

Name: (as it would appear on official course roster)		
Email address:		@umail.ucsb.edu
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")		

1
h08
CS24 W19

h08: Chapter 6, sections 6.1-6.6

ready?	assigned	due	points
true	Wed 02/27 02:00PM	Mon 03/04 09:00AM	20

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the lowest scores (if you have zeros, those are the lowest scores.)

Complete your reading of Chapter 6, section 6.1 -6.2, 6.3 (ignore references to multiset of page 139), 6.4-6.6 (If you don't have a copy of the textbook yet, there is one on reserve at the library under "COMP000-STAFF - Permanent Reserve").

1. (10 pts) Write a template function named `isEqual()` that compares two items and returns a boolean. If the items are equal the function should return true otherwise it should return false. The function has two parameters of the same type. The type may be any type that has a copy constructor and the `==` operator defined.

```
template<class Item>
bool isEqual(Item i1, Item i2) {
    return (i1 == i2);
}
```

Please:

- No Staples.
- No Paperclips.
- No folded down corners.

2. (10 pts) Re-read pages 314-315. Then consider the Sequence class provided at the link <https://github.com/ucsb-cs24-w18/hw5/blob/master/sequence.h>. The Sequence class currently stores a list of words in a static array. Convert this class to a template class that can be used to store a collection of any datatype instead of just `std::string`. Write the new definition of the Sequence class below. You don't have to implement the methods.

```
template<class Item>
class Sequence {
public:
    Sequence();
    void append(Item item);
    Item& operator[](unsigned int index);
    unsigned int size() const;
    unsigned int getCapacity() const;
```

Item operator [] (unsigned int index) const;
private.

Item data [10];

unsigned int used;

unsigned int capacity;

};

2

h08

CS24 W19