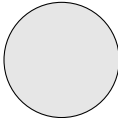
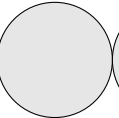
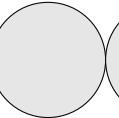
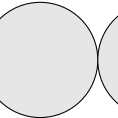
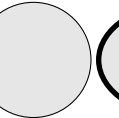
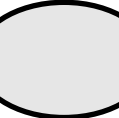


\$Id: cmpls109-2013q1-exam3.mm,v 1.33 2013-03-21 13:35:22-07 - - \$

page 1	page 2	page 3	page 4	page 5	Total / 52	<i>Please print clearly :</i>				
						<table><tr><td>Name :</td><td></td></tr><tr><td>Login :</td><td>@ucsc.edu</td></tr></table>	Name :		Login :	@ucsc.edu
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. **C++:** Define a template function which will merge two ranges into a single container. **[5✓]**
 - (a) It has three template parameters : a forward iterator, a container, and a binary predicate.
 - (b) It has six actual parameters : two iterators indicating one range, two iterators indicating another range, a container into which the ranges are to be merged, and a binary predicate returning a boolean which computes a less-than function.
 - (c) Assume the output container has a **push_back** method.
 - (d) Example call : `merge (c1.begin(), c1.end(), c2.begin(), c2.end(), vec, less);`
will assume the **c1** range is already sorted into order given by the **less** function, as is the **c2** range. The two ranges will then be copied in sorted order into the output container **vec**. From inside **merge**, the call **less(a,b)** will be true if **a** is considered less than **b**.

2. **C++:** Define a template class **stack** with all methods defined inline. **[5✓]**
 - (a) It uses a private **vector** field to hold the stack.
 - (b) None of the four default members are declared, since the automatic defaults are acceptable.
 - (c) **pop** removes but does not return the top of the stack.
 - (d) **push** pushes a new element onto the stack.
 - (e) **top** returns the top of the stack but does not modify the stack.
 - (f) **empty** tells whether the stack is empty or not.

3. **C++:** Write a template function `swap` that will exchange the values of any two variables. Example: `swap(a,b);` should exchange the values of `a` and `b` if the type of those variables has a copy constructor and an `operator=`. [2✓]
4. **Java:** Write a client/server application that shares the current date and time. Assume, but do not code, any necessary `import` statements.
- (a) Class `date_client` connects to port 10000 of `localhost`, reads one line, prints it, then exits. [2✓]
- ```
class date_client {
 public static void main (String[] args) throws IOException {
```
- (b) Class `date_server` creates a server socket on port 10000, then goes into an infinite loop waiting for clients. Whenever a client connects, it replies by sending the current date, after which it disconnects the client and waits for the next client. Create a new `Date()` object each time a date is needed and use its `toString` method. [3✓]
- ```
class date_server {  
    public static void main (String[] args) throws IOException {
```
5. **C++:** Define a template function called `find`. Its first template argument is a forward iterator, and its second template argument is a predicate of one argument. The function itself has three arguments, the first and second being a pair of iterators that bound the search space, and the third being a function of one argument which returns a boolean value. Return the first occurrence in the range for which the predicate is true, and if not found, return the end iterator. Example call: `find (c.begin(), c.end(), ispositive);` [3✓]

6. C++: Write a loop that will copy program arguments (but not the program name) into the vector. [1✓]

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

7. C++: Consider the trivial class shown here.

- (a) Write the prototypes for the `operator=` and the destructor that would otherwise be provided by default. [1✓]

```
struct box {  
    size_t size;  
    int *data;
```

- (b) Implement `operator=` for this class. It needs to copy the data instead of sharing it with the source object. Code it as it would appear in the implementation (.cpp) file. [3✓]

- (c) Implement the destructor as it would appear in the implementation file. [1✓]

8. C++: Finish the function, which returns the average value in the range. [2✓]

```
template <typename iterator>  
double average (const iterator &begin, const iterator &end) {
```

