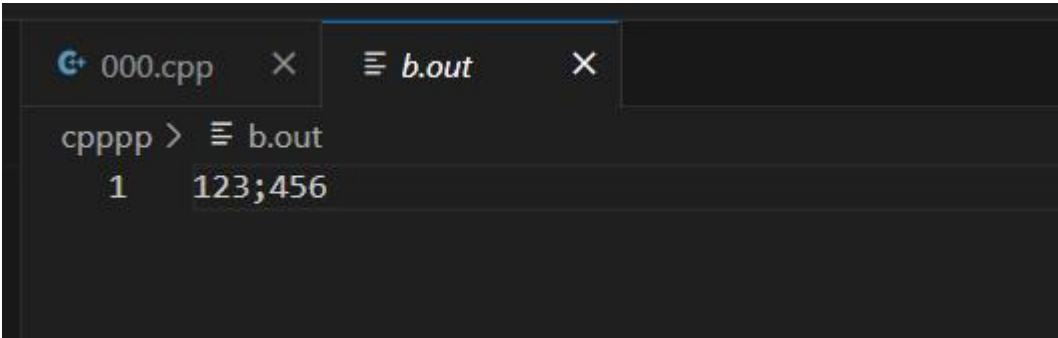



# 计算机学院\_高级语言程序设计\_课程实验报告

实验题目：流类库与输入输出流		学号：202200130048
日期：2023.5.16	班级：6	姓名：陈静雯
Email：1205037094@qq.com		
<p>实验步骤与内容：</p> <ol style="list-style-type: none"><li>1. 练习流类库中常用的类及成员函数的用法。</li><li>2. 练习标准输入/输出及格式控制</li><li>3. 练习文件的操作（文本文件、二进制文件）</li></ol>		
<p>结论分析与体会：</p> <p>1.</p>  <pre>cpppp &gt; ≡ b.out 1 123;456</pre> <p>2.</p>  <pre>Zoot      1 Jimmy     4e+01 Al        7e+02 Stan      4e+03 PS D:\code_repository\codes&gt;  Zoot      1.23 Jimmy     35.36 Al        653.7 Stan     124358.2 PS D:\code_repository\codes&gt;</pre>		

```

prmpdml.has --dbgexe=D:\mingw64\bin\gdb.exe --interp
Zoot      1.23
Jimmy     35.36
Al        653.7
Stan      4358.24
PS D:\code_repository\codes>

```

解释：setprecision 表示保留几位有效数字，setw 是控制输出字符宽度

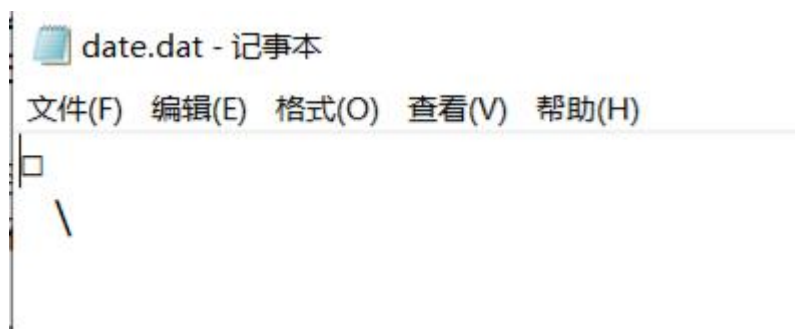
```

yremzow.vad --dbgexe=D:\mingw64\bin\gdb.exe --interp
Zoot      1.23
Jimmy     35.36
Al        653.7
Stan      124358
PS D:\code_repository\codes>

```

?? Double “124358.24” 输出原数字只输出 124358 六位

3.



```

G+ 000.cpp  ≡ date.dat  X
cpppp > ≡ date.dat
1  ACKNULNULNUL
2  NULNULNUL \NULNULNUL

```

二进制文件打开

00000000 06 00 00 00 0A 00 00 00 5C 00 00 00 \ - - - - -

4.

```
PS D:\code repository\code> .\04job.exe --dbgExe=D:\mingw64\bin\gdb.exe --interpreter=
```

5.

```
Type a line terminated by 't'
hello worldt
hello world
PS D:\code repository\code>
```

```
Type a line terminated by 't'
123djfnrj rfjetmsdj t
123djfnrj rfje
PS D:\code repository\code>
```

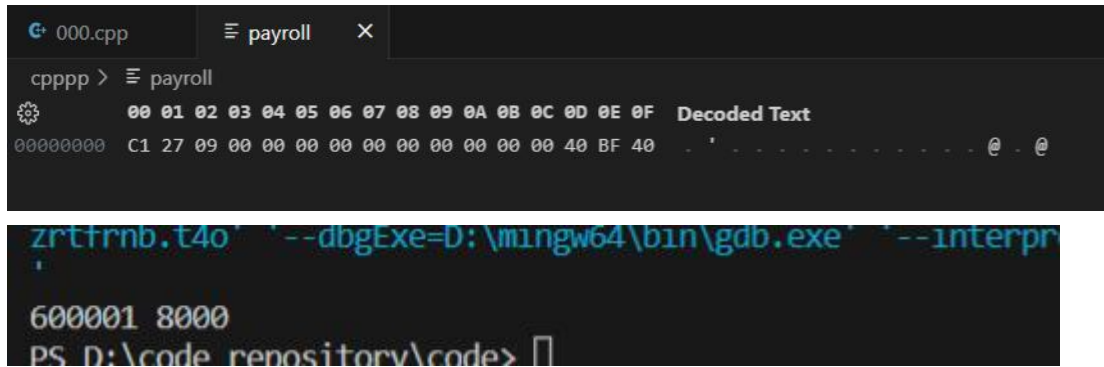
解释: `getline(cin, line, 't');` 读取键盘输入存到 `line` 中, 遇到字符 `t` 停止读取

