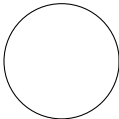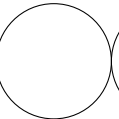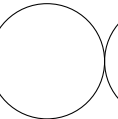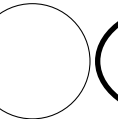page 1   page 2   page 3   page 4   page 5   Total / 54   *Please print clearly :*

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. 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, polymorphism. **[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. Define the inline *prefix* `operator++` which will increment the count field, but will avoid any race conditions if called simultaneously be multiple threads. **[2✔]**
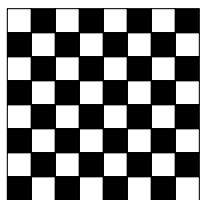
```
class counter {
      size_t count {0};
      mutex lock;
   public:
```

3. Fill in the following definitions of colors using either 0 or 255 as appropriate. **[1✔]**

3 correct : 1 ✔   `const GLubyte RED[] { _____, _____, _____};`
2 correct : ½ ✔
else : 0 ✘   `const GLubyte GREEN[] { _____, _____, _____};`

                       `const GLubyte BLUE[] { _____, _____, _____};`

4. Define constants `GLubyte` for the colors `BLACK` and `WHITE`. Define a vector of pointers to `GLubyte`s initialized to `BLACK` at index 0 and `WHITE` at index 1. **[1✔]**

5. Write a function to draw a chessboard. It consists of alternating black and white squares, with the upper left and lower right square being white, and the lower left and upper right being black. Use the vector from the previous question. Obviously, use an inner loop nested within an outer loop. Use the coördinates (x_0, y_0) for the lower left corner, and $(x_0 + size, y_0 + size)$ for the upper right corner. **[4✔]**


```
void draw_chessboard (GLfloat x0, GLfloat y0, Glfloat size) {
```

6. Define the template function `find`, which does a linear search. It has two template arguments: the type of the iterator and the type of the element to search for. It has three function arguments: the begin iterator, the end iterator, and the element to search for. it returns an iterator pointing at the first element equal to the third argument. Use `operator==` for comparisons. **[2✔]**

7. Code a binary search. Given are two random access iterators and a key to be located in the range bounded by the iterators. You may use any operator defined on random access iterators. Assume that the range is sorted into increasing order by `operator<` on the elements in the range. Do not use any comparison operator other than `operator<` on the elements pointed at by the iterators. Return an iterator pointing at the element found, or the end iterator. **[4✔]**

```
template <typename itor, typename item>
itor binary_search (itor begin, itor end, const item& it) {
```

8. A polynomial can be respresented as a vector, with coefficients stored at the position given by their power. For example, $8x^5 + 9x^3 + 7x^2 + 10$ would be represented as the vector `{10, 0, 7, 9, 0, 8}`. That is, the term ax^n is stored in a vector `v` with `v[n]==a`. The partial header file `polynomial.h` is shown here. Write code that would be placed in the implementation file `polynomial.cpp`.

```
class polynomial {
    friend ostream& operator<< (ostream&, const polynomial&);
    private: vector<double> v;
    public: bool operator== (const polynomial&) const;
            polynomial& operator+= (const polynomial&);
}
polynomial operator+ (const polynomial&, const polynomial&);
```

    (a) Code `operator==`. **[1✔]**

    (b) Code `operator+=`. **[3✔]**

9. Continuing the example from the previous page, code `operator+`, which is neither a member nor a friend of the polynomial class. Show what would appear in the `cpp` file. **[1✔]**

10. Define the function `canonicalize` as it would fit in with the bigint project. It is a private function which takes a vector of digits and removes all high-order zeros. The number zero is represented by an empty vector. **[2✔]**

```
using ubigvalue_t = vector<unsigned char>;
void canonicalize (ubigvalue_t& vec) {
```

11. Inheritance. Define three classes, but show only what is specifically required. Code all functions inline.

   (a) Class `shape` is abstract and has the abstract functions `circumference` and `area`. They take no arguments and return a `double`. Disable the copy constructor and copy `operator=`. **[3✔]**

   (b) Class `square` inherits from `shape`. It has a private `width` field and an explicit constructor with a `double` argument, which defaults to 0. **[2✔]**

   (c) Class `circle` inherits from `shape`. It has a private `radius` field and an explicit constructor with a `double` argument, which defaults to 0. Assume the header `<cmath>`, which defines `M_PI`. (For the mathematically untutored : $A = \pi r^2$ and $C = 2\pi r$.) **[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. If the number `0x12345678` on a little-endian machine is passed through the function `htonl`, what will be the result?
   (A) `0x12345678`
   (B) `0x56781234`
   (C) `0x78563412`
   (D) `0x87654321`

2. What shows the current memory availability in a `vector`, including both used and unused space?
   (A) `v.capacity()`
   (B) `v.end()`
   (C) `v.length()`
   (D) `v.size()`

