

- 表达式语句的作用是执行表达式并丢弃掉求值结果，以分号结束
- 一个块block {} 就是一个作用域，不以分号结束
- switch语句的内部控制流:如果某个case标签匹配成功，则从该标签开始执行后面的所有标签
- while 循环语句适用的情况：1.不知道迭代的次数 2.想在循环结束时访问循环控制变量
- for循环语句的多重定义

```
for (init_statement; condition; express);
// init_statement 可以定义多个对象，但只能有一个声明语句，即变量基础类型要相同，如：
decltype(v.size()) i = 0, s2 = v.size();
```

- 范围for循环：

```
for (declaration : expression) {
    statement;
}
// expression可以是vector, string, 或{a,h,c}, 要求其能返回迭代器的begin和end
```

// 以下两个for循环是等价的

```
for (auto & r : c)
for (auto beg = c.begin(), end = c.end(); beg != end; ++beg)
```

// 范围for循环是用迭代器来执行的，expression是vector时，不能改变其元素数量，因为这可能会使end函数变得无效

- 跳转语句：

1. break : 终止近处的while, dowhile, for, switch
2. continue : 终止循环中的当前迭代，开始下一次的迭代
3. goto : 不建议使用

- try语句块和异常处理（包括检测和处理）

1. throw 表达式
2. try 语句块
3. 一套异常类

异常处理暂时跳过