$Id: cmps109-2014q2-exam2.mm,v 1.24 2014-05-13 19:17:22-07 - - $

page 1 page 2 page 3 Total/32

Please print clearly:

Name:

Login: @ucsc.edu

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. Code the function `merge`. It takes a pair of iterators bounding one sorted input range, and another pair of iterators bounding another sorted input range. It has an iterator pointing at an output range, which is assumed to be large enough. It also takes a comparison function. It copies elements from the input containers into the output container so that it is sorted. Example:

   ```
   merge (a.begin(), a.end(), b.begin(), b.end(), c.begin(), less<int>());
   ```

   will merge `vector<int>a` and `vector<int>b` into `vector<int>c` using `less<int>`. **[4✔]**

   ```
   template <typename iter, typename oiter, typename Less>
   void merge (iter b1, iter e1, iter b2, iter e2, oiter ob, Less less) {
   ```

2. Define a complex number as a `struct` so that all members are public. It contains the following:
   (i) Two `double` fields, `real` and `imag`, whose default values are explicitly set to 0.0. **[1✔]**
   (ii) A constructor which automatically converts a `double` to a complex number with a 0.0 imaginary part. **[1✔]**
   (iii) A constructor which accepts two `doubles`. **[1✔]**
   (iv) The member function binary `operator+`. **[1✔]**
   (v) The member function binary `operator+=`. **[1✔]**
   (vi) The non-member function `operator<<`, which will write out a number, as in, for example, `(3.8,4.2i)`, i.e., both parts of the number in parentheses as shown here. **[1✔]**

   Code all functions as inline functions as they might appear in a header file.

3. Write a function `execute` which will take a map argument and a string argument, and look up the function in the map. If the function is found, call it. If not, print an error message. **[2✔]**

```
typedef map<string, void(*)()> funmap;
void execute (const funmap& map, const string& key) {
```

4. Write the function `accumulate` which takes a pair of iterators, an identity element, and a function, and uses the function to accumulate a result. Example:

```
   int s = accumulate (v.begin(), v.end(), 0, add);
   int p = accumulate (v.begin(), v.end(), 1, mul);
```

will find the sum and product of all elements of `v`. **[2✔]**

```
template <typename T, typename itor>
T accumulate (itor begin, itor end, T identity, T(*f)(T,T)) {
```

5. Write a function which accepts an `ostream` and a pair of iterators and prints out each element specified by the range, with a space between each element, but not before the first or after the last. **[2✔]**

```
template <typename iterator>
void print (ostream& out, iterator begin, iterator end) {
```

6. Write a function called `filter` that takes a `vector<string>` as an argument and returns another vector containing only those elements for which the predicate is true. **[2✔]**

```
typedef bool (*predicate) (const string&);
vector<string> filter (const vector<string>& v, predicate p) {
```

7. Given the beginning of a program as shown here, write code to declare `vector<string>` args which is initialized to the elements of `argv` from 1 to `argc - 1`, i.e., all of the command line arguments other than the program name. **[2✔]**

```
int main (int argc, char** argv) {
```

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, which count 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. Which of the following will permit the fastest search for a key?
   (A) `list`
   (B) `map`
   (C) `unordered_map`
   (D) `vector`

2. The class `shared_ptr` makes use of what method of memory management?
   (A) Copying collection with semispaces.
   (B) Dangling pointers.
   (C) Mark and sweep garbage collection.
   (D) Reference counting.

3. In an object-oriented hierarchy, how is a function marked in such a way as to require that it must be overridden in a derived class?
   (A) `= 0`
   (B) `= abstract`
   (C) `= delete`
   (D) `= virtual`

4. What kind of cast can be used to move the bits in a pointer to the bits in a `unsigned long`?
   (A) `const_cast`
   (B) `dynamic_cast`
   (C) `reinterpret_cast`
   (D) `static_cast`

5. What is the type of the expression `sizeof(vector<int>)`?
   (A) `float`
   (B) `int`
   (C) `int64_t`
   (D) `size_t`

6. In the absence of the keywords `private`, `protected`, or `public`, members of a `class` are __(x)__, and members of a `struct` are __(y)__.
   (A) (x) = private, (y) = private
   (B) (x) = private, (y) = public
   (C) (x) = public, (y) = private
   (D) (x) = public, (y) = public

7. Which data structure is specified as using a contiguous block of heap memory?
   (A) `deque`
   (B) `list`
   (C) `map`
   (D) `vector`

8. For `vector<int> v`:
   (A) `v.size()==v.end()+v.begin()`
   (B) `v.size()==v.end()-v.begin()+1`
   (C) `v.size()==v.end()-v.begin()-1`
   (D) `v.size()==v.end()-v.begin()`

9. Which concept is most closely related to object-oriented programming?
   (A) conversion
   (B) inheritance
   (C) overloading
   (D) templates

10. The idea of an iterator is based on a half-open interval $[a, b)$, which refers to what set?
    (A) $\{ x \mid a < x < b \}$
    (B) $\{ x \mid a < x \leq b \}$
    (C) $\{ x \mid a \leq x < b \}$
    (D) $\{ x \mid a \leq x \leq b \}$

11. After searching a map with the following statement, what statement can be used to print the value associated with the key?
    ```
    auto& i = m.find (key);
    ```
    (A) `cout << (*i)->second;`
    (B) `cout << i->second;`
    (C) `cout << i.second;`
    (D) `cout << i;`

12. If a base class `foo` is declared to have virtual functions and is involved in inheritance, what other member of the class must be virtual?
    (A) all other function members
    (B) copy constructor
    (C) destructor
    (D) `operator<<`