9. C++: Define `operator<<` which can print all of the elements of a vector of any type, assuming that the constituent elements already have an appropriate `operator<<` defined. The operator prints all of the elements with a single space between successive elements. [2✓]

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		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	0
column total $c = \max(a - b, 0)$	11		$= c$

- If **v** is a **vector**, then **v.begin()** is the same as :
 - v[0]**
 - &v[0]**
 - *v[0]**
 - ~v[0]**
- Which of the following operators does a bidirectional iterator have that a forward iterator does not have ?
 - *i**
 - ++i**
 - i**
 - i->f**
- For an integer **a**, if **a** is (x) then **a/8** is equivalent to (y) .
 - (x) **signed**; (y) **a<<3**
 - (x) **signed**; (y) **a>>3**
 - (x) **unsigned**; (y) **a<<3**
 - (x) **unsigned**; (y) **a>>3**
- What is the declaration of the postfix operator **++** which is a member of class **foo** ?
 - foo &operator++ ();**
 - foo &operator++ (foo &);**
 - foo operator++ (foo &, int);**
 - foo operator++ (int);**
- Which of the following instance fields will most strongly suggest that a default **operator=** is inappropriate for a class ?
 - inline object
 - pointer
 - primitive
 - reference
- The outermost container in a Java GUI is a
 - JArea**
 - JFrame**
 - JGrid**
 - JPanel**
- Storage management by reference counting fails on what kind of data structure ?
 - acyclic graph
 - binary tree
 - cyclic graph
 - hash table
- The **std::map** data structures is a :
 - directed acyclic graph
 - hash table
 - linear linked list
 - red-black tree
- Which data structure uses a contiguous block of heap memory ?
 - std::deque**
 - std::list**
 - std::map**
 - std::vector**
- If the exception **exn** is thrown, what is the proper way to catch it ?
 - catch (exn &e)**
 - catch (exn *e)**
 - catch (exn ~e)**
 - catch (exn e)**
- The first language to use concepts such as *class* and *virtual* was :
 - Algol
 - Fortran
 - Lisp
 - Simula

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		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	0
column total $c = \max(a - b, 0)$	11		$= c$

- When two threads update the same variable without synchronization, leading to unpredictable results, this is called a ____ condition.
 - daemon
 - deadlock
 - race
 - spinlock
- An abstract class :
 - can not be instantiated
 - can not specify an interface
 - has no virtual functions
 - must be derived from a base class
- The Java GUI considers the coordinate (0,0) to be at what part of the screen ?
 - lower left
 - lower right
 - upper left
 - upper right
- The Postscript coordinate system considers (0,0) to be at what part of the page ?
 - lower left
 - lower right
 - upper left
 - upper right
- What is an interface that requires implementation of the method `run` ?
 - Runnable**
 - Synchronized**
 - Thread**
 - Throwable**
- In C++11, if an implicitly generated member should be suppressed as inappropriate for the class, what should be done ?
 - Declare it as private and not implement it.
 - Declare it as public and throw a `logic_error` exception if it is called.
 - Declare it as public but instead of implementing it, mark it as `delete`.
 - Declare it but put an `assert(false)` statement in the body.
- Which operator may be declared with any number of arguments, depending on the semantics of the operator ?
 - `operator()`
 - `operator++`
 - `operator<>`
 - `operator[]`
- A keyword in C++11 that has a radically different meaning from what it has in ANSI C is :
 - `auto`
 - `struct`
 - `typedef`
 - `volatile`
- Following the **Makefile** dependency `%.o : %.cpp` an appropriate compilation command would be :
 - `${GPP} %.cpp`
 - `${GPP} -c $<`
 - `${GPP} -c $@`
 - `${GPP} -o $<`
- Given the declaration of `i`, what will print out a string ?


```
vector<string*>::iterator i;
```

 - `cout<<i;`
 - `cout<<*i;`
 - `cout<<**i;`
 - `cout<<***i;`
- Following `foo x;` the statement `foo y = x;` is a call to :
 - assignment `operator=`
 - the copy constructor
 - the default constructor
 - the destructor