Quiz ›

# Review answers

| | |
|---|---|
| Start date: | 4 minutes ago |
| Complete date: | A moment ago |
| Question 1: | Which statement is <u>false</u> about the following code? |

```C++
// Predicate determining if the value satisfies a cri
struct Predicate
{
    bool operator()(int v)
    {
        return (v%2)>0;
    }
};

int main()
{
    vector<int> v(5);
    v[0]=10; v[1]=14; v[2]=9; v[3]=15; v[4]=8;

    // Find the first number satisfying the given cri
    vector<int>::iterator result=find if(v begin()  v
```

○ Instead of a class with an operator round bracket (function object) we can also pass a global function to the *find_if()* function.

○ The predicate determines what element will be found.

○ The same *Predicate* struct can be used when finding elements in a list<int> instead of a vector<int>.

⊘ This code finds the first even number in the vector.

| Question 2: | Which statements are <u>false</u> about algorithms? |
|---|---|

☐ Removing algorithms are a special kind of mutating algorithms.

☑ Modifying algorithms can modify the elements of data structures and change the order.

☐ Mutating algorithms change the order of elements but not the elements themself.

☑ STL algorithms accept a start- and end-iterator instead of the complete container.

---

**Question 3:**　Which two statements are <u>false</u> about STL containers?

☑ Vectors generally allocate more memory than needed for the elements it stores.

☐ Arguments STL container operations are checked for correctness.

☑ STL containers cannot store pointers.

☐ Elements stored in an STL container must be copyable (must provide a copy constructor).

---

**Question 4:**　Which statement is <u>false</u> about iterators?

○ A begin iterator points to the first element of a container.

○ A regular pointer in a regular array is also an STL compatible iterator.

⊘ An end iterator points to the last element of a container.

○ To access the data an iterator is pointing to, you must dereference the iterator (*).

---

**Question 5:**　Which statements are <u>true</u> about the following code?

```cpp
// Print the list contents.
template <typename T>
void Print(const T& ds)
{
    // Typedef for the iterator to simplify code.
    typedef T::const_iterator iterator;

    // Print the list elements.
    cout<<"Data: ";
    iterator end=ds.end();
    for (iterator it=ds.begin(); it!=end; it++) cout<
    cout<<endl;
}

int main()
{
```

☐ 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*.

☑ Instead of *it!=end* we can also use *it<end* because the iterator of vector supports the < operator

☑ To make this code more flexible, you can change the *Print()* function to accept two iterators.

---

**Question 6:**  Which statement is <u>false</u> about iterators?

○ An input iterator can only read from the current position once and

QuantNet          Z ✉ 🔔 ⚡ Q

times. You can read what you just wrote.

⊘ 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.

○ An output iterator can only write to the current position once and must then be incremented. The current position cannot be read from.

---

**Question 7:**  Which statement is <u>false</u> about 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.

○ 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.

---

**Question 8:**  Which statement is <u>true</u> about predicates?

⊘ A predicate is a functor (function object or global function) that returns a boolean.

○ 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 (function object or global function) that changes an element.

○ A predicate is a brand of dog food.

---

**Question 9:**  Which statement is <u>false</u> about sequence containers?

⊘ Sequence containers order their data.
◯ 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 store their data linearly.

| Question 10: | Which statement is <u>false</u> about sequence containers? |
|---|---|
| | ⊘ The *std::list* class supports the square bracket [] operator.<br>◯ Inserting elements in an *std::list* is faster than inserting elements in an *std::vector*.<br>◯ 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.<br>◯ Traversing to a certain element in a vector is faster than in a list. |
| Score: | 6 (60.00%) |
| Pass/Fail: | Failed |

**Quiz** >