實驗原理

2. 常量和變量

(2) 常量

- 概念：在程序運行過程中，其值不能被改變的量
- 包括：整型常量、實型常量、字符常量和字符串常量等
- 1) 整型常量：
 - 十進制整數：由正號(+)或負號(-)開始的、接著為首位非0的若干個十進制數字所組成。如38，-25，+120，74286等
 - 十六進制整數：由數字0和字母x（大、小寫均可）開始的、後接若干個十六進制數字（0-9，A-F或a-f）所組成。同八進制整數一樣，十六進制整數也均為正數。如0x0，0X25，0x1ff，0x30CA等

實驗原理

2. 常量和變量

(2) 常量

- 概念：在程序運行過程中，其值不能被改變的量
- 包括：整型常量、實型常量、字符常量和字符串常量等
- 符號常量
 - 格式：`#define` 符號 常量
 - 例如：`#define num 50`
 - 處理：編譯前，預處理器將所此符號置換成常量。由此可知，符號常量不同于變量，不占用內存空間
 - 好處：做到“一改全改”

實驗原理

2. 常量和變量

(2) 常量

- 概念：在程序運行過程中，其值不能被改變的量
- 包括：整型常量、實型常量、字符常量和字符串常量等
- 轉義字符：
 - 格式：`\`特定字符
 - 性質：改變原先字符的含義，實現特定功能
 - 例如：`\n`（換行功能），`\t`（下一制表位）
 - 還允許用反斜綫引導一個具有1至3位的八進制整數(形式：`'\ddd'`)或一個以字母x作為開始標記的具有1至2位的十六進制整數(形式：`'\xhh'`)，對應的字符就是以這個整數作為ASCII碼的字符
 - 例如：`\101`、`\x41` 均表示ASCII為65的字符,即 `'A'`

實驗原理

2. 常量和變量

(2) 常量

- 常用的轉義字符：

转义字符	含义	ASCII码值 (十进制)
<code>\a</code>	响铃 (bell)	7
<code>\b</code>	退格 (backspace)	8
<code>\n</code>	换行 (newline)	10
<code>\r</code>	回车 (carriage return)	13
<code>\t</code>	水平制表 (horizontal tab)	9
<code>\v</code>	垂直制表 (vertical tab)	11
<code>\\</code>	反斜杠	92
<code>\'</code>	单引号	39
<code>\''</code>	双引号	34
<code>\0</code>	空格符 (null)	0
<code>\ddd</code>	任意字符	3位八进制数
<code>\xhh</code>	任意字符	2位十六进制数

實驗原理

2. 常量和變量

3) 字符串常量：

- 格式：“...” ，即：用一對雙引號括起來的字符序列(字符個數可以0、1或多個，也可以是轉義字符)，起標識、提示作用。字符串以雙引號為定界符，雙引號不作為字符串的一部分。如：“Hello” ， “Good Morning!” ， “廣州大學” ...
- 長度：字符串的字符數。如：“Hello” 的長度為5.
- 存儲：在內存中占一串連續的存儲單元，系統自動在字符串的末尾加以字符串結束標志，即轉義字符 ‘\0’ 。

實驗原理

2. 常量和變量

4) 浮點常量：

- 格式：有兩種
- 小數形式：1.23，0.23，-999.34，3.14159，...
- 指數形式：1.23e5，2.3e-1，-0.999E3，...
- (注：e或E均可，e前是小數，e後是整數)
- 通常，浮點數當作double型看待。可在尾部加f或F指定為float型

5) 怎樣確定常量的類型？

- 字符常量：由單撇號(' ')括起來的單個字符或轉義字符
- 字符串常量：由雙撇號(" ")括起來的單個字符或轉義字符
- 整型常量：不帶小數點的數值，系統根據數值的大小確定int型還是long型等
- 浮點型常量：凡以小數形式或指數形式出現的實數

