Quiz >

Review answers

Start date:	9 minutes ago
Complete date:	A moment ago
Question 1:	Which statement is false about sequence containers? ✓ A dequeu can insert elements at the beginning and extract elements from the end but can also insert elements at the end and extract elements from the beginning. ○ The *std::list* class supports the square bracket [] operator. ○ Inserting elements in an *std::list* is faster than inserting elements in an *std::vector*. ○ Traversing to a certain element in a vector is faster than in a list.
Question 2:	Which statement is false about sequence containers? ○ Sequence containers store their data linearly. ✓ A vector is like an array that can add elements at the end but not at the beginning of the array. ○ The *list<T>::pop_front()* function does only remove the first element but does not return the first element. ○ Sequence containers order their data.
Question 3:	Which statement is false about iterators? ○ A begin iterator points to the first element of a container. ✓ An end iterator points to the last element of a container. ○ A regular pointer in a regular array is also an STL compatible iterator. ○ To access the data an iterator is pointing to, you must dereference the iterator (*).
Question 4:	Which statement is true about predicates? ✓ A predicate is one of the two main parts of a sentence, the other being the subject, which the predicate modifies.

◯ A predicate is a functor (function object or global function) that returns a boolean.
◯ A predicate is a brand of dog food.
◯ A predicate is a (function object or global function) that changes an element.

Question 5: Which statement is false about iterators?

◯ With the correct use of iterators (in combination with templates) you can write functions that work with every STL container.
◯ Stream iterators are adapters that allows us to use a stream as source or destination in code that uses iterators.
✓ On iterators you can only use the pre-increment (++it) operator and not the post-increment (it++).
◯ Insert iterators are adaptors that transform an assignment (*it=value) to an insert, push_back or push_front operation on a container.

Question 6: Which two statements are false about STL containers?

☐ Arguments STL container operations are checked for correctness.
☑ STL containers cannot store pointers.
☐ Vectors generally allocate more memory than needed for the elements it stores.
☑ Elements stored in an STL container must be copyable (must provide a copy constructor).

Question 7: Which statements are false about algorithms?

☑ STL algorithms accept a start- and end-iterator instead of the complete container.
☑ Removing algorithms are a special kind of mutating algorithms.
☐ Mutating algorithms change the order of elements but not the elements themself.
☐ Modifying algorithms can modify the elements of data structures and change the order.

Question 8: Which statements are true about the following code?

C++:

```
1  // Print the list contents.
2  template <typename T>
3  void Print(const T& ds)
4  {
5      // Typedef for the iterator to simplify code.
```

```
6       typedef T::const_iterator iterator;
7
8       // Print the list elements.
9       cout<<"Data: ";
10      iterator end=ds.end();
11      for (iterator it=ds.begin(); it!=end; it++) cout<
12      cout<<endl;
13  }
14
15  int main()
16  {
```


☑ Instead of an std::vector, you can pass an std::list or any other data structure that supports input iterators.
☐ For the typedef we can also use *list<T>::iterator* instead of *const_iterator*.
☐ To make this code more flexible, you can change the *Print()* function to accept two iterators.

Question 9: Which statement is false about iterators?

⊘ A forward iterator can read and write the current position multiple times. You can read what you just wrote.
○ An output iterator can only write to the current position once and must then be incremented. The current position cannot be read from.
○ An input iterator can only read from the current position once and must then be incremented. The current position cannot be written to.
○ On a random access iterator you can use the square bracket operator [] to access elements a few steps before or after the current iterator position.

Question 10: Which statement is false about the following code?

C++:

```
1  // Predicate determining if the value satisfies a cri
2  struct Predicate
3  {
4      bool operator()(int v)
5      {
6          return (v%2)>0;
7      }
8  };
9
```

```
10  int main()
11  {
12      vector<int> v(5);
13      v[0]=10; v[1]=14; v[2]=9; v[3]=15; v[4]=8;
14
15      // Find the first number satisfying the given cri
```

○ Instead of a class with an operator round bracket (function object) we can also pass a global function to the *find_if()* function.
○ The same *Predicate* struct can be used when finding elements in a list<int> instead of a vector<int>.
○ The predicate determines what element will be found.
✓ This code finds the first even number in the vector.

Score:	3 (30.00%)
Pass/Fail:	Failed

Quiz ›