

一、数学库常用函数

【大纲内容】

【难度 3】绝对值函数、四舍五入函数、下取整函数、上取整函数、平方根函数、常用三角函数、对数函数、指数函数

在 C++ 中使用数学函数，需要先引入数学库头文件 `<cmath>` (C Math Library)：

```
#include <cmath>
```

(一) 绝对值函数 (Absolute Value)

绝对值 (Absolute Value) 函数返回一个数去掉正负号后的值。

英文函数名：**abs(x)** (整数或浮点数) 或 **fabs(x)** (浮点数)

“无论你向前走 5 步还是向后退 5 步，距离都是 5 步”。

示例代码：

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    cout << abs(-7) << endl; // 输出 7
    cout << abs(-7.1) << endl; // 输出 7.1
    cout << fabs(-3.5) << endl; // 输出 3.5
    return 0;
}
```

易错点：

- `abs()` 用于整数或小数，`fabs()` 用于小数。

- 不能写 `abs a`, 要加括号 `abs(a)`。

(二) 四舍五入函数 (Round)

- 四舍五入 (Round) 函数会把小数**四舍五入**到最接近的整数。
- 函数名: **round(x)**
- 你买糖果花了 **3.6 元**, 妈妈说: “就给 **4 元** 的零花钱吧! ”

示例代码:

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    cout << round(3.4) << endl; // 输出 3
    cout << round(3.5) << endl; // 输出 4
    cout << round(3.6) << endl; // 输出 4
    cout << round(-3.4) << endl; // 输出-3
    cout << round(-3.6) << endl; // 输出-4
    return 0;
}
```

易错点:

- `round(3.5)` 会向上取整变成 `4`;
- `round(-3.5)` 会变成 `-4` (不是 `-3`! 注意负数四舍五入方向)。

(三) 下取整函数 (Floor)

- 下取整 (Floor) 函数是把小数变成**不大于它的最大整数**。
- 函数名: **floor(x)**
- 你身高 `165.8` 厘米, 只能进 “`165cm 以下`” 的游乐园项目, 你就被算成了 `165`。

示例代码：

```
cout << floor(3.9) << endl; // 输出 3
cout << floor(-2.1) << endl; // 输出 -3
```

易错点：

- 负数下取整会向“更小”的整数靠，比如 $-2.1 \rightarrow -3$ ；
- 不要误以为是去掉小数点，实际上方向不同于 `int()` 强制转换。

(四) 上取整函数 (Ceiling)

- 上取整 (`Ceil`) 函数会把小数变成**大于等于它的最小整数**。
- 函数名：`ceil(x)`
- 你要交 3.1 元买票，但售票机只收整数，它会收你 4 元。

示例代码：

```
cout << ceil(2.3) << endl; // 输出 3
cout << ceil(-2.3) << endl; // 输出 -2
```

易错点：

- 不等于 `round()`, `ceil(2.1)` 总是向上；
- 对负数来说是“向 0 靠近”。

(五) 平方根函数 (Square Root)

- **平方根 (Square Root)** 函数**返回一个数的正平方根**。
- 函数名：`sqrt(x)`
- 什么数乘自己等于 9？答案是 3，因为 $3 \times 3 = 9$ 。

示例代码：

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

```

int main() {

    cout << sqrt(64) << endl;    // 输出 8

    cout << sqrt(196) << endl;    // 输出 14

    cout << sqrt(3) << endl;    // 输出 1.73205

    return 0;

}

```

易错点：

- `sqrt()` 的参数**不能是负数（会出错）**；
- 如果你传入 `sqrt(-4)`，程序会返回 `Nan`（不是一个数）。

第 1 题

以下 C++ 代码的输出结果是什么？

```

#include <iostream>

#include <cmath>

using namespace std;

int main() {

    cout << abs(-12) << endl;

    return 0;

}

```

- A. -12
 B. 12
 C. 0
 D. 编译错误

答案与解析：

正确答案是 B。`abs()` 函数用于计算整数的绝对值，-12 的绝对值是 12。注意 `abs()` 可以处理整数和浮点数，但整数版本在 `<cmath>` 或 `<cstdlib>` 中均可用。

第 2 题

以下代码段中，调用 `fabs(-8.3)` 的返回值是：

- A. -8.3
- B. 8
- C. 8.3
- D. -8

答案与解析：

正确答案是 C。`fabs()` 专门用于浮点数的绝对值计算，返回 8.3。注意 `fabs()` 与 `abs()` 的区别在于参数类型，但 `abs()` 也可用于浮点数（C++11 起）。

