$Id: cmps109-2015q2-exam3.mm,v 1.57 2015-06-04 13:03:59-07 - - $

page 1	page 2	page 3	page 4	page 5	Total / 54

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.

For all answers, assume : `using namespace std;`

1. Write the function `purple_square`, which draws a square with corners at $(0,0)$, $(1,0)$, $(1,1)$, and $(0,1)$. Purple's `GLubyte` components are : red 160, green 32, blue 240. **[3✔]**

2. Write a function `accumulate` which has three template parameters : an iterator type, an element type, and a function type. It has four parameters : begin and end iterators, an identity element, and a combining function. For example,

```
s = accumulate (v.cbegin(), v.cend(), 0, plus<double>());
p = accumulate (v.cbegin(), v.cend(), 1, multiplies<double>());
```

will find the sum and product of the elements of `v`. **[3✔]**

3. Code a template class `stack`, writing all functions inline as they would appear in a header file. It contains a private `vector` and the following public functions, which contain forwarding to `vector` functions : `push`, `pop`, `top`, `size`, `empty`. Make sure to declare functions `const` when appropriate, and `top` should have both a constant and a non-constant version. **[4✔]**

4. Complete the following declarations of primary and secondary colors. Use either `0` or `255` in your answers. (Points: 0 if none or one correct; 1/2 if two or three correct; 1 if four correct.) **[1✔]**

`const GLubyte MAGENTA[] = { ___, ___, ___};`	`const GLubyte YELLOW[] = { ___, ___, ___};`
`const GLubyte BLACK[] = { ___, ___, ___};`	`const GLubyte WHITE[] = { ___, ___, ___};`

5. Write a function `fork_server` which when called, creates a child process. In the child process, call the function `run_server` (with no parameters). In the parent process, print the process id of the child. **[2✔]**

```
void fork_server() {

}
```

6. Given the function `void hello (const string&)` write a main function which: runs `hello` in a separate thread for each command line argument; puts each thread into a vector; waits for each thread to finish; then returns a success exit status. Do not bother to check for race conditions. **[2✔]**

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

}
```

7. Write a template function that takes a vector as an argument and returns a pair of vectors as its result. The first vector in the pair contains all of the elements of the argument with even subscripts and the second of the pair has all elements of the argument with odd subscripts. Example: splitting $\{1,2,3,4,5\}$ returns $\{\{1,3,5\},\{2,4\}\}$. **[3✔]**

```
template <typename T>
pair<vector<T>,vector<T>> split (const vector<T>& v) {
```

8. Write the prototypes for the implicitly generated members of class `foo` that were present in C++98. **[2✔]**

```
class foo {
```

9. Code a template function `print`, which takes three arguments : a begin iterator, an end iterator, and a string. It prints out all of the values indicated by the iterator range, each separated from the next by the string. The string is not printed before the first nor after the last element in the sequence. **[2✔]**

10. Code the linear search template function `find`, which takes a pair of iterators a the first two arguments, and an element as the third argument. It returns the first element in the sequence equal to the third argument. If not found, return the end iterator. Assume the elements have `operator==` defined. **[2✔]**

11. Code a template class `fixarray`, whose template parameters are an element type and a constant size, which must be specified at compile time. Do not code any of the implicitly generated member functions. Its private field is a raw array, not a vector. It should have `operator[]` in constant and non-constant versions, as well as the functions `begin` and `end`. Code the iterator as it would appear when declared inside the class, ahead of `begin` and `end`. Code all functions inline as they would appear in a header file. Your code should be sufficient to work with the following :