實驗原理

2. 常量和變量

例2 輸入圓的半徑，輸出其周長、面積

解題思路：

圓的半徑用scanf()輸入，再用公式計算其周長、面積

圓的周長、面積計算都要用PI，可用符號常量表示

```
#include <stdio.h>
#define PI 3.14159
int main()
{
    float r,c,s;
    printf("请输入圆的半径:");
    scanf("%f",&r);
    c=2*PI*r;
    s=PI*r*r;
    printf("圆的周长=%f,面积=%f\n",c,s);
    return 0;
}
```

```
请输入圆的半径:10
圆的周长=62.831799,面积=314.158997
```

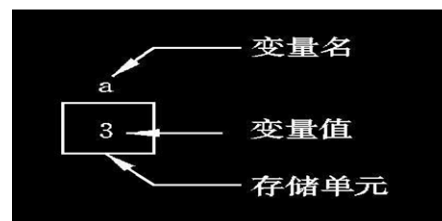
實驗原理

2. 常量和變量

(3) 變量

- 概念：在程序運行過程中，其值可以被改變的量
- 區分：變量與變量名的不同
- 變量名：實際上是一個符號地址，
- 在對程序編譯連接時由系統給每一
- 個變量名分配一個內存地址。
- 注：變量名是標識符，命名要“見名知意”
- 變量值：變量的取值，可以在程序中改變，它對應著一種數據類型
- 在程序中，實際上是通過變量名找到相應的內存地址，從該存儲單元中讀取數據

int a=3;



實驗原理

2. 常量和變量

(3) 變量

- 變量定義：數據類型 變量名1 [,變量名2,...];
- 實質：在編譯時為其分配相應的存儲單元
- 例如：`int a,b,c; float r; double root1,root2;`
- 說明：
 - 變量定義時，都有一個確定的數據類型，這樣才能分配存儲單元；
 - C語言規定，變量要“先定義，後使用”，編譯時系統進行檢查。例如，如果在定義時為 `int num;` 而在執行語句中錯寫成 `nmu`，編譯時就會輸出“變量 `nmu` 未經聲明”的信息
 - 變量定義後，如果未賦值，則其值是不確定的。應避免這種情況的出現
 - 變量初始化：數據類型 變量名1=初值 [,變量名2=初值,...];
 - 功能：將變量定義、賦值“合二為一”

實驗原理

2. 常量和變量

字符數據與整數的相互轉換

- 字符數據存放在內存的是其ASCII碼，可與當作整數處理；
- 熟悉常用字符的ASCII(十進制數)：
A~Z(65~90), a~z(97~122), 0~9(48~57)，空格(32), null(0)等

例3 將字符賦給整型變量

```
#include <stdio.h>
int main()
{
    int i,j;
    i='A';
    j='D';
    printf("輸出字符的ASCII:
%d,%d\n",i,j);
    return 0;
}
```

輸出字符的ASCII: 65,68

例4 字符數據與整數進行算術運算

```
#include <stdio.h>
int main()
{
    char c1,c2;
    c1='A';
    c2='B';
    c1=c1+32;
    c2=c2+32;
    printf("輸出整數對應的字符:
%c,%c\n",c1,c2);
    return 0;
}
```

輸出整數對應的字符: a,b

實驗原理

3. 運算符和表達式

(1) 一些概念

- 運算符(操作符)：是對數據進行運算的符號
- 操作數(運算對象)：參與運算的數據
- 表達式：由操作數和操作符連接而成的有效的式子。表達式可以嵌套，例如： $2+3+(5*\text{sizeof}(\text{int}))/345$
- 運算符分類：（按操作數個數的多少劃分）
 - 單目運算符：一般位於操作數的前面，如：正負號(+、-)
 - 雙目運算符：運算符一般位於兩個操作數之間，如： $a+b$
 - 三目運算符：只有一個，即為條件運算符，它含有兩個字

