**Midterm**

1. **Array**

**Class A**

**{**

**Float info[200]; // the largest valid array index is 199 (not 200)**

**Public:**

**A() { info[200] = 0; // not ok**

**For( int i=0; i< 200; i++)**

**Info[i] = 0;**

**}**

**X[20][30]**

**Float sum=0;**

**For(int i=0; i<20; i++)**

**For(int j=0; j<30; j++)**

**{**

**Sum += x[i][j];**

**}**

**Z[20][30][40]**

**For(int i=0; i<20; i++)**

**For(int j=0; j<30; j++)**

**For(int k=0; k<40; k++)**

**{**

**Sum += z[i][j][k];**

**}**

1. **Struct**

**Struct B**

**{**

**Int id;**

**String name;**

**Void foo() { id++; }**

**};**

1. **Class**

**Post-fix ++**

1. **Create a temp object**
2. **Store the current values to this temp object**
3. **Do ++ operation**
4. **Return this temp object (pass by value)**

**Class C**

**{**

**Public:**

**C& operator ++() // pre-fix ++ operator**

**{**

**// do ++ operation**

**…**

**Return \*this;**

**}**

**C operator ++(int) // post-fix ++ operator**

**{**

1. **Create a temp object**
2. **Store the current values to this temp object**
3. **Do ++ operation**
4. **Return this temp object (pass by value)**

**}**

**Pre-fix ++**

1. **Do ++ operation**
2. **Return \*this;**

**Search:**

1. **Unsorted list: one-way traversal via a for loop (linear search)**
2. **Sorted list: besides linear search, you can also use a binary search**

**Binary Search:**

**Basic idea: Divide & Conquer**

**Template:**

**Void foo(int x)**

**{ …. }**

**Void foo(float y)**

**{**

**….**

**}**

**Void foo(Student z)**

**{**

**….**

**}**

**template <class T> // T is a generic data type**

**void foo2( T x) // template function**

**{**

**….**

**}**

**Int main( )**

**{**

**Int a=10;**

**Foo2(a);**

**Float b = 3.14;**

**Foo2(b);**

**Student c;**

**Foo2(c);**

**}**