`$Id: cmps109-2019q3-final.mm,v 1.57 2019-08-15 13:58:50-07 - - $`

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

PLEASE PRINT CLEARLY :

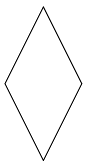
Name :
CruzID : `@ucsc.edu`

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. Define the following operator. Assume the canonical form of a `ubigint` as specified in the assignment. **[3✔]**

```
bool ubigint::operator< (const ubigint& that) {
```

2. Write the OpenGL function that draws a red diamond, given the *x* and *y* coördinates of its center, and its width and height. A diamond is a quadrilateral such as is shown to the left of the code. The top and bottom points have the same *x* coördinate. The left and right points have the same *y* coördinate. Also fill in the blank for the definition of **`RED`**. **[3✔]**



```
const GLubyte RED[] = ______________________;
void draw_red_diamond (GLfloat xcenter, GLfloat ycenter,
                       GLfloat width, GLfloat height) {
```

3. Code a binary search, given a range specified by a pair of iterators. Assume the elements of the range are sorted into increasing order as given by the `less_t` function argument. Return an iterator pointing at the element found. Handle the not-found case in the usual way. Assume the iterators are random access iterators — their distance can be computed by subtraction and that they can be compared by any of the standard comparison operators. They can also be subscripted in the same way as pointers can be used to perform pointer arithmetic. **[4✔]**

```
template <typename itor, typename item_t, class less_t>
itor binary_search (itor begin, itor original_end,
                    const item_t& item, less_t less) {
```

4. Assume `foo x; foo y;`, and that `operator+=` is a member of `foo`, define a non-member function that is called by the expression `x + y`. **[1✔]**

5. Each of the following boxes represents one kind of polymorphism. In each box, write the letter **U** if it represents universal polymorphism, and **A** for ad-hoc polymorphism. Also, write **C** for conversion, **O** for overloading, **P** for parametric, and **I** for inclusion.
Score: 1 point if 4 correct; ½ point if 2 or 3 correct; 0 point if 0 or 1 correct. **[1✔]**

`void f(int);`
`void f(string);`
	`class baz: private qux {`
	`};`

`template <typename T>`
`T sum (T*);`
	`void f(double);`
	`int x; f(x);`

6. Define `struct complex` (all members are public).
   (a) It has two `double` fields `real` and `imag`, which are explicitly initialized to 0.0. **[1✔]**
   (b) It has a constructor which accepts 0, 1, or 2 arguments, defaulting to 0.0. This constructor will allow a `double` to be implicitly converted to `complex`. **[1✔]**
   (c) A member `operator+=` which returns a reference the left side operand. **[1✔]**
   (d) An operator which converts a `complex` to `bool` and is false if both fields are 0.0. **[1✔]**

7. Define `merge`, which merges two input ranges into a single output range, given a pair of iterators for each of the input ranges, an iterator for the output range, and a less-than comparison function. Assume both input ranges are sorted into ascending order as specified by the `Less` argument. Also assume that the output iterator can be used with expressions similar to `*result++` without overflowing the output container. Do not use `operator<` or make any assumptions about the types of the elements being merged. **[4✔]**
Example call: `merge (a.begin(), a.end(), b.begin(), b.end(), back_inserter(v), less<int>());`

```
template <typename Initor1, typename Initor2,
          typename Outitor, typename Less>
void merge (Initor1 begin1, Initor1 end1, Initor2 begin2, Initor2 end2,
            Outitor result, Less less) {
```

8. Define the function `inner_product`. It has two arguments, each of type `vector<double>` and produces a `double` result. Throw a `domain_error` if the two vectors are of different sizes. The formula for an inner product is given at the left. **[2✔]**

$$p = \sum_{i=0}^{n-1} u_i v_i$$

9. Class `forward_list`.
   (a) Given the outline of class `forward_list` shown here, complete the necessary operators that are members of class `forward_list<item_t>::iterator` so that the following statement compiles correctly.
   `for (int i: fli) cout << << i;`
   Code all iterator functions inline. **[3✔]**

```
template <typename item_t>
class forward_list {
  private:
    struct node {
      item_t item {};
      node* link;
    };
    node* head {};
  public:
    ~forward_list();
    void pop_front();
    bool empty() {
      return head == nullptr;
    }
    . . .
    class iterator;
    iterator begin() { return head; }
    iterator end() { return nullptr; }
};
```

```
template <typename item_t>
class forward_list<item_t>::iterator {
  private:
    node* curr;
  public:











};
```

   (b) Code `pop_front` as it would appear defined outside of the class. **[3✔]**

   (c) Code the destructor as it would appear outside of the class. Make use of `pop_front` and `empty`. **[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. **[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. What keyword prevents a constructor from being used as an implicit conversion operator ?
   (A) `explicit`
   (B) `implicit`
   (C) `operator`
   (D) `virtual`