實驗原理

3. 運算符和表達式

- 在學習運算符時要掌握幾個要點：

- 符號：怎樣書寫
- 功能：做什麼
- 目數：操作數的多少
- 優先級：用來決定它在表達式中的運算次序。如計算表達式 $a + b * (c - d) / e$ 時，運算次序依次為：-, *, /, +。
- 結合性：是指表達式中出現同等優先級的操作符時，該先做哪個操作的規定
- 從左到右：如 $d = a + b - c$; （大多數表達式）
- 從右到左：如 $d = a = 3$; // C++ 規定，賦值號是先右後左，先做 $a = 3$, 其結果再賦給 d （單目、賦值、條件運算符適用）

- 左值和右值：

- 左值：是能出現在賦值表達式左邊的表達式。左值表達式具有存放數據的空間，允許其存放數據

實驗原理

3. 運算符和表達式

- 例如：`int a=3;` //a是變量，所以a是左值
`const int b=4;`//b是常量，所以b不是左值。顯然常量不是左值，因為C語言規定常量的值一旦確定就不能更改的。
- 右值：只能出現在賦值表達式的右邊。左值表達式也可以作為右值表達式。
- 例如：
`int a,b=6;`
`a=b;` //b是變量，是左值，此處作右值
`a=8;` //8是常量，只能作右值，不能作左值

實驗原理

3. 運算符和表達式

(2) 算術運算符與表達式

● 運算符(5個)：

- $+$, $-$, $*$: 含義與數學上相同; (注：兩個數相乘時， $*$ 不能省略)
- $/$: 對於整型數則為除法取整操作,例如， $5/2$ 得到結果2；
- 如果除數或被除數中有一個為負值，舍入方向不固定。例如， $-5/2$ ，有的系統中得到的結果為-2，在有的系統中則得到結果為-3。
- 對於浮點數則為通常意義的除法,例如， $5.0/2.0$ 得到結果為2.5
- 由此可見， $/$ 操作符可以對不同的數據類型進行不同的操作

實驗原理

3. 運算符和表達式

(3) 自增、自減運算符與表達式

● 運算符(2個)：++、--

- 功能：使變量的值加 1 或減 1
 - 前綴 ++i、--i：在使用 i 之前，先使 i 的值加(減)1
 - 後綴 i++、i--：在使用 i 之後，再使 i 的值加(減)1
 - 例如：int p; int i=3; p=++i; //i 的值：4，p 的值：4
 - int q; int j=3; q=j++; //j 的值：4，q 的值：3
- 說明：++、-- 只適合於變量，不能用於常量，5++、a++b、(a++b)++ 都是錯誤的

前綴與後綴自增(減)運算符的比較：

- ◆ 對自增(減)的作用是相同的，均能夠增加1(減少1)；
- ◆ 對當中的取值是有影響的，涉及先取值，還後取值問題。

實驗原理

3. 運算符和表達式

例5 前綴與後綴自減運算符(--)的比較

```
#include <stdio.h>
int main()
{
    int a=10;
    printf("a=%d, 執行b=--a;語句後:",a);
    int b=--a;           //相當於a=a-1;b=a;
    printf("a=%d, b=%d\n\n",a,b);

    printf("a=%d, 執行c=a--;語句後:",a);
    int c=a--;           //相當於c=a;a=a-1;
    printf("a=%d, c=%d\n\n",a,c);
    return 0;
}
```

