2019-20春夏OOP期末考试参考答案

阅读程序写结果

4-1

\1. H
\2. H
\3. i

4-2

\1. A()
\2. A()
\3. B()
\4. ~B()
\5. ~A()
\6. ~A()

注：注意父类和子类构造和析构的顺序。

4-3

\1. 9
\2. 9

4-4

\1. A
\2. A
\3. B
\4. A
\5. C

4-5

\1. i
\2. d

4-6

\1. B::F(double)
\2. B::F(double)
\3. A::F2(int)

注：本题并未体现面向对象程序设计的多态性，而仅仅是本身就调用对应的函数。

4-7

\1. B
\2. 1

注：本题有一定难度，题解可参考下面这个帖子：

https://www.cc98.org/topic/5354330

4-8

\1. 5.7
\2. 5

4-9

\1. 1

注：本题体现int和const int是不同的数据类型；若将主函数中变量i的类型改为const int或int const则答案应为2。

4-10

\1. operator[](int)const
\2. operator[](int)

注：看似不太符合常理，但实际上只要能想到等号从右往左输出就可以做出来了；另外主要掌握const写在返回值类型前和函数体前的区别，前者意思是不能修改返回值，后者意思是不能更改成员变量。建议在Linux环境下验证该程序：

```
clang++ 4-10.cpp -std=c++98 （但其实C++98、C++11、C++17跑出来都是一样的）

./a.out
```

4-11

\1. f(double)

注：原因在于const A和A不同；但如果将void f(int)改为void f(int) const答案就是f(int)了。

4-12

\1. FAIL
\2. OK2
\3. OK3
\4. OK4

注：当且仅当某个数据类型能被隐式转换成为与该表达式相同类型时，static_cast才成立；换句话说，只要“大类型”相同，就能使用static_cast了，即使它在实际运行过程中可能是不安全的，因此父类和子类之间的上/下转换static_cast都能成立。但dynamic_cast则会在运行时进行检查，故上转换不一定能够成立。

4-13

\1. 5---4
\2. 5---6

4-14

\1. A()
\2. A()
\3. operator=(const A&)
\4. A(const A &)

注：这题一定要掌握，其知识点可认为是整门课程的核心：构造函数、拷贝构造函数和=运算符的重载。如果有问题，可以参考南方科技大学于仕琪老师的慕课：

https://www.bilibili.com/video/BV1Vf4y1P7pq

4-15

\1. I 'm in Derived
\2. I 'm in Derived
\3. Destructing Derived
\4. Destructing Base

注：这种输出题还是直接复制吧，当年这题因为空格死了不少人。

完善程序

5-1

注：许威威老师说本题只需要考虑能否过编译，所以填的答案只要能过编译大概就能认为自己做对了。但也正因为如此，第9和第10两空不能**不填**。

\1. m_pNext=&n
\2. strlen(s)+1
\3. delete [] m_s
\4. Node *t=p->m_pNext
\5. p=t
\6. &n
\7. const
\8. p=p->m_pNext
\9. new
\10. new

我觉得9 10应该是*new

5-2

\1. this->numerator=numerator
\2. this->denominator=denominator
\3. const Fraction &
\4. denominator

\5. operator
\6. Fraction
\7. const Fraction &f1
\8. denominator*f1.numerator
\9. friend
\10. Fraction((numerator*f1.denominator+denominator*f1.numerator),
(denominator*f1.denominator))

类设计

答案略。