```
OOP244 Quiz 7
#include <iostream>
using namespace std;
class DblContainer {
   double *valAdd_;
public:
   DblContainer() {
      valAdd_ = new double;
   }
   DblContainer(double value) {
      valAdd_ = new double(value);
   }
   DblContainer(const DblContainer& D) {
      valAdd_ = new double(*D.valAdd_);
   }
   DblContainer& operator=(const DblContainer& D) {
      if (this != &D) {
         *valAdd_ = *D.valAdd_;
      }
      return *this;
   }
   ~DblContainer() {
      delete valAdd_;
   }
   ostream& display(ostream& os)const {
      return os << (*valAdd_);
   }
   istream& read(istream& is) {
      return is >> (*valAdd_);
   }
   double& operator=(const double& Ref) {
      *valAdd_ = Ref;
      return *valAdd_;
   }
};
ostream& operator<<(ostream& os, const DblContainer& Dbl) {
   return Dbl.display(os);
}
istream& operator >> (istream& is, DblContainer& Dbl) {
   return Dbl.read(is);
}
```

The above class is a double container that keeps a single double value dynamically and give other programmers access to read print and access its double value.
Convert the above class to a template and rename it to “Container” so it can be used to keep any object and not only a double.
Also after converting explain for what capabilities an object must have to be able to be held by container.

Container template example:

```
int main() {
   Container<double> D = 2.34;
   cout << D << endl;
   Container<Employee> E;
   cin >> E;
   cout << E << endl;
   return 0;
```

```
}
```

Submit your answer as follows:

Create a file called quiz7.txt and answer the question in it.
Submit that file with this command:

~fardad.soleimanloo/submit oop_q7