a=10, 執行b=--a;語句後:a=9, b=9

a=9, 執行c=a--;語句後:a=8, c=9

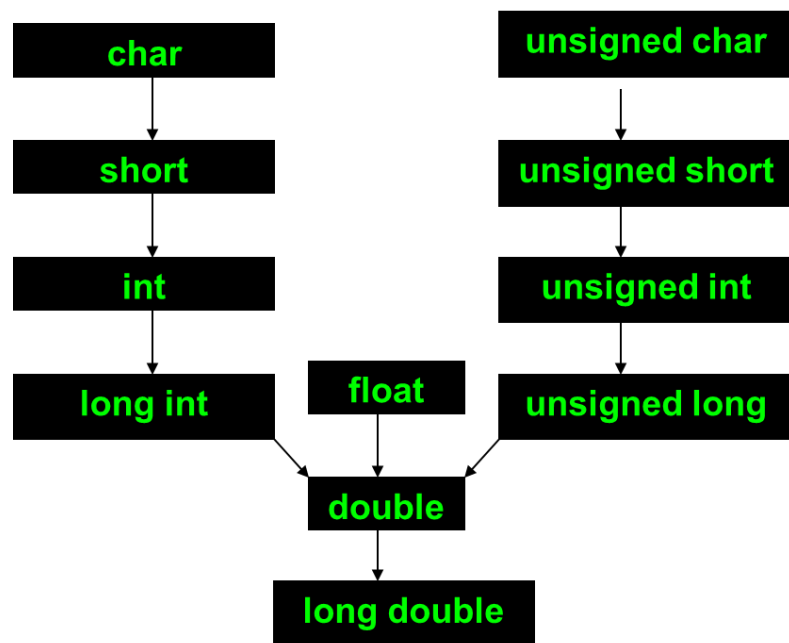
- 說明：若用自增、自減運算符構成表達式，則越簡單越好，最好是類似“a++”或“++a”這種表達式，以增強代碼的可讀性

實驗原理

3. 運算符和表達式

(4) 數據類型轉換

- 有兩種不同方式：
- 隱式轉換(自動轉換)：不同類型數據進行混合運算，會將兩個操作數自動作適當的類型轉換，然後進行運算



實驗原理

3. 運算符和表達式

(4) 數據類型轉換

- `+`、`-`、`*`、`/` 運算的兩個數中有一個數為float或double型，結果是double型。系統將float型數據都先轉換為double型，然後進行運算；
- 如果int型與float或double型數據進行運算，先把int型和float型數據轉換為double型，然後進行運算，結果是double型；
- char型數據與int型數據進行運算，就是把字符的ASCII代碼與整型數據進行運算
- 顯式轉換(強制轉換)：
 - 格式：類型名 (表達式) 或 (類型名) 表達式 或 (類型名) (表達式)
 - 例如：`(double)a` //將a轉換成double類型
 - `(int) (x+y)` //將x+y的值轉換成int型
 - `(float)(5%3)` //將5%3的值轉換成float型



實驗內容

1. 上機C語言 online IDE，例如
(https://www.onlinegdb.com/online_c_compiler) 輸入長方形的長和寬，輸出其周長、面積。
 - 可以參考本節裏面的例2編寫方法
 - 代碼截圖在實驗報告裏，并截圖輸出結果。





實驗內容

2. 上機C語言 online IDE，例如
(https://www.onlinegdb.com/online_c_compiler) 輸入**正方形**的邊長，輸出其周長、面積。
- 可以參考本節裏面的例2編寫方法
 - 代碼截圖在實驗報告裏，并截圖輸出結果。





思考題

思考題請回答并寫在實驗書面報告的：“六、實驗分析”。

1. 如果你想用C語言做程序設計，計算圓形的面積，應該選擇定義什麼數據類型比較合適，并給出具體的解釋。
2. 如果你想用C語言做程序設計，輸出一段文字，例如：Hello World，應該選擇定義什麼數據類型比較合適，并給出具體的解釋。






實驗要求

- 按要求完成解釋回答問題，記錄實驗結果
- 熟悉和掌握實驗內容的操作
- 總結實驗結果和結論
- 完成實驗報告





注意事項

- #本次實驗報告6-1，6-2合并提交。
 - #本次實驗報告6-1，6-2合并提交。
 - #本次實驗報告6-1，6-2合并提交。
 - 實驗報告模板裏的：三、實驗器材，請簡短的描述自己使用的計算機信息和编程IDE的信息。
 - 實驗報告模板裏的：四、實驗步驟，請描述你操作的具體過程，例如打開什麼設置，能看到什麼，然後怎麼操作，最後怎麼看到實驗結果。
 - 實驗報告模板裏的：五、實驗數據及處理，請描述你的相關實驗結果，并回答實驗內容裏面提到的問題。
 - 思考題請回答并寫在實驗書面報告的：“六、實驗分析”。
- 



感謝觀賞

Thank you for listening.