3. Which container has the worst locality of reference?
   (A) `T[n]`
   (B) `deque<T>`
   (C) `list<T>`
   (D) `vector<T>`

4. In the boundary tag method of storage management, what is the memory overhead per allocated chunk of memory?
   (A) 2 chars
   (B) 2 doubles
   (C) 2 pointers
   (D) 2 strings

5. What kind of cast just moves bits, without doing any kind of type checking?
   (A) `const_cast`
   (B) `dynamic_cast`
   (C) `reinterpret_cast`
   (D) `static_cast`

6. Which function is called by the following statements?
   `int n = 6; foo (&n);`
   (A) `void foo(int&&);`
   (B) `void foo(int&);`
   (C) `void foo(int);`
   (D) `void foo(int*);`

7. Using the OpenGL coordinate system from the project, if the point $(0, 0)$ is shown at the center. Where is the point $(-1, -1)$?
   (A) upper left
   (B) upper right
   (C) lower right
   (D) lower left

(A)	(B)
	(0,0)
(D)	(C)

8. Assuming `c` is a container of objects, what is the most efficient way to iterate over it?
   (A) `for (auto c: i)`
   (B) `for (auto i: c)`
   (C) `for (auto& i: c)`
   (D) `for (auto* i: c)`

9. If `int status` contains the 16-bit number returned by a `waitpid` system call, how can the value returned from the `main` function be accessed?
   (A) `status & 0x7F`
   (B) `status >> 7 & 1`
   (C) `status >> 8`
   (D) `status | 0xFF`

10. If the variable `int x` is acted on by multiple threads by using the expression `++x` and no `mutex`es are used, what will result?
    (A) deadlock
    (B) livelock
    (C) race condition
    (D) starvation

11. If two processes communicating via a socket repeatedly write into their end of the socket, but neither process reads from the socket, what will be the result?
    (A) deadlock
    (B) livelock
    (C) race condition
    (D) starvation

12. What language, besides C, is an ancestor of C++?
    (A) Algol 60
    (B) Fortran IV
    (C) Lisp
    (D) Simula 67

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. Inside the function `f`, how may the variable `p` be referred to for the following call?
   `p->f(x,y)`
   (A) `auto`
   (B) `left`
   (C) `p`
   (D) `this`

2. In an object-oriented heirarchy that uses virtual functions, what must also be marked as virtual? Assume the base class is `object`.
   (A) `~object()`
   (B) `bool operator== (const object&)`
   (C) `object& operator= (const object&)`
   (D) `object()`

3. In an object-oriented heirarchy that uses virtual functions, what should be marked as `= delete`? Assume the base class is `object`.
   (A) `~object()`
   (B) `bool operator== (const object&)`
   (C) `object& operator= (const object&)`
   (D) `object()`

4. What is the expected speed of insertion into the middle of each of the following data structures?
   (A) `vector<T>` is $O(1)$ and `list<T>` is $O(1)$
   (B) `vector<T>` is $O(n)$ and `list<T>` is $O(1)$
   (C) `vector<T>` is $O(1)$ and `list<T>` is $O(n)$
   (D) `vector<T>` is $O(n)$ and `list<T>` is $O(n)$

5. A destructor for a derived class destroys in what order?
   (A) base class first,
       then fields in declaration order.
   (B) base class first,
       then fields in reverse declaration order.
   (C) fields in declaration order first,
       then base class.
   (D) fields in reverse declaration order first,
       then base class.

6. Given `string* s; string* t;` what is impossible?
   (A) `s != t and *s != *t`
   (B) `s != t and *s == *t`
   (C) `s == t and *s != *t`
   (D) `s == t and *s == *t`

7. In the producer-consumer problem, what is used to make the consumer wait if the buffer is empty?
   (A) `condition_variable`
   (B) `mutex`
   (C) `this_thread::sleep_for`
   (D) `unique_lock`

8. When a server process has created a socket, what system call is used to suspend itself pending an attempt by a client to `connect`?
   (A) `accept`
   (B) `bind`
   (C) `create`
   (D) `listen`

9. When a `main` function executes `return status;` how many bits of the variable `status` are returned to the parent process?
   (A) 8
   (B) 16
   (C) 24
   (D) 32

10. What is the efficiency of `unordered_map::find`?
    (A) $O(1)$
    (B) $O(\log_2 n)$
    (C) $O(n)$
    (D) $O(n \log_2 n)$

11. What is the efficiency of `map::find`?
    (A) $O(1)$
    (B) $O(\log_2 n)$
    (C) $O(n)$
    (D) $O(n \log_2 n)$

12. Who invented C++?
    (A) John Backus
    (B) Grace Hopper
    (C) Dennis Ritchie
    (D) Bjarne Stroustrup