`$Id: cmps109-2013q1-exam1.mm,v 1.22 2013-01-30 21:32:51-08 - - $`

page 1 ○ page 2 ○ page 3 ○ Total / 31 ○

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. Write the function `filter`. It takes a pair of iterators values which indicate the bounds of an input vector, and a predicate which decides whether its argument is good or bad. Return a list containing a copy of all elements in the original range which are considered good. Note that both `list` and `vector` have a function `push_back`. **[2✔]**

```
typedef vector<string>::const_iterator citer;
typedef bool (*isgood) (const string &);
list<string> filter (const citer &begin, const citer &end, isgood p) {
```

2. Write a header file for class `queue`. Do not show anything that might appear in the implementation file. Show only those parts of the header file required here. Use proper judgement as to what is public and what is private.
   (a) It has a `head` and a `tail` pointer to a linked list of `node`s, which are private to the queue. Each node contains an `int` and a pointer to the next node. Define the head and tail pointers and the nested class node. **[2✔]**
   (b) Define the friend declaration and prototype for a shift (`<<`) operator which can be used to print it. **[2✔]**
   (c) Define `pop_front`, `front`, and `push_back` in a way consistent with STL conventions. **[2✔]**
   (d) Define the four members which would otherwise be supplied as default members. **[2✔]**

3. Using a typedef, define a type mathfn that is a pointer to a function with a single double argument and which returns a single double result. Declare a map with string keys and mathfn values. Use assignment statements to load the functions log, sqrt, and sin into the table with keys that spell their names. **[2✔]**

4. Referring to class queue from the previous page, code the function push_back as it would appear in the implementation (cpp) file. **[3✔]**

5. Write a function which will find the lexicographically least string in a vector of strings. Return an iterator pointing at that string. If the vector is empty, return the end iterator. If the smallest string occurs more than once, you may return an iterator pointing at any of them. **[3✔]**

```
vector<string>::iterator least (
   vector<string>::const_iterator begin,
   vector<string>::const_iterator end) {
```

6. Complete the following beginning of a program so that all of the command line arguments, but not the program name, are pushed onto a vector in the same order as they appear on the command line. **[2✔]**

```
int main (int argc, char **argv) {
   vector<string> argvec;
```

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. **[11✓]**

number of correct answers		× 1 =	$= a$
number of wrong answers		× ½ =	$= b$
number of missing answers		× 0 =	0
column total $c = \max(a - b, 0)$	11		$= c$

1. What is the way to stop an iterator loop for all iterators in the following **`for`** statement? What goes in the blank?
   `for (i = c.begin(); ____ ; ++i)`
   (A) `i == c.end()`
   (B) `i != c.end()`
   (C) `i < c.end()`
   (D) `i > c.end()`

2. If a vector has 10 elements, `v.end()` is effectively the address of:
   (A) `v[-1]`
   (B) `v[0]`
   (C) `v[9]`
   (D) `v[10]`

3. The most efficient way to increment an iterator:
   (A) `++i`
   (B) `i++`
   (C) `i=1+i`
   (D) `i=i+1`

4. Which is a copy constructor?
   (A) `foo (const foo &);`
   (B) `foo (foo &);`
   (C) `foo (const foo);`
   (D) `foo (foo);`

5. A search of a **`std::map`** is done in how much time?
   (A) $O(1)$
   (B) $O(\log_2 n)$
   (C) $O(n)$
   (D) $O(n \log_2 n)$

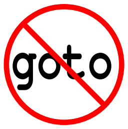
6. Given class **`foo`** has a field **`bar x`**, what is the most efficient way to write a constructor?
   (A) `foo (const bar &y) { x (y); }`
   (B) `foo (const bar &y) { x = y; }`
   (C) `foo (const bar &y): x (y) {}`
   (D) `foo (const bar &y): y = x {}`

7. A single-argument constructor that should not be available as an implicit conversion operator is preceded by what keyword?
   (A) **`delete`**
   (B) **`explicit`**
   (C) **`override`**
   (D) **`virtual`**

8. If **`p`** is a pointer or a direct-access iterator, and **`n`** is an integer, which one of the following operations is illegal?
   (A) `n - n`
   (B) `n - p`
   (C) `p - n`
   (D) `p - p`

9. In the absence of any other considerations, what is the most efficient way to pass an object of most classes?
   (A) `void f (foo_t);`
   (B) `void f (foo_t &);`
   (C) `void f (const foo_t);`
   (D) `void f (const foo_t &);`

10. Which of the following is the definition of a destructor?
    (A) `~foo(){}`
    (B) `foo~(){}`
    (C) `foo()~{}`
    (D) `foo(){}~`

11. 

    (A) Edsger Dijkstra
    (B) Donald Knuth
    (C) Bjarne Stroustrup
    (D) Andrew Tanenbaum