2. Which system call creates an endpoint for bidirectional communication between two processes, possibly running on different hosts ?
   (A) `exec`
   (B) `fork`
   (C) `pipe`
   (D) `socket`

3. If a `map` contains 10 000 000 elements, approximately what is the expected number of comparisons needed to search it ?
   (A) 1
   (B) 7
   (C) 24
   (D) 3 162

4. What should be placed in the blank in the following prototype ?
   `ostream& operator<< (____, const foo&);`
   (A) `const ostream&`
   (B) `ostream&`
   (C) `ostream&&`
   (D) `ostream*`

5. Which container uses the largest number of auxiliary (not client declared) pointers to implement the data structure ?
   (A) `deque<int> c;`
   (B) `list<int> c;`
   (C) `map<int> c;`
   (D) `vector<int> c;`

6. If any function in a class is declared as virtual, then what must also be declared as virtual ?
   (A) all constructors
   (B) all private member fields
   (C) all privately inherited base classes
   (D) the destructor

7. When a program executes `exit(n)`, how many bits of `n` are stored in the parent's wait status variable ?
   (A) 8
   (B) 16
   (C) 24
   (D) 32

8. For a vector, if `v.size()==5`, then `v.end()` points at which element ?
   (A) `v[0]`
   (B) `v[1]`
   (C) `v[4]`
   (D) `v[5]`

9. In the `main` function, which expression has type `char` ?
   (A) `argv`
   (B) `*argv`
   (C) `**argv`
   (D) `***argv`

10. Which is the proper declaration of `foo::operator->` ?
    (A) `foo& operator->();`
    (B) `foo& operator->(__field_name);`
    (C) `foo* operator->();`
    (D) `foo* operator->(__field_name);`

11. How is `GLubyte` defined ?
    (A) `typedef signed char GLubyte;`
    (B) `typedef signed char* GLubyte;`
    (C) `typedef unsigned char GLubyte;`
    (D) `typedef unsigned char* GLubyte;`

12. which is a move constructor ?
    (A) `foo (const foo&&);`
    (B) `foo (const foo&);`
    (C) `foo (foo&&);`
    (D) `foo (foo&);`

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 container's contents may be passed directly to a C function ?
   (A) `deque<char>`
   (B) `list<char>`
   (C) `map<char>`
   (D) `vector<char>`

2. Of these, which data structure has the worst locality of reference ?
   (A) `deque`
   (B) `list`
   (C) `string`
   (D) `vector`

3. Which regular expression which match any number of characters other than a newline, but only as few characters as possible.
   (A) `.*`
   (B) `.*?`
   (C) `[a-z]*`
   (D) `\S*`

4. Which shell command will redirect both standard out and standard error from the program `foo` into the file `bar` ?
   (A) `foo <bar 1>&2`
   (B) `foo <bar 2>&1`
   (C) `foo >bar 1>&2`
   (D) `foo >bar 2>&1`

5. Given `foo x;` what is the definition of the operator that is called by the expression `x++` ?
   (A) `foo operator++ (const foo&, int);`
   (B) `foo operator++ (foo&, int);`
   (C) `foo operator++ (foo, int);`
   (D) `foo& operator++ (foo&, int);`

6. On a little-endian architecture (e.g., x86-64), what is the output from the following ?

```
uint32_t x = 0x12345678;
int n = *reinterpret_cast<char*> (&x);
cout << hex << n << endl;
```

   (A) `12`
   (B) `34`
   (C) `56`
   (D) `78`

7. Which of the standard streams is not affected by the following command ?

```
foo <bar >baz
```

   (A) `<iostream>`
   (B) `std::cerr`
   (C) `std::cin`
   (D) `std::cout`

8. Which of the following commands will cause output to include files whose first character is a dot ?
   (A) `ls -.`
   (B) `ls -a`
   (C) `ls -g`
   (D) `ls -l`

9. How is an `unordered_map` implemented ?
   (A) array of pointers to elements
   (B) balanced binary search tree
   (C) hash table
   (D) linear linked list

10. What kind of cast is used to convert bits from one type to another completely unrelated type ?
    (A) `const_cast`
    (B) `dynamic_cast`
    (C) `reinterpret_cast`
    (D) `static_cast`

11. In the definition of `operator=`, what statement is used to return a reference to the current object ?
    (A) `return &this;`
    (B) `return *this;`
    (C) `return ++this;`
    (D) `return this;`

12. How long would it take to insert a new element into the middle of a `vector` ?
    (A) $O(1)$
    (B) $O(\log n)$
    (C) $O(n)$
    (D) $O(n \log n)$