```
// Adapted from Reference: Bjarne Stroustrup. 2014. Programming: Principles and Practice Using C++ (2nd. ed.).
// Addison-Wesley Professional, pg. 644-645.
#include <iostream>
#include <memory>
#include <vector>
using namespace std;

struct X {
  int val;
  void out(const string& s, int nv) {
      cerr << this << "->" << s << ":" << val << "(" << nv << ")\n";
  }

  X(){out("X()",0); val = 0;}  // default constructor
  X(int v) :val(v) {out("X(int)",v);}

  X(const X& x) :val(x.val) {out("X(X&)",x.val);}   // copy constructor

  X& operator=(const X&a) {    // copy assignment operator
    out("X::operator=()", a.val); val=a.val; return *this;
  }

  ~X() {out("~X()",0);}  // destructor
  };

  X glob(2); // Global variable
  X copy(X a) { return a;}
  X copy2(X a) { X aa = a;  return aa;}
```

```
X& ref_to(X& a) {return a;}

unique_ptr<X> make(int i) {X a(i); return make_unique<X>(a);}

struct XX {X a; X b;};

// Trace what is output by main.
// What is printed to std error?  You can run it and see.
// What function is called by each statement?
int main() {
X loc{4};  // local variable
X loc2{loc}; // copy construction
loc = X{5}; // copy assignment
loc2 = copy(loc); // call by value and return;
loc2 = copy2(loc);
X loc3{6};
X& r = ref_to(loc);
unique_ptr<X> p1 = make_unique<X>(7);
p1.reset(); // delete the X from the heap
p1 = make_unique<X>(8);
p1.reset(); // delete the X from the heap
vector<X> v(4);
XX loc4;
p1 = make_unique<X>(9);  // create X on heap and then delete it
p1.reset();
unique_ptr<X[]> p2 = make_unique<X[]>(5); //create array of X on heap and delete
p2.reset();
}
```

Trace output

0x604214->X(int):2(2)

0x7fff773a9fa0->X(int):4(4)

0x7fff773a9fb0->X(X&):4(4)

0x7fff773a9fc0->X(int):5(5)

0x7fff773a9fa0->X::operator=():4(5)

0x7fff773a9fc0->~X():5(0)

0x7fff773a9fd0->X(X&):5(5)

0x7fff773a9fe0->X(X&):5(5)

0x7fff773a9fb0->X::operator=():4(5)

0x7fff773a9fe0->~X():5(0)

0x7fff773a9fd0->~X():5(0)

0x7fff773a9ff0->X(X&):5(5)

0x7fff773aa000->X(X&):5(5)

0x7fff773a9fb0->X::operator=():5(5)

0x7fff773aa000->~X():5(0)

0x7fff773a9ff0->~X():5(0)

0x7fff773aa010->X(int):6(6)

0x1a3fc20->X(int):7(7)

0x1a3fc20->~X():7(0)

0x1a3fc20->X(int):8(8)

0x1a3fc20->~X():8(0)

0x1a3fc20->X():0(0)

0x1a3fc24->X():0(0)

0x1a3fc28->X():0(0)

0x1a3fc2c->X():0(0)

0x7fff773aa040->X():2000331088(0)

0x7fff773aa044->X():32767(0)

0x1a3fc40->X(int):9(9)

0x1a3fc40->~X():9(0)

0x1a3fc68->X():0(0)

0x1a3fc6c->X():0(0)

0x1a3fc70->X():0(0)

0x1a3fc74->X():0(0)

0x1a3fc78->X():0(0)

0x1a3fc78->~X():0(0)

0x1a3fc74->~X():0(0)

0x1a3fc70->~X():0(0)

0x1a3fc6c->~X():0(0)

0x1a3fc68->~X():0(0)

0x7fff773aa044->~X():0(0)

0x7fff773aa040->~X():0(0)

0x1a3fc20->~X():0(0)

0x1a3fc24->~X():0(0)

0x1a3fc28->~X():0(0)

0x1a3fc2c->~X():0(0)

0x7fff773aa010->~X():6(0)

0x7fff773a9fb0->~X():5(0)

0x7fff773a9fa0->~X():5(0)

0x604214->~X():2(0)