```
size_t constexpr N = 5; fixedarray<int,N> a; int n = 1;
for (size_t i = 0; i != N; ++i) { a[i] = ++n; n = a[i]; }
for (auto j = a.begin(); j != a.end(); ++j) cout << *j << endl;
```

   (a) Code for the class, ignoring the iterator. **[3✔]**

   (b) Code for the iterator. **[3✔]**

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. Threads that are dispatched via library functions, not the kernel, are called :
   (A) red threads
   (B) green threads
   (C) blue threads
   (D) white threads

2. What kind of iterator does the library class `list` provide ?
   (A) bidirectional
   (B) direct access
   (C) forward
   (D) input

3. What kind of protection allows access to fields of a class by members of the class and also by members of derived classes, but not by any other classes ?
   (A) `friend:`
   (B) `private:`
   (C) `protected:`
   (D) `public:`

4. Given a pointer `p` to an object and the call `p->f(x,y)`, what is the equivalent C code ?
   (A) `((*p).f)(x,y)`
   (B) `(p->class_table->f)(p,x,y)`
   (C) `(p->class_table->f)(x,y)`
   (D) `(p->f)(p,x,y)`

5. Which of the following data structures will likely have the worst locality of reference ?
   (A) `deque<int> d;`
   (B) `int a[100];`
   (C) `list<int> l;`
   (D) `vector<int> v;`

6. When a child process completes normally, and `int status` contains the result returned by the `wait`(2) system call, what expression will allow the parent to discover the argument previously returned by the child via the `exit`(2) system call ?
   (A) `status & 0x7F`
   (B) `status & 0x80`
   (C) `status << 8`
   (D) `status >> 8`

7. The following declaration will put the numbers 1 to 4 into which memory segment ?
   `vector<int> v {1, 2, 3, 4};`
   (A) init data
   (B) uninit data
   (C) heap
   (D) function call stack

8. Which operator may be declared with any number of arguments ?
   (A) `operator()`
   (B) `operator<>`
   (C) `operator[]`
   (D) `operator||`

9. A static variable is bound to an absolute (virtual) address at :
   (A) compile time.
   (B) link time.
   (C) exec time.
   (D) function call time.

10. For the declaration `map<K,V> x;`, the expression `x.begin()` returns an iterator that points at a :
    (A) `V`
    (B) `const K`
    (C) `pair<K,V>`
    (D) `pair<const K,V>`

11. Fields of a derived class are initialized in what order ?
    (A) base class, then fields in declaration order
    (B) base class, then fields in initializer-list order
    (C) fields in declaration order, then base class
    (D) fields in initializer-list order, then base class

12. In which version of C++ was `<thread>` first included in the standard library ?
    (A) C++98
    (B) C++03
    (C) C++11
    (D) C++14

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 thread A is waiting on thread B to proceed, which thread B is waiting on thread A to proceed, this is called :
   (A) deadlock
   (B) livelock
   (C) race condition
   (D) starvation

2. What is the declaration of the postfix `operator--` when it is not a member of class `foo` ?
   (A) `foo operator-- (const foo&);`
   (B) `foo operator-- (const foo&, int);`
   (C) `foo operator-- (foo&);`
   (D) `foo operator-- (foo&, int);`

3. What is the amortized time for looking up a key in an `unordered_map` ?
   (A) $O(1)$
   (B) $O(\log_2 n)$
   (C) $O(n)$
   (D) $O(n \log_2 n)$

4. When a parent process waits for a completed child process, how many bytes are there in the exit status received from the `wait` system call ?
   (A) 1
   (B) 2
   (C) 3
   (D) 4

5. If we have `char *p` and `uintptr_t u`, what cast would be appropriate ?
   (A) `u = const_cast <uintptr_t> (p);`
   (B) `u = dynamic_cast <uintptr_t> (p);`
   (C) `u = reinterpret_cast <uintptr_t> (p);`
   (D) `u = static_cast <uintptr_t> (p);`

6. Given
   `map<string,int> m;`
   `auto& i = m.begin();`
   what is the type of `*i` ?
   (A) `const string`
   (B) `int`
   (C) `pair<const string,int>`
   (D) `pair<string,int>`

7. With standard OpenGL coördinates, the point $(1,1)$ is _____ of the point $(0,0)$.
   (A) above left
   (B) above right
   (C) below left
   (D) below right

8. What C++11 library type is used to enforce mutual access to a shared variable ?
   (A) `mutex`
   (B) `pthread`
   (C) `semaphore`
   (D) `spin lock`

9. If multiple threads are doing `++` on some variable without synchronization, this is called a :
   (A) deadlock
   (B) livelock
   (C) race condition
   (D) starvation

10. What kind of polymorphism is exhibited by templates ?
    (A) ad hoc conversion
    (B) ad hoc overloading
    (C) universal inclusion
    (D) universal parametric

11. Given `int* p; int i;` which answer will produce a compile-time error ?
    (A) `i + i`
    (B) `i + p`
    (C) `p + i`
    (D) `p + p`

12. The first version of C++ was designed in :
    (A) 1973
    (B) 1983
    (C) 1993
    (D) 2003