```
#include <iostream>    // inh3n.cpp
using namespace std;

class b { long l;
  public: b(long l=0):l(l){cout<<"Konstruktor b\n";}
          virtual ~b(){ cout<<"Destruktor b"<<endl;}
          long get_l(){ return l;}
};

class d : public b {
          long l;
  public: long p;
          d(long lb=0, long ld=0, long p=0):
          b(lb),l(ld),p(p){cout<<"Konstruktor d\n";}
          ~d(){cout<<"Destruktor d"<<endl;}
          long get_l(){ return l;}
          long get_bl(){ return b::get_l();}
};

void main(){
 d d1(1,3,5);
 b *b1=&d1, &b2=d1, b3(d1), //moegliche Zuweisungen
 b4=(b)d1;    // cast, auch: b4=static_cast<b>(d1);

 cout<<"b1->l = "<<b1->get_l()<<endl;            // 1

 cout<<"d1.p  = "<<((d *)b1)->p<<endl;           // 5

 cout<<"d1.p  = "<<static_cast<d *>(b1)->p<<endl;// 5

 cout<<"b1->l = "<<((d *)b1)->get_bl()<<endl;    // 1
 cout<<"d1.l  = "<<((d *)b1)->get_l()<<endl;     // 3

 cout<<"d1.p = "<<((d &)b2).p<<endl;             // 5
 cout<<"b2.l = "<<((d &)b2).get_bl()<<endl;      // 1
 cout<<"d1.l = "<<((d &)b2).get_l()<<endl;       // 3

//hier hilft selbst ein cast nicht mehr weiter:

//cout<<"b3.l="<<((d)b3).get_bl()<<endl;//Compile Err.
//cout<<"b4.p="<<b4.p<<endl; //Compile Err., no member

 b *b5=new b(7);
 cout<<"b5->l = "<<((d *)b5)->get_bl()<<endl; // 7

 cout<<"b5->p = "<<((d *)b5)->p<<endl;        //Zufall
 cout<<"(d*)b5->l="<<((d *)b5)->get_l()<<endl;//Zufall
 delete b5; b5=0; cin.ignore();
}
```