第 3 题

`round(4.7)` 的返回值是：

- A. 4
- B. 5
- C. 4.0
- D. 5.0

答案与解析：

正确答案是 B。`round()` 函数对小数四舍五入到最接近的整数，4.7 离 5 更近，因此返回 5（整数类型）。

第 4 题

`round(-2.5)` 的返回值是：

- A. -2
- B. -3
- C. 2.5
- D. -2.0

答案与解析：

正确答案是 B。负数四舍五入时，-2.5 会向更小的整数 -3 舍入，而非 -2。这是易错点，需注意与正数情况的区别。

第 5 题

`floor(5.9)` 的返回值是：

- A. 5
- B. 6
- C. 5.0
- D. 6.0

答案与解析：

正确答案是 A。`floor()` 函数向下取整，返回不大于原数的最大整数，5.9 向下取整为 5。

第 6 题

`floor(-3.2)` 的返回值是：

- A. -3
- B. -4
- C. -3.0
- D. -4.0

答案与解析：

正确答案是 B。负数向下取整会向更小的整数方向取整，-3.2 向下取整为 -4，而非 -3。

第 7 题

`ceil(6.1)` 的返回值是：

- A. 6
- B. 7
- C. 6.0
- D. 7.0

答案与解析：

正确答案是 B。`ceil()` 函数向上取整，返回不小于原数的最小整数，6.1 向上取整为 7。

第 8 题

以下代码的输出结果是什么？

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
```

```

cout << sqrt(144) << endl;

return 0;

}

```

- A. 12
- B. -12
- C. 14
- D. 编译错误

答案与解析:

正确答案是 A。sqrt() 计算平方根，返回正数根。 144 的平方根是 12 。注意参数不能为负数，否则返回 NaN。

(六) 常用三角函数 (Trigonometric Functions)

中文名称	英文函数	说明	示例
正弦	$\sin(x)$	角度 x 的对边比斜边	$\sin(0) = 0$
余弦	$\cos(x)$	邻边比斜边	$\cos(0) = 1$
正切	$\tan(x)$	对边比邻边	$\tan(0) = 0$

- 注意：C++ 里的三角函数使用的是弧度制 (radian) 而不是角度！

示例代码：

```

#include <cmath>

#define PI 3.14159

double angle = PI / 2; // 90 度

cout << sin(angle) << endl; // 输出 1

```

易错点：

- $\sin(90) \neq 1$ ，因为 90 是角度，要换成弧度；

- 转换公式：角度 $\times \pi \div 180$ 。

(七) 对数函数 (Logarithm)

函数	功能	示例
$\log(x)$	自然对数，底数为 e	$\log(e) = 1$
$\log_{10}(x)$	以 10 为底的对数	$\log_{10}(100) = 2$

- 对数就是“你要乘几次才得到这个数？”
- 比如： $10 \times 10 = 100 \rightarrow$ 就是 $\log_{10}(100) = 2$

☒ 示例代码：

```
cout << log(2.71828) << endl; // 输出约 1
cout << log10(1000) << endl; // 输出 3
```

易错点：

- $\log()$ 是自然对数，不是 $\log_{10}()$ ；
- 不可以传入 0 或负数，程序会出错。

(八) 指数函数 (Exponential Function)

函数	说明	示例
$\exp(x)$	返回 e 的 x 次方，即 e^x	$\exp(1) = e \approx 2.71828$
$\text{pow}(x, y)$	返回 x 的 y 次幂	$\text{pow}(2, 3) = 8$

类比说明：

- $\text{pow}(2, 3)$ 就是 $2 \times 2 \times 2 = 8$ ；
- $\exp(1)$ 就是 e 的 1 次方。

☒ 示例代码：

```
cout << exp(1) << endl; // 约等于 2.71828
```

```
cout << pow(3, 2) << endl; // 9
```

考试注意事项:

- `pow(x, y)` 两个参数都可以是整数或小数;
- 指数函数结果很容易溢出，数值过大要小心！

复习表格：常用数学函数总览

函数	中文名称	功能说明	示例
<code>abs(x)</code>	绝对值	去掉正负号	<code>abs(-3) = 3</code>
<code>round(x)</code>	四舍五入	取最近的整数	<code>round(3.6)=4</code>
<code>floor(x)</code>	下取整	向下变整数	<code>floor(3.9)=3</code>
<code>ceil(x)</code>	上取整	向上变整数	<code>ceil(3.1)=4</code>
<code>sqrt(x)</code>	平方根	求一个数的平方根	<code>sqrt(25)=5</code>
<code>sin(x)</code>	正弦	三角函数，参数是弧度	<code>sin(PI/2)=1</code>
<code>log(x)</code>	自然对数	e 为底	<code>log(e)=1</code>
<code>log10(x)</code>	常用对数	10 为底	<code>log10(100)=2</code>
<code>exp(x)</code>	指数	e 的 x 次方	<code>exp(1)=e</code>
<code>pow(x,y)</code>	幂函数	x 的 y 次方	<code>pow(2,3)=8</code>