$Id: cmps109-2016q2-exam2.mm,v 1.46 2016-05-05 18:33:29-07 - - $

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 ! Points will be deducted for messy or unreadable answers. Do your scratch work elsewhere and enter only your final answer into the spaces provided.

1. Write the C++11 prototypes for class `foo` which will be implicitly generated unless otherwise specified. Show them as they would appear in a header file inside the class definition. Show prototypes only. Point allocation is given in the table at left. **[2✔]**

6 correct :	**2** ✔
5 correct :	**1½** ✔
4 correct :	**1** ✔
3 correct :	**½** ✔
else :	**0** ✔

2. Rewrite the `for`-statement using the two semi-colon version of a `for`-loop but not the colon. Explicitly use the iterator. **[1✔]**

```
for (auto& i: t) foo(i);
```

3. Using ***only*** `foo::operator<`, code the non-member `operator<=` as an inline function. **[1✔]**

4. Code the template function `copy` which can copy data from one container into another. Its two template arguments are : the type of the input iterators, and the type of the output iterator. Its three function arguments are the begin and end iterators for the source container and the begin iterator for the output container. The following statement will copy from `src` to `dst`, which need not be of the same type. **[3✔]**

```
copy (src.begin(), src.end(), dst.begin());
```

5. Given a class string outlined here, code the inline destructor. **[1✔]**

```
class zstring {
      size_t size_; char* data_;
   public:
      ~zstring() { ________________________________________________}
};
```

6. Code a function equivalent to the library `<algorithm>` function `for_each`. It has two template parameters : an iterator type, and a function type. It has three function parameters : a begin iterator, an end iterator, and a function. It applies the function to each object in the range. **[2✔]**

7. Code the function `print_map`. It has one template argument, which may be any type of `map`. It has one function argument, which may be any map. It prints out each entry in the map in lexicographic order, one pair per line: first the key, then an equal (`=`) sign, then the associated value, then the end of line. **[2✔]**

8. Write a function `differentiate` which performs symbolic differentiation on a polynomial. For example,
$$\frac{d}{dx}\, ax^3 + bx^2 + cx + d \quad = \quad 3ax^2 + 2bx + c$$
In other words, for each term in the sum of the form kx^n, the resulting derivative term is knx^{n-1}.
Represent the polynomial with the exponent as the subscript and the coefficient as the value. **[3✔]**
For example, $v = 5x^3 + 9x^2 + 8x + 10$ is represented as `polynomial v {10, 8, 9, 5};`
Its derivative $d = 15x^2 + 18x + 8$ is represented as `polynomial d {8, 18, 15};`

```
using polynomial = vector<double>;
polynomial differentiate (const polynomial& p) {
```

9. Write a function `zipwith`. It has two template parameters: an arbitrary element type, and a binary function type. It has three actual parameters: two vectors of the element type, passed in by constant reference, and a pointer to a binary function. Its result is a vector of the element type. Elements of the argument types are combined pairwise with the binary function to produce the output function. Throw a `domain_error` if the vectors have different lengths. For example,
```
   vector<int> v1{1,2,3}, v2{4,5,6}, v3;
   v3 = zipwith (v1, v2, plus<int>());
```
will set `v3` to `{5,7,9}`. **[3✔]**

```
template <typename T, typename Func>
vector<T> zipwith (const vector<T>& v1, const vector<T>& v2, Func f) {
```

10. Given class `complex` shown here, define public inline members so that no definition is needed in the `cpp` file:
    (a) Define a a single constructor that performs all of the following functions: It may be used to implicitly convert a `double` to a `complex`. It ensures that the default value is $0 + 0i$ if given with no arguments. If given with one argument, it sets the imaginary part to 0. It may be given with two arguments, the real part first, and the imaginary part second. **[1✔]**
    (b) Define a member `operator+`. Mathematics: $(a + bi) + (c + di) = ((a + c) + (b + d)i)$ **[1✔]**

```
class complex {
      double real;
      double imag;
   public:
```

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. If `p` is a pointer, what will increment whatever `p` points at ?
   (A) `*++p`
   (B) `*p++`
   (C) `++*p`
   (D) `++p*`

2. Which container exhibits the best locality of reference ?
   (A) `deque`
   (B) `forward_list`
   (C) `list`
   (D) `vector`

3. Which container exhibits the largest space overhead per data item ?
   (A) `deque`
   (B) `forward_list`
   (C) `list`
   (D) `vector`

4. What operator may be defined with any number of arguments as given by the situational requirements ?
   (A) `operator()`
   (B) `operator++`
   (C) `operator-=`
   (D) `operator->`

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

6. As a prototype in a header file inside the declaration of class `foo`, how is the postfix `operator++` declared ?
   (A) `foo operator++ (foo&, int);`
   (B) `foo operator++ (int);`
   (C) `foo& operator++ ();`
   (D) `foo& operator++ (foo&);`

7. If `m` is a map, and `k` is a key, after the following statement, how can the value associated with the key be accessed ?
   `auto& i = m.find(k);`
   (A) `i->first`
   (B) `i->second`
   (C) `i.first`
   (D) `i.second`

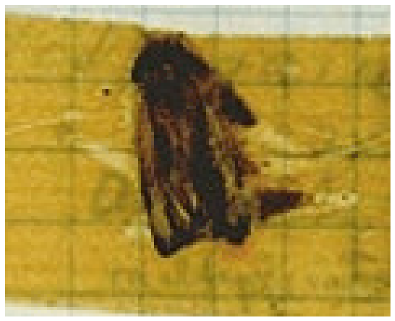
8. What prototype says that `foo` accepts an rvalue reference to a `bar` ?
   (A) `void foo (bar&&);`
   (B) `void foo (bar&);`
   (C) `void foo (bar);`
   (D) `void foo (bar*);`

9. Which file descriptor is associated with `cerr` and `stderr` ?
   (A) 0
   (B) 1
   (C) 2
   (D) 3

10. What windowing system does Linux use ?
    (A) W10
    (B) X11
    (C) Y12
    (D) Z13

11. What does the following statement do ?
    `cout << (0.0/0.0) << endl;`
    (A) Prints : `nan`
    (B) Prints : `0`
    (C) Prints : `inf`
    (D) Throws an exception.

12. An `unordered_map` is implemented by what kind of data structure ?
    (A) abstract syntax tree
    (B) doubly linked list
    (C) hash table
    (D) red-black tree



The First "Computer Bug". Moth found trapped between points at Relay #70, Panel F, of the Mark II Aiken Relay Calculator while it was being tested at Harvard University, 9 September 1947. The operators affixed the moth to the computer log, with the entry: "First actual case of bug being found." They put out the word that they had "debugged" the machine, thus introducing the term "debugging a computer program". In 1988, the log, with the moth still taped by the entry, was in the Naval Surface Warfare Center Computer Museum at Dahlgren, Virginia.