$Id: cmps109-2015q2-exam1.mm,v 1.42 2015-04-21 15:15:09-07 - - $

page 1 page 2 page 3 Total / 32

Please print clearly :

Name :

Login : @ucsc.edu

Code only in C++11. No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Neatness counts ! Do your scratch work elsewhere and enter only your final answer into the spaces provided.

1. Given `vector<string> v;` — You may use `auto`.
   (a) Using the colon (foreach) form of the for loop, write out each element of the vector, one item per line. **[1✔]**

   (b) Using iterators explicitly, using the two-semicolon form of the for loop, write out each element of the vector, one item per line. **[1✔]**

2. Finish the function `sum`, which sums all of the elements in the range given by a pair of forward iterators. Assume the iterators point at `double`s. **[2✔]**

```
template <typename Iterator>
double sum (Iterator begin, Iterator end) {
```

3. Define an `operator<<` so that the statements on the left will print out the word `false` or `true`, as appropriate. Assume the library does ***not*** have an operator which will do this. **[2✔]**

```
// Sample code
bool b;
cout << " b = " << b << endl;
```

```
// Operator definition
```

4. Write a `using` statement which defines a data type called `mathfn`, which is a pointer to a function whose argument is a `double` by value, and which returns a result of type `double`. Write a declaration for an `unordered_map` named `functions` whose `key_type`s are `string`s and whose `mapped_type`s are `mathfn`s. **[2✔]**

5. Write a loop (not a complete function or program) which will read `double`s from `cin` until end of file and put them in a `vector` in the same order that they appeared on an input file. Hint : `cin>>d` will read a `double` into the variable `d` and return true if it succeeded and false if not. **[2✔]**

```
vector<double> v;
double d;
```

6. Define a single constructor for class `complex` which takes zero, one, or two arguments. For zero arguments, both fields are initialized to 0.0. For a single argument, the real field is initialized to its value, and the imaginary field to 0.0. For two arguments, the real field's value is the first argument, and the imaginary field's value is the second argument. Define the ctor inline in the header. **[1✔]**

```
class complex:
   private:
      double real;
      double imag;
   public:
      complex (
```

7. Complete `operator<` for a name. One name is less than the other if the last name is less than the other's last name. If the last names are equal, first names are compared. Your implementation may only use `string::operator<` as the only comparison operator. **[2✔]**

```
struct name {
   string last;
   string first;
   bool operator< (const name& that) {
```

8. Finish the definitions of the members of `vec` and `vec::iterator` as appropriate, making them all inline. The following code must work given your code :

   `vec<int,10> v; for (auto i = v.begin(); i != v.end(); ++i) f(*i);`

   Assume that `f` and any initializations have been done elsewhere. This is not a complete implementation. In order to be useable (outside the scope of this test), many more functions would have to be added.

   (a) Code the inline operators unary `operator*`, prefix unary `operator++`, and binary `operator!=` for the iterator. Also code the iterator constructor which takes a `T*` as an argument. **[4✔]**

   (b) Code the `vec` functions `begin` and `end`. **[1✔]**

```
template <typename T, size_t N>
struct vec {
   private: T data[N];
   public:
      class iterator {
         private: T* curr;
         public:
            // Code the ctor and operators *, ++, != here.









      };
      // Code begin and end here.

};
```

9. Write a function `smallest` which returns an iterator pointing at the smallest element of a range given by a pair of iterators. Assume the elements have `operator<` defined. If there is more than one "smallest" element, return the first one. Return what would normally be expected if the range is empty. **[2✔]**

```
template <typename Iterator>
Iterator smallest (Iterator begin, Iterator end) {
```

Multiple choice. To the ***left*** of each question, write the letter that indicates your answer. Write ***Z*** if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[12✔]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	12		$= c$

1. The last line of `operator=` should be :
   (A) `return *auto;`
   (B) `return *this;`
   (C) `return nullptr;`
   (D) `return this;`

2. Which C++11 standard library class uses reference counting to manage memory ?
   (A) `T*`
   (B) `auto_ptr<T>`
   (C) `shared_ptr<T>`
   (D) `unique_ptr<T>`

3. For a raw vector `int a[10];`, what is the equivalent to `a.end()`, if `int[]` were a class ?
   (A) `&a[0]`
   (B) `&a[10]`
   (C) `&a[11]`
   (D) `&a[9]`

4. Which of the following does not need to move objects in the container when adding an arbitrary number of new elements, but still can access individual elements in $O(1)$ time ?
   (A) `deque`
   (B) `list`
   (C) `map`
   (D) `vector`

5. Which of the following containers uses a balanced binary search tree ?
   (A) `deque`
   (B) `forward_list`
   (C) `map`
   (D) `unordered_map`

6. Given `string s;` and `string t;` which will compare their addresses, not values ?
   (A) `&&s == &&t`
   (B) `&s == &t`
   (C) `*s == *t`
   (D) `s == t`

7. Inside class `foo`, a move constructor's prototype is :
   (A) `foo (const foo&&);`
   (B) `foo (const foo&);`
   (C) `foo (foo&&);`
   (D) `foo (foo&);`

8. What declares the postfix `operator++` as a non-member of class `foo` ?
   (A) `foo operator++ ();`
   (B) `foo operator++ (foo&);`
   (C) `foo operator++ (foo&, int);`
   (D) `foo operator++ (int);`

9. After the library includes in a C++ program, what statement is usually used ?
   (A) `#include <namespace/std>`
   (B) `explicit namespace std;`
   (C) `import namespace std;`
   (D) `using namespace std;`

10. A constructor may be used implicitly as an automatic conversion operator. To prevent this, what keyword is used ?
    (A) `default`
    (B) `delete`
    (C) `explicit`
    (D) `implicit`

11. In the following, what statement is executed immediately after `continue` ?

```
for (i = 0; i < n; ++i) {
   f(); continue; g();
} h();
```

    (A) `++i`
    (B) `g()`
    (C) `h()`
    (D) `i<n`

12. Which of the following operators is most usually declared as a friend, not as a member ?
    (A) `operator+`
    (B) `operator+=`
    (C) `operator<`
    (D) `operator<<`