Question 1

Which statement is false about sequence containers?

1.  Sequence containers order their data.
2.  The *list<T>::pop\_front()* function does only remove the first element but does not return the first element.
3.  Sequence containers store their data linearly.
4.  A vector is like an array that can add elements at the end but not at the beginning of the array.

Question 2

Which statement is false about sequence containers?

1.  The *std::list* class supports the square bracket [] operator.
2.  Inserting elements in an *std::list* is faster than inserting elements in an *std::vector*.
3.  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.
4.  Traversing to a certain element in a vector is faster than in a list.

Question 3

Which two statements are false about STL containers?

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

Question 4

Which statement is false about iterators?

1.  An end iterator points to the last element of a container.
2.  A regular pointer in a regular array is also an STL compatible iterator.
3.  A begin iterator points to the first element of a container.
4.  To access the data an iterator is pointing to, you must dereference the iterator (\*).
5. You answered this question correctly.

Question 5

Which statement is false about iterators?

1.  An input iterator can only read from the current position once and must then be incremented. The current position cannot be written to.
2.  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.
3.  A forward iterator can read and write the current position multiple times. You can read what you just wrote.
4.  An output iterator can only write to the current position once and must then be incremented. The current position cannot be read from.

Question 6

Which statement is false about iterators?

1.  With the correct use of iterators (in combination with templates) you can write functions that work with every STL container.
2.  Stream iterators are adapters that allows us to use a stream as source or destination in code that uses iterators.
3.  On iterators you can only use the pre-increment (++it) operator and not the post-increment (it++).
4.  Insert iterators are adaptors that transform an assignment (\*it=value) to an insert, push\_back or push\_front operation on a container.
5. You answered this question correctly.

Question 7

Which statements are true about the following code?

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;
8. // Print the list elements.
9. cout<<"Data: ";
10. iterator end=ds.end();
11. for (iterator it=ds.begin(); it!=end; it++) cout<<\*it<<", ";
12. cout<<endl;
13. }
15. int main()
16. {
17. vector<int> v(5);
18. v[0]=10; v[1]=14; v[2]=9; v[3]=15; v[4]=8;
19. Print(v);
20. }

1.  To make this code more flexible, you can change the *Print()* function to accept two iterators.
2.  Instead of *it!=end* we can also use *it<end* because the iterator of vector supports the < operator
3.  For the typedef we can also use *list<T>::iterator* instead of *const\_iterator*.
4.  Instead of an std::vector, you can pass an std::list or any other data structure that supports input iterators.
5. You answered this question correctly.

Question 8

Which statements are false about algorithms?

1.  Modifying algorithms can modify the elements of data structures and change the order.
2.  STL algorithms accept a start- and end-iterator instead of the complete container.
3.  Removing algorithms are a special kind of mutating algorithms.
4.  Mutating algorithms change the order of elements but not the elements themself.
5. You answered this question correctly.

Question 9

Which statement is true about predicates?

1.  A predicate is one of the two main parts of a sentence, the other being the subject, which the predicate modifies.
2.  A predicate is a functor (function object or global function) that returns a boolean.
3.  A predicate is a brand of dog food.
4.  A predicate is a (function object or global function) that changes an element.

Question 10

Which statement is false about the following code?

Code (C++):

1. // Predicate determining if the value satisfies a criteria.
2. struct Predicate
3. {
4. bool operator()(int v)
5. {
6. return (v%2)>0;
7. }
8. };
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;
15. // Find the first number satisfying the given criteria.
16. vector<int>::iterator result=find\_if(v.begin(), v.end(), Predicate());
17. cout<<"First number found: "<<\*result<<endl;
18. }

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