

二進位檔案 I/O



使用二進位模式開啟檔案，在寫入或讀取檔案時並不會發生字元轉換，數值在記憶體中的位元是如何，寫入檔案時就是如何，而讀入時也是相同。

下面這個程式可以讀入任意檔案，每次讀入一個位元組，並將讀入的資料以16進位數顯示，若讀入的資料前導位元為1，為了輸出的對齊，則使用其補數加以顯示：

```
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

int main(int argc, char* argv[]) {
    char ch;
    int count = 0;

    if(argc != 2) {
        cout << "指令: read <filename>" << endl;
        return 1;
    }
```

```
ifstream fin(argv[1], ios::in | ios::binary);
if(!fin) {
    cout << "無法讀取檔案" << endl;
    return 1;
}

while(!fin.eof()) {
    fin.get(ch);
    if(ch < 0)
        ch = ~ch; // 負數取補數
    cout << setw(2) << hex << (int)ch << " ";
    count++;
    if(count > 16) { // 換行
        cout << endl;
        count = 0;
    }
}
cout << endl;

fin.close();

return 0;
}
```

執行結果：

```
read main.cpp
```

```
23 69 6e 63 6c 75 64 65 20 3c 69 6f 73 74 72 65 61
```

```
6d 3e a 23 69 6e 63 6c 75 64 65 20 3c 66 73 74 72
65 61 6d 3e a 23 69 6e 63 6c 75 64 65 20 3c 69 6f
6d 61 6e 69 70 3e a 75 73 69 6e 67 20 6e 61 6d 65
73 70 61 63 65 20 73 74 64 3b a a 69 6e 74 20 6d
61 69 6e 28 69 6e 74 20 61 72 67 63 2c 20 63 68 61
略....
```

下面這個程式可以讓您將任意的檔案複製至另一個指定的名稱：

```
#include <iostream>
#include <fstream>
using namespace std;

int main(int argc, char* argv[]) {
    char ch;

    if(argc != 3) {
        cout << "指令: copy <input> <output>" << endl;
        return 1;
    }

    ifstream fin(argv[1], ios::in | ios::binary);
    ofstream fout(argv[2], ios::out | ios::binary);

    if(!fin) {
        cout << "檔案讀入失敗" << endl;
        return 1;
    }
}
```

```
if(!fout) {
    cout << "檔案輸出失敗" << endl;
    return 1;
}

while(!fin.eof()) {
    fin.get(ch);
    if(!fin.eof())
        fout.put(ch);
}

fin.close();
fout.close();

return 0;
}
```

在寫入或讀取檔案時，我們也可以用`read()`與`write()`函式以區塊的方式寫入，它們的函式雛型如下：

```
istream &read(char *buf, streamsize num);
```

```
ostream &write(const char* buf, streamsize num);
```

其中`num`是所要寫入的資料位元組數目，通常使用`read()`或`write()`函式時，都會使用二進位模式來存取檔案；下面這個程式則示範如何將陣列資料寫入檔案，然後再將之讀出：

```
#include <iostream>
#include <fstream>
using namespace std;
```





PayPal
Web
Payment

PayPal

SoI
com
Study

```
int main(int argc, char* argv[]) {
    ofstream fout("temp", ios::out | ios::binary);
    if(!fout) {
        cout << "無法讀取檔案" << endl;
        return 1;
    }

    int arr[5] = {1, 2, 3, 4, 5};
    fout.write((char*) arr, sizeof(arr));
    fout.close();

    ifstream fin("temp", ios::in | ios::binary);
    if(!fin) {
        cout << "無法讀取檔案" << endl;
        return 1;
    }

    fin.read((char*) arr, sizeof(arr));
    cout << "arr: ";
    for(int i = 0; i < 5; i++)
        cout << arr[i] << ' ';
    cout << endl;

    fin.close();

    return 0;
}
```

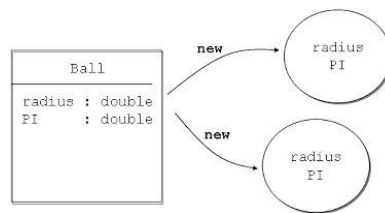
執行結果：

arr: 1 2 3 4 5





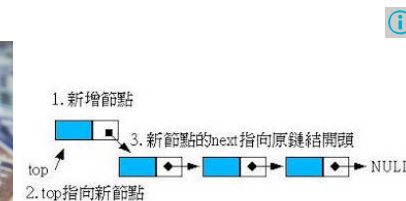
《黑色沙漠》產包1+1



static 類別成員



比較各大銀行真實信貸額度



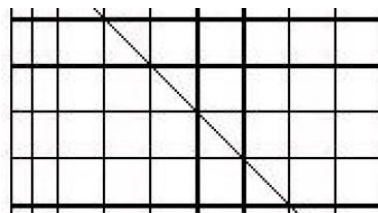
堆疊- 使用鏈結實作 (C 語言動態記憶體宣告)

廣告 BLACK DESERT



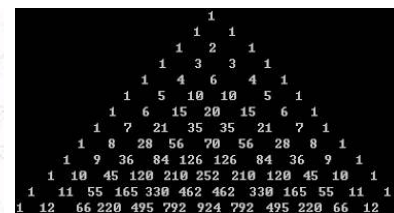
Google Cloud 免費試用

openhome.cc



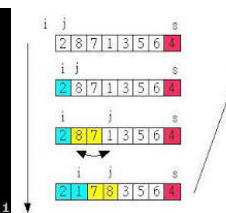
八個皇后

廣告 AlphaLoan實貸比較網



巴斯卡三角形

openhome.cc



快速排序法 (三

廣告 Google Cloud Platform

openhome.cc

openhome.cc

openhome.cc