$Id: cmps109-2019q2-final.mm,v 1.190 2019-06-05 12:36:40-07 - - $

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. Each of the following boxes represents one kind of polymorphism. In each box, write the letter **U** for universal, or **A** for ad-hoc. Also : write **C** for conversion, **O** for overloading, **P** for parametric, or **I** for inclusion. Score : ½ each box with two correct letters. **[2✔]**

`void f(int);` `void f(string);`	`class baz: private qux {` `};`
`template <typename T>` `T sum (T*);`	`void f(double);` `int x; f(x);`

2. Write the prototypes for all functions and other members that would be implicitly generated for class `foo`. Show their prototypes as they would be declared inside the class. **[2✔]**

2 ✔ if 6 correct
1½ ✔ if 5 correct
1 ✔ if 4 correct
½ ✔ if 3 correct
0 ✔ otherwise

```
class foo {
   public:
```

3. Give the name of each color defined in the following table according to the sample statement shown here. **[2✔]**
`const GLubyte` _____ `[] {0x??, 0x??, 0x??};`

2 ✔ if 8 correct
1½ ✔ if 7 or 6 correct
1 ✔ if 5 or 4 correct
½ ✔ if 3 or 2 correct
0 ✔ if 1 or 0 correct

`{0x00, 0x00, 0x00};`	`{0x00, 0x00, 0xFF};`
`{0x00, 0xFF, 0x00};`	`{0x00, 0xFF, 0xFF};`
`{0xFF, 0x00, 0x00};`	`{0xFF, 0x00, 0xFF};`
`{0xFF, 0xFF, 0x00};`	`{0xFF, 0xFF, 0xFF};`

4. Show the prototypes for the prefix and postfix `operator++` as they would appear inside and outside of the class definitions. Assume the declaration `foo x;` in the operator calls at the left. Score : ½ each answer. **[2✔]**

	`class foo { // as members`	`// Not as members of class foo.`
`++x`		
`x++`		

5. Define the member function `swap` which exchanges the value in the box with its argument and returns the old value. Be sure there are no race conditions when used by multiple concurrent threads. Show the code as it would appear in the implementation file. **[2✔]**

```
class lockbox {
   private:
      mutex lock;
      size_t value {0};
   public:
      size_t swap (size_t argval);
}
```

6. Write the function **ubigint::trim** which removes high order zeros from a **ubigint** as per the project specifications. Remember that zero is represented by an empty vector. **[1✔]**

```
class ubigint {
   private:
      vector<unsigned char> value;
      void trim() {
};
```

```
void ubigint::trim() {
```

7. The function **minimum** takes a pair of iterators indicating a range and a less than comparison function and returns an iterator pointing at the smallest element in the range. If there is more than one smallest element, the first one is chosen. **[2✔]**

```
template <typename itor, typename less_t = less<decltype(*itor())>>
itor minimum (itor begin, itor end, less_t less = less_t()) {
```

8. Consider the following abstract base class used as a base for expression tree evaluation. Code all functions inline inside of the class declarations. Then define derived classes **number** and **adder** which override the abstract functions.

```
class expr {
   public:
      virtual double eval() const = 0;
      virtual void print (ostream&) const = 0;
};
```

   (a) Define the class **number** which has a private field holding a **double**. Override the base class functions, and add a constructor whose argument is a **double** which has a default value of 0. **[2✔]**

   (b) Define the class **adder** whose private fields are **shared_ptr**s to **expr**s called **left** and **right**. **Eval** returns the sum of the values of its children. **Print** prints out the tree itself by printing an open parenthesis, followed by printing the left subtree, then a comma, then the right subtree, then a closing parenthesis. The constructor takes two arguments which are used to initialize the left and right pointers. **[4✔]**

   (c) Define a non-member **operator<<** which dispatches the print function based on the right argument. **[1✔]**

9. Define the function **inner_product** which takes two pairs of iterators and returns the inner product of the elements. Throw a **domain_error** if the ranges are of different sizes. Assume the iterators point at **double**s, and only satisfy the requirements of input iterators. The formula for an inner product is given at the left. **[2✔]**

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

```
template <typename itor>
double inner_product (itor begin1, itor end1, itor begin2, itor end2) {
```

10. Define the function **draw_purple_square**. Its first two arguments are the x and y positions (in that order) of the center of the square. Its third argument is the length of one edge. Make the color a static local variable. **[2✔]**

```
-bash-14$ grep 'purple' /usr/share/X11/rgb.txt
160  32 240            purple
```

11. Finish the function **draw_circle**. **[2✔]**

```
void draw_circle (GLfloat xcenter, GLfloat ycenter, GLfloat radius,
                  const GLubyte* color_ubv) {
   glBegin (GL_POLYGON);
   glColor3ubv (color_ubv);
   const GLfloat delta = 2.0 * M_PI / 64;
   for (GLfloat theta = 0; theta < 2.0 * M_PI; theta += delta) {


   }
   glEnd();
}
```

12. Show the code for **thing::iterator** as it would appear in a header file, but outside the class **thing**. All of the iterator's operators should be defined inline. Show only those members that are needed for the following statements to compile. **thing t; for (auto& i: t) cout << i << endl;** Class **thing** is shown here. **[2✔]**

```
class thing {
   vector<int> v;
public:
   class iterator;
   iterator begin();
   iterator end();
};
```

