$Id: cmps109-2016q1-exam3.mm,v 1.47 2016-03-04 19:04:41-08 - - $

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 ! 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. Using OpenGL, write a function `draw_square` which will draw a blue square with the lower left corner at (10,10) and the upper right corner at (30,30). **[3✔]**

2. Define a template class `queue` with a single typename parameter `T`. Show only what should be in `queue.h`, not anything from `queue.cpp`, except for when an inline is explicitly requred.
   (i) Declare the private node which contains a T and a link to the next node in the list. **[1✔]**
   (ii) Declare the head and tail pointers to be automatically initiallized to the null pointer. **[1✔]**
   (iii) Disable the copy constructor and copy `operator=`. **[1✔]**
   (iv) Declare a `front` function which is a constant function returning a constant reference to the data in the first node. **[1✔]**
   (v) Declare a non-constant `front` function which returns a non-constant reference to the data in the first node. **[1✔]**
   (vi) Declare `pop_front` in the way consistent with other containers. Do not show the implementation. **[1✔]**
   (vii) Declare `push_back`. Do not show the implementation. **[1✔]**

3. Fill in each of the following declarations with the numbers 0 or 255, as appropriate. Score: ¼ point for each correct answer, rounded up to the next ½ point. **[2✓]**

`const GLubyte BLACK[]   = {     ,     ,     };`	`const GLubyte BLUE[]    = {     ,     ,     };`
`const GLubyte CYAN[]    = {     ,     ,     };`	`const GLubyte GREEN[]   = {     ,     ,     };`
`const GLubyte MAGENTA[] = {     ,     ,     };`	`const GLubyte RED[]     = {     ,     ,     };`
`const GLubyte WHITE[]   = {     ,     ,     };`	`const GLubyte YELLOW[]  = {     ,     ,     };`

4. Define a non-member template `operator==` which checks to see if two containers are equal. The type of the container is a template parameter. Assume the containers have the usual forward iterators. Ranges are identical if they are of the same length and pairwise values are equal. Assume `operator==` for the values. **[2✓]**

```
template <typename T>
bool operator== (const T& a, const T& b) {
```

5. Define a function `equal`. It has two template parameters which are iterators. It has four function arguments: a begin and end pair for the first iterator type, and then a begin and end pair of the second iterator type. Assume the iterators are only input iterators. Return true if the elements in the ranges are `operator==` true and that the ranges are of the same length. **[2✓]**

```
template <typename I1, typename I2>
bool equal (I1 begin1, I1 end1, I2 begin2, I2 end2) {
```

6. Define the template `operator<<` whose second argument is a constant reference to a `pair`. It has two template arguments which give the type of the `first` and `second` fields. It prints out a left brace (`{`), then the first of the pair, then a comma (`,`), then the second of the pair, then a right brace (`}`). Example: a `pair<int,int>` would look like `{3,4}`. **[2✓]**

7. Define a template `operator<` which takes two `pair`s by constant reference and returns true if the first is lexicographically less than the second. Definition: $(a,b) < (c,d)$ if $a < c$ or $a = c$ and $b < d$. You may assume `operator<` is available on both parts of the pair. Do not use any other comparison operator. **[2✓]**

8. Define an object-oriented hierarchy as shown in the three parts described here. Show all code as it would appear in a header file so that no implementation file is needed. All classes have the virtual functions `area` and `circumference` as well as suitable constructors. Make sure the constructors can not be used in implicit conversions.

   (a) Class `shape` is the base class with abstract functions, and a suitably protected constructor. **[2✔]**

   (b) Class `circle`[1] has a single radius field which is initialized by the constructor. For the mathematically challenged: $A = \pi r^2$, $C = 2\pi r$, and `<cmath>` defines `M_PI`. **[2✔]**

   (c) Class `square` has a single edge field (the length of one edge) initialized by the constructor. **[2✔]**

9. Given the outline of `array` presented here, fill in the blanks so that the main function will compile without errors. It need not execute without errors. **[4✔]**

```
struct array {
   size_t n; int* a;
   struct iterator {



   };
   iterator begin() {

   }
   iterator end() {

   }
};
int main() { array a; for (int& i: a) cout << i; return 0; }
```

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. Which condition is likely to cause unpredictable results during the execution of a multi-threaded process?
(A) deadlock
(B) livelock
(C) race condition
(D) starvation

