指出每行输出结果

1. #include <iostream>
2. using namespace std;
3. struct A
4. {
5. A() { cout << "A"; }
6. };
7. struct B : A
8. {
9. A a;
10. B() : A() { cout << "B"; }
11. };
12. struct C : virtual A
13. {
14. C() : A() { cout << "C"; }
15. };
16. struct D : virtual A, C
17. {
18. C c;
19. D() : A(), C() { cout << "D"; }
20. };
21. struct E : virtual B, virtual C
22. {
23. E() : B(), C() { cout << "E"; }
24. };
25. struct F : virtual B, D, virtual E
26. {
27. D d;
28. E e;
29. F() : B(), D(), E() { cout << "F"; }
30. };
31. int main()
32. {
33. A a;
34. cout << "\n"; *// 输出：*
35. B b;
36. cout << "\n"; *// 输出：*
37. C c;
38. cout << "\n"; *// 输出：*
39. D d;
40. cout << "\n"; *// 输出：*
41. E e;
42. cout << "\n"; *// 输出：*
43. F f;
44. cout << "\n"; *// 输出：*
45. }

答：输出为(空格是为了阅读方便，实际输出不带空格)

A

AA B

A C

AC AC D

AAB AC E

AAB ACE CACD ACACD AABACE F

解释：

1. A没有父类，直接输出A，完成A的构造
2. 首先构造B的父类A，然后构造B的成员A a，最后输出B，完成B的构造
3. C虚继承A，先构造A，然后输出C，完成C的构造
4. D虚继承A，先构造A，D同时继承C，再构造C(此时不必再重新构造A了),接着构造数据成员c(输出AC),最后完成D的构造函数，输出D，完成D的构造
5. E虚继承B,C，先构造B，输出AAB，然后构造C，输出AC，最后输出E，完成E的构造
6. F虚继承B，E，**先构造B**，输出AAB,后构造E，构造E要**构造C**(输出AC),然后输出E完成**E的构造**，然后**构造D**，D要继承c，输出C（虚基类A已经构造），然后构造D的数据成员c输出AC（另开一棵树），最后输出D完成D的构造。

接着，完成E的两个数据成员的构造，结果同前。

一个用于理解的**并不规范**的示意图

