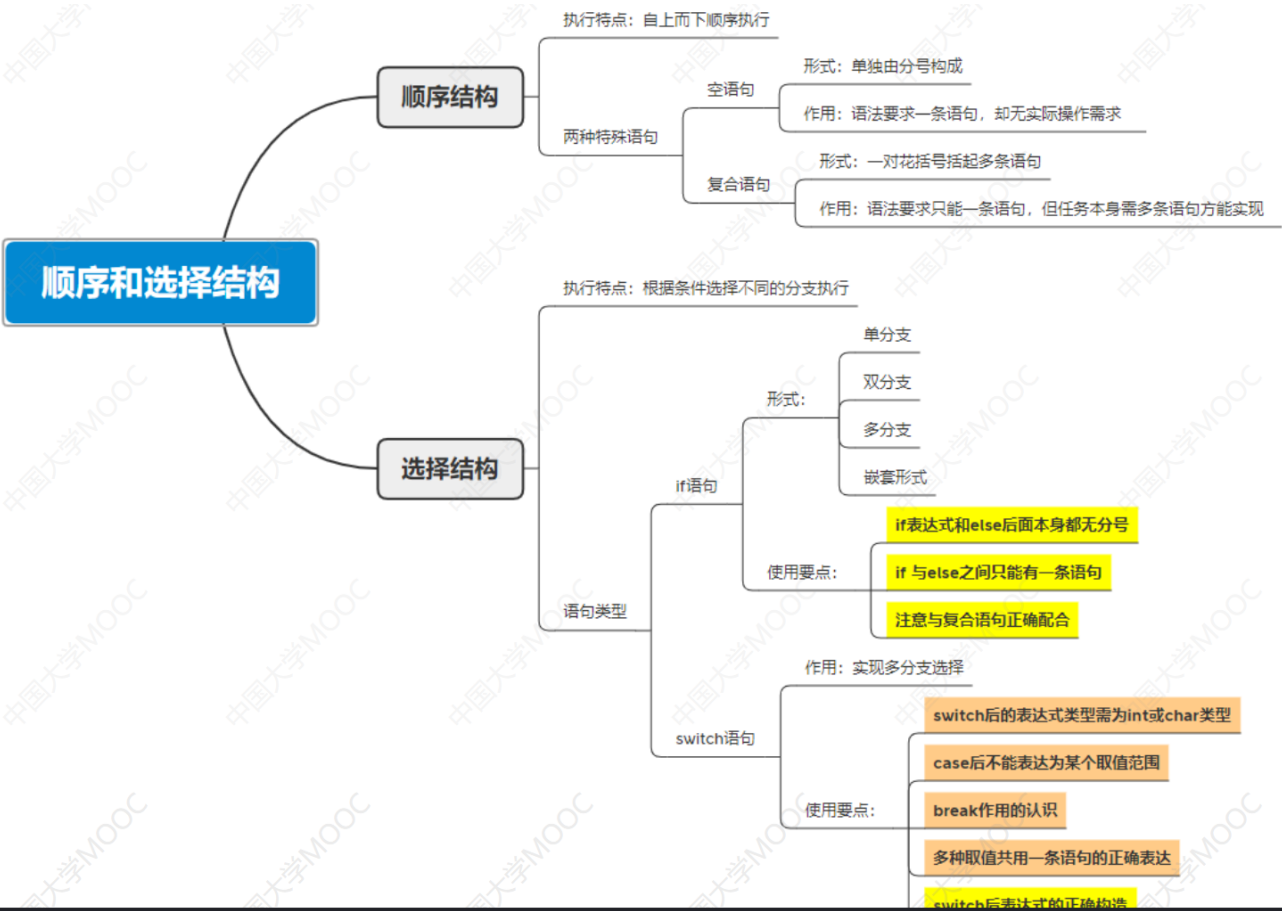



第三章测试



重点1

```
语句组1:
if(表达式1)
    if(表达式2) 语句1
else 语句2
```

```
语句组2:
if(表达式1)
{ if(表达式2) 语句1
else 语句2 }
```

 未使用大括号的，else与第一个if配对。使用大括号的，明确else与内部的if配对。

重点2

在switch语句中，break语句是语法要求还是逻辑要求？

switch(表达式)

```
{  
  case 常量表达式1: 语句组1; [break; ]  
  case 常量表达式2: 语句组2; [break; ]  
  .....  
  case 常量表达式n: 语句组n; [break; ]  
  [default: 语句组n+1]  
}
```

- switch后表达式类型须为**整型或字符型**;
- case后的常量表达式必须为可一一列举的具体取值, 不能是取值范围;
- 多个常量表达式可共用一组语句;