2. A critical section in a multi-threaded process can be protected by means of a:
(A) exception
(B) firewall
(C) mutex
(D) thread

3. A process that sits in the background doing nothing until a client attempts to connect is called a:
(A) daemon
(B) fork
(C) mutex
(D) zombie

4. If memory is managed by the boundary tag method, and the size of a pointer is 8 bytes, what is the minimum byte distance from one allocation to the next?
(A) 8
(B) 16
(C) 32
(D) 64

5. A `shared_ptr` has a reference count which is stored:
(A) at another location on the heap outside the object
(B) in a statically allocated area in the data segment
(C) inside the object it points at
(D) inside the shared pointer itself

6. If `foo` is an abstract class, and we declare: `Foo* p; foo f;`, then the declaration of:
(A) both are OK
(B) both are in error
(C) `f` is OK, but `p` is in error
(D) `p` is OK, but `f` is in error

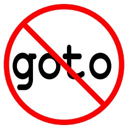
7. Which of the following statements will change meaning if the parentheses are removed?
(A) `a = (i);`
(B) `int (i);`
(C) `return (i);`
(D) `t = sizeof (int);`

8. Which category of iterator has the most operations permitted on it?
(A) bidirectional
(B) forward
(C) input
(D) random access

9. What system call is executed by a server while it is waiting for a client to request a connection?
(A) `accept`(2)
(B) `bind`(2)
(C) `listen`(2)
(D) `socket`(2)

10. When a `runtime_exception exn` is caught, what is used to extract the message from the exception?
(A) `exn.getMessage()`
(B) `exn.what()`
(C) `strerror(errno)`
(D) `to_string(exn)`

11. What keyword causes a function to be dynamically dispatched at runtime instead of being the direct target of a call instruction?
(A) `friend`
(B) `inline`
(C) `static`
(D) `virtual`

12.


(A) Edsger Dijkstra
(B) Donald Knuth
(C) Dennis Ritchie
(D) Bjarne Stroustrup

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. Which of the following containers does not allow `--i` if `i` is an iterator ?
   (A) `deque`
   (B) `forward_list`
   (C) `list`
   (D) `vector`

2. If `i` is an iterator and `n` is an int, what kind of iterator allows the computation `i[n]` ?
   (A) bidirectional
   (B) forward
   (C) input
   (D) random access

3. How many bits are there in an IPv4 address ?
   (A) 16
   (B) 32
   (C) 64
   (D) 128

4. If the hexadecimal number `0x12345678` is sent from a little-endian host to a big-endian host without any consideration for network byte order, what is its interpretation on the other machine ?
   (A) `0x21436587`
   (B) `0x56781234`
   (C) `0x78563412`
   (D) `0x87654321`

5. What statement will increment the object being pointed at by the iterator `i` ?
   (A) `*++i;`
   (B) `*i++;`
   (C) `++*i;`
   (D) `++i*;`

6. If `x` is an object of some particular class type, which is probably the most efficient way to increment it ?
   (A) `++x;`
   (B) `x++;`
   (C) `x=1+x;`
   (D) `x=x+1;`

7. If a C++ program wishes to indicate failure, and the main function executes `returnn;` where `n` is an `int` variable, then the value of `n` must be any value :
   (A) equivalent to false
   (B) greater than 0
   (C) not a multiple of 256
   (D) other than 0

8. For which of the following is there no restriction on the number of arguments that may be passed to it ?
   (A) operator()
   (B) operator*
   (C) operator->
   (D) operator<<

9. What keyword restricts access to members of a class to the class itself and all classes that inherit from it ?
   (A) `friend`
   (B) `private`
   (C) `protected`
   (D) `public`

10. Which of the following containers will allow insertion at an arbitrary position in not more than $O(1)$ time ?
    (A) `deque`
    (B) `list`
    (C) `map`
    (D) `vector`

11. What is the most likely implementation of a `map` ?
    (A) array of pointers to an array of pointers to objects
    (B) double-ended queue
    (C) hash table
    (D) red-black tree

12. It is necessary to make the destructor virtual if :
    (A) any constructor is virtual.
    (B) any member function is virtual.
    (C) some instance fields are pointers.
    (D) the class inherits from some other class.