

X

[swayam.gov.in](https://swayam.gov.in)[https://swayam.gov.in/nc\\_details/NPTEL](https://swayam.gov.in/nc_details/NPTEL)

rg0102@srmist.edu.in ▾

NPTEL (<https://swayam.gov.in/explorer?ncCode=NPTEL>) » Fundamentals of Object Oriented Programming (course)[Announcements \(announcements\)](#) [About the Course \(preview\)](#) [Q&A \(forum\)](#) [Progress \(student/home\)](#) [Mentor \(student/mentor\)](#)[Review Assignment \(assignment\\_review\)](#) [Course Recommendations New! \(/course\\_recommendations\)](#)

Click to register for  
Certification exam

[https://examform.nptel.ac.in/2025\\_01/ExamForm/Registration](https://examform.nptel.ac.in/2025_01/ExamForm/Registration)

If already registered, click  
to check your payment  
status

Course outline

About NPTEL ()

## Week 9. Assignment 9

The due date for submitting this assignment has passed.

As per our records you have not submitted this assignment.

1) Which of the following is an associative container in STL?

- ☐ vector
- ☐ deque
- ☐ set
- ☐ list

No, the answer is incorrect.

**Due on 2025-03-26, 23:59 IST.**

**1 point**

**How does an NPTEL online course work? ()****Week 0 ()****Week 1 ()****Week 2 ()****Week 3 ()****Week 4 ()****Week 5 ()****Week 6 ()****Week 7 ()****Week 8 ()****Week 9 ()**

- ☐ Introduction to Standard Template Library (unit? unit=61&lesson=103)
- ☐ Associative Containers (unit? unit=61&lesson=104)
- ☐ Unordered Containers, Iterators (unit? unit=61&lesson=105)

Score: 0

Accepted Answers:

*set*

2) Which of the following containers allows bidirectional iterators in STL?

**1 point**

- ☐ vector
- ☐ deque
- ☐ set
- ☐ array

No, the answer is incorrect.

Score: 0

Accepted Answers:

*set*

3) In a C++ program to:

**1 point**

- Create a vector<int> containing the elements 5, 15, 10, 20, 25.
- Use std::max\_element() to find the maximum element.
- Print the maximum element.

What is the output of the program?

- ☐ 25
- ☐ 20
- ☐ Compilation error
- ☐ Undefined behavior

No, the answer is incorrect.

Score: 0

Accepted Answers:

*25*

4) Which of the following is a non-modifying algorithm in STL?

**1 point**

- ☐ std::sort()

- ☐ STL Algorithms (unit? unit=61&lesson=106)
- ☐ Case Studies - Library Management System, Real-Time Stock Tracker (unit? unit=61&lesson=107)
- ☐ **Quiz: Week 9: Assignment 9 (assessment? name=124)**

- ☐ Solution for Week 9 (unit? unit=61&lesson=132)

**Week 10 ()**

**Week 11 ()**

**Week 12 ()**

**Download Videos ()**

**Weekly Feedback ()**

- ☐ std::copy()
- ☐ std::find()
- ☐ std::remove()

No, the answer is incorrect.

Score: 0

Accepted Answers:

*std::find()*

5) In a C++ program to:

- Create a vector<int> containing 50, 10, 40, 30, 20.
- Sort the vector in ascending order using std::sort().
- Print the sorted elements.

**1 point**

What is the correct sequence of output?

- ☐ 10 20 30 40 50
- ☐ 50 40 30 20 10
- ☐ Compilation error
- ☐ Undefined behavior

No, the answer is incorrect.

Score: 0

Accepted Answers:

*10 20 30 40 50*

6) In a C++ program that:

- Creates a map<string, int> to store student names and their marks.
- Adds the entries Alice: 90, Bob: 85, Charlie: 95.
- Prints all entries using an iterator.

**1 point**

What is the correct output?

- ☐ Alice: 90, Bob: 85, Charlie: 95
- ☐ Bob: 85, Charlie: 95, Alice: 90
- ☐ Compilation error

☐ Undefined behavior

No, the answer is incorrect.

Score: 0

Accepted Answers:

*Alice: 90, Bob: 85, Charlie: 95*

7) In a program to:

- Create a set<int> containing the elements 50, 10, 30, 20, 40.
- Use an iterator to traverse the set and print the elements.
- Use std::find() to check if 30 exists in the set.

**1 point**

Which of the following is true?

- ☐ The output is sorted in ascending order, and 30 is found.
- ☐ The output is sorted in descending order, and 30 is not found.
- ☐ The elements are printed in the order they were added.
- ☐ Compilation error.

No, the answer is incorrect.

Score: 0

Accepted Answers:

*The output is sorted in ascending order, and 30 is found.*

8) Which of the following operations is not directly supported by the std::list container in C++ (which implements a doubly linked list)?

**1 point**

- ☐ Insertion of elements at the front of the list
- ☐ Insertion of elements at the back of the list
- ☐ Direct access to elements by index
- ☐ Removal of elements from the front of the list

No, the answer is incorrect.

Score: 0

Accepted Answers:

*Direct access to elements by index*

9) Consider the following C++ code snippet. What will be the output?

**1 point**

```
#include <iostream>
#include <vector>

int main() {
    std::vector<int> vec = {1, 2, 3, 4, 5};
    std::vector<int>::iterator it = vec.begin();
    std::advance(it, 2);
    *it = 10;
    for (auto i = vec.begin(); i != vec.end(); ++i) {
        std::cout << *i << " ";
    }
    return 0;
}
```

- ☐ 1 2 3 4 5
- ☐ 1 2 10 4 5
- ☐ 1 2 3 10 5
- ☐ 1 2 3 4 10

No, the answer is incorrect.

Score: 0

Accepted Answers:

1 2 10 4 5

10) What will be the output of the following C++ program?

**1 point**

```
#include <iostream>
#include <deque>
using namespace std;

int main() {
    deque<int> dq;
    dq.push_back(10);
    dq.push_front(20);
    dq.push_back(30);
    dq.push_front(40);

    dq.pop_back();
    dq.pop_front();

    for (int x : dq) {
        cout << x << " ";
    }
    return 0;
}
```

- ☐ 10 30
- ☐ 20 10
- ☐ 20 30
- ☐ 10 20

No, the answer is incorrect.

Score: 0

Accepted Answers:

20 10