在switch语句中, break语句是逻辑要求, 因为它用于控制程序流, 避免落入下一个case的代码块中。如果一个case执行后没有break语句, 程序将继续执行后续的case语句, 这被称为“fall-through”。这种行为在某些情况下是有意为之, 但通常情况下, 使用break语句可以清晰地划分每个case的执行范围, 从而提高代码的可读性和可维护性。

因此, 虽然在语法上不强制要求使用break, 但从逻辑结构和程序设计的角度来看, 使用break是明智的选择。

1 单选 (2分) 若给定条件表达式 $(M) ? (a++) : (a--)$, 则其中表达式M()。

得分/总分

- ☐ A. 和 $(M==1)$ 等价
- ☒ B. 和 $(M!=0)$ 等价
- ☐ C. 和 $(M==0)$ 等价
- ☐ D. 和 $(M!=1)$ 等价

✓2.00/2.00

正确答案: B 你选对了

提问



在条件表达式 $(M) ? (a++) : (a--)$ 中, 当 M 为真时 (即 M 不为 0), 会执行 $a++$, 当 M 为假时 (即 M 为 0), 会执行 $a--$ 。

因此, 可以总结如下:

- 如果 M 不等于 0, 条件为真, 执行 $a++$ 。
- 如果 M 等于 0, 条件为假, 执行 $a--$ 。

2 单选 (2分) 关于switch语句描述正确的是()。

得分/总分

- ☒ A. switch语句中default子句可能没有, 也可能有一个
- ☐ B. switch语句中的case后面可以是一个关系表达式
- ☐ C. switch语句的每个语句序列中必须有break语句
- ☐ D. switch语句中switch后面的表达式可以是实型

✓2.00/2.00

正确答案: A 你选对了

提问

3 单选 (2分) 以下if语句语法正确的是()。

得分/总分

- ☐ A.

```
if(x>0)
{ x=x+y; cout<<x };
else
cout<<--x;
```
- ☒ B.

```
if(x>0)
{ x=x+y; cout<<x; }
else
cout<<--x;
```
- ☐ C.


```
if(x>0)
{ x=x+y; cout<<x; };
else
cout<<--x;
```
- ☐ D.

```
if(x>0)
cout<<x
else
cout<<--x;
```

✓2.00/2.00

正确答案: B 你选对了

提问

 分号写在里面, 外面不要! 而且两个分支都要写!

4 单选 (2分) 当 $a=1, b=2, c=4, d=3$ 时, 执行完下面一段程序后 x 的值是 ()。

得分/总分

```
if(a<b)
```

```
if(c<d) x=1;
```

```
else
```

```
if(a<c)
```

```
if(b<d) x=2;
```

```
else x=3;
```

```
else x=4;
```

```
else x=5;
```

☐ A. 4

☐ B. 3

☒ C. 2

✓2.00/2.00

☐ D. 1

正确答案: C 你选对了

提问

5 单选 (2分) 执行以下程序段后a的值为 ()

得分/总分

```
int a=0;

if (3&&2)

    a=1;

else

    a = 2;
```

☐ A. 2

☒ B. 程序有误

✗0.00/2.00

☐ C. 0

☐ D. 1

正确答案: D 你错选为B

提问



`if (3 && 2)`：这里使用了逻辑与运算符 `&&`。在C++中，非零值被视为真 (true)，零值被视为假 (false)。因此，3和2都是非零值，都被视为真。逻辑与运算符 `&&` 会返回两个操作数都为真时的真值，否则返回假。在这个例子中，3和2都是真，所以 `3 && 2` 的结果是真。

6 判断 (2分) 一个switch语句必须包含一个default语句。()

得分/总分

☒ A. ✗

✓2.00/2.00

☐ B. ✓

正确答案: A 你选对了

提问

7 判断 (2分) 在C程序中，if语句中不能嵌套switch语句。 ()

得分/总分

☐ A. ✓

☒ B. ✗

✓2.00/2.00

正确答案: B 你选对了

提问

8 判断 (2分) C语言中规定，if语句的嵌套结构中，else总是无条件地和最近的if配对。

得分/总分


☐ A. ✓

☒ B. ✗

✓2.00/2.00

正确答案: B 你选对了

提问

 这个重点1中讲过

9 填空 (2分) 假设有定义int a=2,b=-1,c=2; 则执行以下程序段后c的值为()。

得分/总分

```
if(a<b)
```

```
if(b<0) c=0;
```


```
else c++;
```

3

✗ 0.00/2.00

正确答案: 2

提问

 确实是直接执行了c++，但是这个是先赋值然后再加

10 填空 (2分) 假设有定义int m = 5; 则执行以下程序段的输出结果是()

得分/总分

```
if(m++>5) cout<<m;
```

```
else cout<<m--;
```

6

✓ 2.00/2.00

正确答案：6

提问