《运算符与表达式》课堂笔记

1. 新的输出语句`printf()`

1.1 `printf()`的语法

```
printf("格式控制符", 变量名);
```

2. 格式化输出

2.1 输出固定宽度的整数

示例：

03:对齐输出

总时间限制: 1000ms　内存限制: 65536kB

描述

读入三个整数，按每个整数占8个字符的宽度，右对齐输出它们。

输入

只有一行，包含三个整数，整数之间以一个空格分开。

输出

只有一行，按照格式要求依次输出三个整数，之间以一个空格分开。

样例输入

```
123456789 0 -1
```

样例输出

```
123456789        0       -1
```

`cout`语句解法：

```cpp
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
  // 输入三个整数
  int a, b, c;
  cin >> a >> b >> c;

  // 以8个字符宽度右对齐输出这三个整数
  cout << setw(8) << a << ' ' << setw(8) << b << ' ' <<  setw(8)  << c;
  return 0;
}
```

- 由于本次输出有格式要求，需要在输出变量之前加上格式函数`setw(n)`，n代表以多少个字符宽度进行输出，且输出的数字是右对齐的。
- 格式函数`setw(n)`只对后面输出的第一个变量有格式控制效果，所以后续输出的第二、三个变量都要在输出前分别写一遍。
- 使用格式函数`setw(n)`需要增加一个头文件`<iomanip>`

`printf()`语句解法：

```cpp
#include <iostream>
using namespace std;
int main(){
  // 输入三个整数
  int a, b, c;
  cin >> a >> b >> c;

  // 以8个字符宽度右对齐输出这三个整数
  printf("%8d %8d %8d", a, b, c);
  return 0;
}
```

- `printf()`是C语言库中自带的函数，所以不需要另外增加头文件。
- 输出函数`printf()`的格式为：`printf("格式控制符", 变量名);`
- 输出整数的格式控制符为`%d`，输出宽度为n的整数的格式控制符为`%md`，m代表以多少个字符宽度进行输出，且`%md`输出的数字是右对齐的。

2.2 输出保留n位小数的浮点数

示例：

05:输出保留12位小数的浮点数

总时间限制: 1000ms　内存限制: 65536kB

描述

读入一个双精度浮点数，保留12位小数，输出这个浮点数。

输入

只有一行，一个双精度浮点数。

输出

也只有一行，保留12位小数的浮点数。

样例输入

```
3.1415926535798932
```

样例输出

```
3.141592653580
```

`cout`语句解法：

```
#include <iostream>
#include <iomanip>
using namespace std;
int main(){
  // 输入一个浮点数
  double a;
  cin >> a;

  // 输出保留12位小数的浮点数
  cout << fixed << setprecision(12) << a;
  return 0;
}
```

`printf()`语句解法：

```cpp
#include <iostream>
using namespace std;
int main(){
  // 输入一个浮点数
  double a;
  cin >> a;

  // 输出保留12位小数的浮点数
  printf("%.12f", a);
  return 0;
}
```