



main.cpp



Run

Output

Clear



```
1 // Online C++ compiler to run C++ program online
2 #include <iostream>
3 using namespace std;
4
5 class A {
6 public:
7     void displayA() {
8         cout << "This is class A" << endl;
9     }
10 };
11
12 class B : public A {
13 public:
14     void displayB() {
15         cout << "This is class B, derived from
class A" << endl;
16     }
17 };
18
19 class C : public A {
20 public:
21     void displayC() {
22         cout << "This is class C, derived from
class A" << endl;
23     }
24 };
25
26 class D : public A {
27 public:
28     void displayD() {
29         cout << "This is class D, derived from
class A" << endl;
30     }
31 };
32
33 class E : public B {
34 public:
35     void displayE() {
36         cout << "This is class E, derived from
class B (which in turn is derived
from class A)" << endl;
37     }
38 };
39
40 int main() {
41     A objA;
42     B objB;
43     C objC;
44     D objD;
45     E objE;
46
47     cout << "Calling display functions for each
class:" << endl;
48
49     objA.displayA();
50
51     objB.displayA();
52     objB.displayB();
53
54     objC.displayA();
55     objC.displayC();
56
57     objD.displayA();
58     objD.displayD();
59
60     objE.displayA();
61     objE.displayB();
62     objE.displayE();
63
64     return 0;
65 }
66
67
```

```
/tmp/6Xc5rh0N2m.o
Calling display functions for each class:
This is class A
This is class A
This is class B, derived from class A
This is class A
This is class C, derived from class A
This is class A
This is class D, derived from class A
This is class A
This is class B, derived from class A
This is class E, derived from class B (which in turn
is derived from class A)
```

```
=== Code Execution Successful ===
```