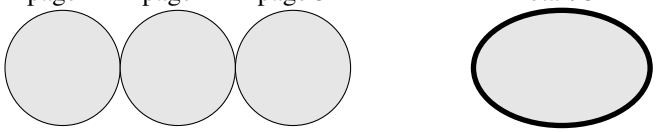
$Id: cmps109-2013q3-exam1.mm,v 1.37 2013-07-17 16:54:10-07 - - $

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. Given class `foo`, write out the prototypes of the four C++98 members that are automatically defined. **[2✔]**

```
class foo {
```

2. Given the declaration

```
vector<int> vec;
```

assume it was filled with data, write a `for`-loop which uses a `vector<int>::const_iterator` to print out all of the integers one per line. **[2✔]**

3. Write a complete function which will print out all of the key and value pairs in a `map<string,string>`. Print each pair, one per line, with the key first, then a space, then the value associated with that key. **[3✔]**

```
typedef map<string,string> mapss;
void print_mapss (const mapss &themap) {
```

4. Write a function to compute the inner product of two vectors of doubles. The inner product is the sum of pairwise products of correstponding elements of each vector. The mathematical formula for this is given in the box at left. Throw the exception `length_error` if the arrays have different lengths. **[3✔]**

$$p = \sum_{i=0}^{n-1} a_i b_i$$

```
double inner_prod (const vector<double> &a, const vector<double> &b) {
```

5. Write a function `list_to_vector`, which accepts two iterators over a list as arguments, and which copies all approved elements of the list into the vector, maintaining the same order. An approved element is one for which the predicate passed in as the third argument, when applied to an element of the list, returns true. Hint: append to a vector with `push_back`. **[3✔]**

```
typedef bool (*predicate) (const string &);
typedef list<string>::const_iterator list_citor;
vector<string> list_to_vector (const list_citor &begin,
                               const list_citor &end,
                               predicate approved) {
```

6. Write a program to read words from `cin`, and using a `map<string,int>`, count the number of times each word appears. At end of file, print out each word, one per line in lexicographic order, followed by the number of times the word occurred. If you declare `string word`, then the loop `while (cin >> word)` will skip white space and read in one word each time it is executed. The loop stops at end of file. **[4✔]**

```
#include <iostream>
#include <cstdlib>
using namespace std;
```

7. Define a class `intstack` which holds a stack of integers. It uses a single private field of type `vector<int>` and has no pointers in the class. Code all functions as inline functions that do nothing more than forward the operation to the vector. Implement the following functions: **[3✔]**
   (a) `push` pushes a new element onto the end of the vector. It uses vector's `push_back`.
   (b) `pop` removes an element from the end of the vector and returns it. It uses vector's `back` and `pop_back`. If `pop` is called on an empty stack, it throws an `out_of_range` exception.
   (c) `empty` returns a `bool` indicating the result of the vector's `empty` function.

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. Given the iterators over some container `c`, where `b=c.begin()` and `e=c.end()`, what will be true for an empty container ?
   (A) `b < e`
   (B) `b == e`
   (C) `b == e + 1`
   (D) `b == e - 1`

2. Given `i=c.begin()` what is the most efficient way to move the iterator `i` toward `c.end()` ?
   (A) `++i`
   (B) `--i`
   (C) `i++`
   (D) `i--`

3. 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) $\{x \mid a \leq x \leq b\}$

4. If `i` is a `map<string,string>::const_iterator`, returned by `map::find`, what expression will yield the value ?
   (A) `(*i)->second`
   (B) `i->*second`
   (C) `i->second`
   (D) `i.second`

5. Which of the following definitions of `f` will cause a copy of its argument of class `T` to be made ?
   (A) `void f (T &&x);`
   (B) `void f (T &x);`
   (C) `void f (T *x);`
   (D) `void f (T x);`

6. What keyword is used in a statement that will permit one class to see the private members of another class ?
   (A) `explicit`
   (B) `friend`
   (C) `inline`
   (D) `operator`

7. Which operator uses lazy evaluation ?
   (A) `++`
   (B) `<<`
   (C) `==`
   (D) `||`

8. What should be the first non-comment line in `foo.h` ?
   (A) `#define __FOO_H__`
   (B) `#ifndef __FOO_H__`
   (C) `#include "foo.h"`
   (D) `class foo {`

9. Given a class outlined as follows
```
class foo {
  private: string s;
  public: foo (const bar &);
};
```
   What is the most efficient way to implement a constructor ?
   (A) `foo::foo (const bar &x) { s = x; }`
   (B) `foo::foo (const bar &x): s(x) {}`
   (C) `void foo (const bar &x): s(x) {}`
   (D) `void foo::foo (const bar &x) { s = x; }`

10. For class `node`, what is the correct way to declare a pointer `p` to point at a `node`, and to allocate a node on the heap ?
    (A) `node &p = new node();`
    (B) `node *p = &new node[];`
    (C) `node *p = new node();`
    (D) `node p = new node();`

11. For `int a` and `int b` if we want to swap them using the call `swap(a,b)`, what is the prototype for `swap` ?
    (A) `void swap (auto a, auto b);`
    (B) `void swap (int &a, int &b);`
    (C) `void swap (int *a, int *b);`
    (D) `void swap (int a, int b);`