13. Define **equal**, which takes two pairs of iterators and returns true if and only if the elements of the ranges given are equal, and the lengths of the ranges are equal. Assume only forward iterators: You may not use **size()** or subtract iterators. Assume **operator==** is defined on the elements of the ranges. **[2✔]**

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

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 uses the smallest number of boundary tags ?
   (A) `list`
   (B) `map`
   (C) `unordered_map`
   (D) `vector`

2. If an allocation is done with the following statement, how should it be freed ?
   ```
   p = new T[n];
   ```
   (A) `delete p`
   (B) `delete p[]`
   (C) `delete[] p`
   (D) `depete p[n]`

3. In the following declaration, which is not valid as a template parameter to fill in the blank ?
   ```
   template <____ T> class foo {};
   ```
   (A) `class`
   (B) `double`
   (C) `size_t`
   (D) `typename`

4. Which of the following library containers provides the best locality of reference ?
   (A) `deque`
   (B) `forward_list`
   (C) `list`
   (D) `vector`

5. What library function, by itself, allows a process to read the output of another processes that is created as a child process ?
   (A) **exec**(3)
   (B) **fopen**(3)
   (C) **fork**(2)
   (D) **popen**(3)

6. What system call creates a child process ?
   (A) **exec**(3)
   (B) **fopen**(3)
   (C) **fork**(2)
   (D) **popen**(3)

7. Which option to `G++` prevents deletion of an object file and causes an executable image not to be linked ?
   (A) `g++ -E`
   (B) `g++ -S`
   (C) `g++ -c`
   (D) `g++ -o`

8. Which keyword, prefixed to a constructor, will prevent that constructor from being used as an implicit conversion operator ?
   (A) `decltype`
   (B) `explicit`
   (C) `implicit`
   (D) `nothrow`

9. What will be printed by the following code on a little-endian architecture like the x86-64 ?
   ```
   int n = 0x12345678;
   cout << hex
        << int(*reinterpret_cast<char*>(&n));
   ```
   (A) `12`
   (B) `34`
   (C) `56`
   (D) `78`

10. When drawing a circle or ellipse, what function is *xxx* in the following line of code ?
    ```
    int xpos = w * xxx (angle) + center;
    ```
    (A) `cos`
    (B) `exp`
    (C) `log`
    (D) `tan`

11. What converts a pointer to an insigned integer ?
    (A) `n = const_cast<uintptr_t>(p);`
    (B) `n = dynamic_cast<uintptr_t>(p);`
    (C) `n = reinterpret_cast<uintptr_t>(p);`
    (D) `n = static_cast<uintptr_t>(p);`

12. What form of polymorphism is implemented by templates ?
    (A) conversion
    (B) inclusion
    (C) overloading
    (D) parametric

> Trying to learn to hack on a Microsoft Windows machine or under any other closed-source system is like trying to learn to dance while wearing a body cast.
>
> It is not possible to effectively secure Windows systems against crack attacks. The code and architecture simply have too many flaws, which makes securing Windows like trying to bail out a boat with a sieve. The only reliable prevention starts with switching to Linux or some other operating system that is designed to at least be capable of security.
>
> — Eric S. Raymond, *"How To Become A Hacker"*
> `http://www.catb.org/~esr/faqs/hacker-howto.html`

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 is the type of `cout` ?
   (A) `fstream`
   (B) `istream`
   (C) `ostream`
   (D) `sstream`

2. What part of a `bash` command will redirect `cerr` into the file `foo` ?
   (A) `0>foo`
   (B) `1>foo`
   (C) `2>foo`
   (D) `3>foo`

3. What is the type of the literal `"abc"` ?
   (A) `char*`
   (B) `char[3]`
   (C) `char[4]`
   (D) `string`

4. When managing a data structure using `shared_ptr` objects, what kind of data structure needs special handling ?
   (A) acyclic graph
   (B) binary search tree
   (C) cyclic graph
   (D) linear linked forward list

5. Which of the following initializers will specify the darkest of these colors ?
   `GLubyte ubvec[3] ____;`
   (A) `{ 0, 0, 0}`
   (B) `{127,127,127}`
   (C) `{255,127, 0}`
   (D) `{255,255,255}`

6. What member declaration will allow an object of class `foo` to be used in a context where a boolean value is required ?
   (A) `bool operator== (const foo&) const;`
   (B) `const foo& operator== (bool) const;`
   (C) `operator bool() const;`
   (D) `static_cast<bool>(const foo&);`

7. After `p = new T[n]`, how should `p` be deleted ?
   (A) `delete p`
   (B) `delete p[]`
   (C) `delete p[n]`
   (D) `delete[] p`

8. After a process calls `exit(n)`, how many bits of `n` are sent to the parent process ?
   (A) 8
   (B) 16
   (C) 32
   (D) 64

9. Which of the following regular expressions will match zero or more characters (not including newline), but as few characters as possible ?
   (A) `.*`
   (B) `.*?`
   (C) `.+`
   (D) `.+?`

10. In a command following a `Makefile` dependency, what is the variable which names the target ?
    (A) `$$`
    (B) `$*`
    (C) `$<`
    (D) `$@`

11. The first preprocessor directive in the file `foo.h` should be :
    (A) `#define __FOO_H__`
    (B) `#endif __FOO_H__`
    (C) `#ifdef __FOO_H__`
    (D) `#ifndef __FOO_H__`

12. How is `vector::size` declared ?
    (A) `const size (size_t);`
    (B) `const size_t size();`
    (C) `size_t size (const);`
    (D) `size_t size() const;`