Get 函数

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    char line[20];
    cout << "Type a line terminated by 't' " << endl;
    cin.get(line, 10, 't');
    cout << line << endl;
    return 0;
}
```

```
Type a line terminated by 't'
12 dj sitdjew
12 dj si
PS D:\code repository\code>
```

6.



The screenshot shows a debugger window with two tabs: '000.cpp' and 'payroll'. The 'payroll' tab is active, displaying a memory dump and assembly code. The memory dump shows a sequence of bytes: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F, with a 'Decoded Text' column showing a series of dashes. Below the memory dump, assembly code is visible, including instructions like 'zrtfrnb.t4o' and '600001 8000'. The window title bar shows 'PS D:\code repository\code> |'.

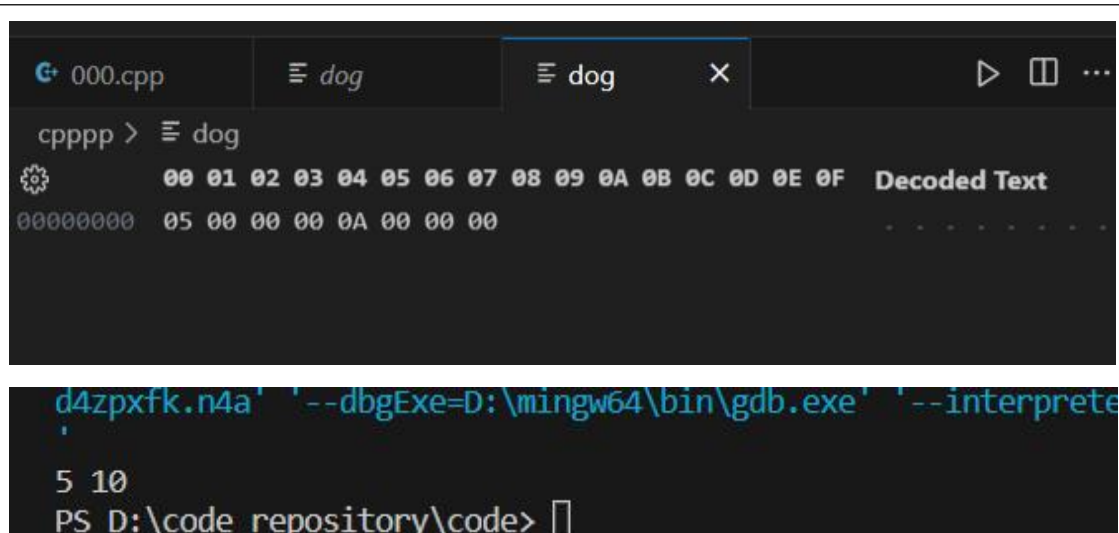
7.

二进制

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

class dog{
public:
    int weight;
    int age;
    dog(int x, int y):weight(x), age(y) {}
    dog() {}
    ~dog() {};
};

int main() {
    dog dog1(5,10);
    ofstream os("dog", ios_base::out| ios_base::binary);
    os.write(reinterpret_cast<char *>(&dog1), sizeof(dog1));
    os.close();
    ifstream is("dog", ios_base::in| ios_base::binary);
    if (is) {
        dog dog2;
        is.read(reinterpret_cast<char *>(&dog2), sizeof(dog2));
        cout << dog2.weight << " " << dog2.age << endl;
    } else {
        cout << "ERROR: Cannot open file 'dog'." << endl;
    }
    is.close();
    return 0;
}
```



The screenshot shows a debugger window with a memory dump and a command prompt. The memory dump is for address 00000000 and shows the value 05 00 00 00 0A 00 00 00. The command prompt shows the command 'd4zpxfk.n4a' and the output '5 10'.

```
00000000  05 00 00 00 0A 00 00 00
```


```
d4zpxfk.n4a' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interprete  
'  
5 10  
PS D:\code repository\code>
```

十进制

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

class dog{
public:
    int weight;
    int age;
    dog(int x,int y):weight(x),age(y) {}
    dog() {}
    ~dog() {};
};

int main() {
    dog dog1(5,10);
    ofstream os("dog");
    os.write(reinterpret_cast<char *>(&dog1), sizeof(dog1));
    os.close();
    ifstream is("dog");
    if (is) {
        dog dog2;
        is.read(reinterpret_cast<char *>(&dog2), sizeof(dog2));
        cout << dog2.weight << " " << dog2.age << endl;
    } else {
        cout << "ERROR: Cannot open file 'dog'." << endl;
    }
    is.close();
    return 0;
}
```



```
000.cpp dog dog X
cdddd > dog
00000000 05 00 00 00 0D 0A 00 00 00 Decoded Text
d4zpxfk.n4a' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interprete
5 10
PS D:\code repository\code>
```

```
9.
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;

int main() {
    ifstream is("11-9.txt");
    string s;
    ofstream os;
    if (is) {
        for(int i=1;i<=5;i++) {
            os.write(reinterpret_cast<char *>(&i), sizeof(i));
        }
        is.read(reinterpret_cast<char *>(&s), sizeof(s));
        os.write(reinterpret_cast<char *>(&s), sizeof(s));
    } else {
        cout << "ERROR: Cannot open file 'dog'." << endl;
    }
    is.close();
    return 0;
}
```