X

(https://swayam.gov.in)

(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/exam_form/dashboard)

If already registered, click to check your payment status

Course outline

About NPTEL ()

Week 8: Assignement 8

The due date for submitting this assignment has passed.

Due on 2025-03-19, 23:59 IST.

Assignment submitted on 2025-03-11, 22:10 IST

1) To write a template function in C++ to find the maximum of two values. Which of the following implementations is correct? ***1 point***

● template<typename T> T max(T a, T b) { return (a > b) ? a : b; }
○ template<typename T> void max(T a, T b) { return (a > b) ? a : b; }
○ template<int T> T max(T a, T b) { return (a > b) ? a : b; }

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 ()

○ Introduction to Templates and Generics (unit?unit=60&lesson=86)

○ Template Class in C++ (unit?unit=60&lesson=87)

○ Generics in Java (unit?unit=60&lesson=88)

○ Generics in Java [contd.] (unit?unit=60&lesson=89)

○ Generics in Python (unit?unit=60&lesson=90)

○ template<typename T> T max(int a, int b) { return (a > b) ? a : b; }

Yes, the answer is correct.
Score: 1

Accepted Answers:
template<typename T> T max(T a, T b) { return (a > b) ? a : b; }

2) Which of the following statements is true about template classes in C++? ***1 point***

○ Template classes cannot have non-template member functions.

● Template classes can be instantiated for any data type.

○ Template classes must inherit from a base class.

○ Template classes cannot have constructors.

Yes, the answer is correct.
Score: 1

Accepted Answers:
Template classes can be instantiated for any data type.

3) To write a C++ program to define a template class Box with the following: ***1 point***

- A private member to store data.
- A constructor to initialize the data.
- A method getData() to return the stored data.

Which of the following correctly instantiates the template class?

● Box<int> intBox(42);

○ Box<int> intBox;

○ Box<int, float> intBox(42);

○ Box stringBox("Hello");

Yes, the answer is correct.
Score: 1

Accepted Answers:
Box<int> intBox(42);

- **Quiz: Week 8: Assignement 8 (assessment? name=101)**
- Solution for Week 8 (unit? unit=60&lesson=129)

Week 9 ()

Week 10 ()

Week 11 ()

Week 12 ()

Download Videos ()

Weekly Feedback ()

4) What is the primary purpose of generics in Java? *1 point*

○ To allow multiple inheritance.
○ To perform runtime type checking.
● To enforce compile-time type safety.
○ To improve code execution speed.

Yes, the answer is correct.
Score: 1

Accepted Answers:
To enforce compile-time type safety.

5) To write a generic class in Java called Container with the following: *1 point*

- A private attribute value of generic type T.
- A constructor to initialize value.
- A method getValue() to return the stored value.

Which of the following correctly defines the class?

● class Container<T> { private T value; }
○ class Container { private T value; }
○ class Container<T, U> { private T value; }
○ class Container(T) { private T value; }

Yes, the answer is correct.
Score: 1

Accepted Answers:
class Container<T> { private T value; }

6) To write a Python function get first element() using generics that: *1 point*

- Accepts a list of any type.
- Returns the first element of the list.

Which of the following correctly defines the function?

○ def get first element(lst): return lst[0]

●

from typing import List, TypeVar

T = TypeVar(’T’)

def get first element(lst: List[T]) -> T: return lst[0]

○ def get first element(lst: List[T]) -> T: return lst[0]

○

from typing import TypeVar

T = TypeVar(’T’)

def get first element(lst) -> T: return lst[0]

Yes, the answer is correct.
Score: 1

Accepted Answers:

from typing import List, TypeVar

T = TypeVar(’T’)

def get first element(lst: List[T]) -> T: return lst[0]

7) Which of the following is a common use case for templates in C++ and generics in Java? ***1 point***

● Creating reusable and type-safe data structures.

○ Enforcing runtime type checking.

○ Avoiding the use of constructors.

○ Improving execution time by avoiding pointers.

Yes, the answer is correct.
Score: 1

Accepted Answers:

Creating reusable and type-safe data structures.

8) Which of the following is true about templates in C++ and generics in Java? ***1 point***

○

Templates in C++ are resolved at runtime, while generics in Java are resolved at compile time.

● Generics in Java use type erasure, while templates in C++ do not.

○ Both templates and generics support multiple inheritance.

○ Templates and generics are identical in their implementation.

Yes, the answer is correct.
Score: 1

Accepted Answers:
Generics in Java use type erasure, while templates in C++ do not.

9) Which of the following is not an advantage of using templates in C++? ***1 point***

● Dynamic Memory Allocation

○ Code Reusability

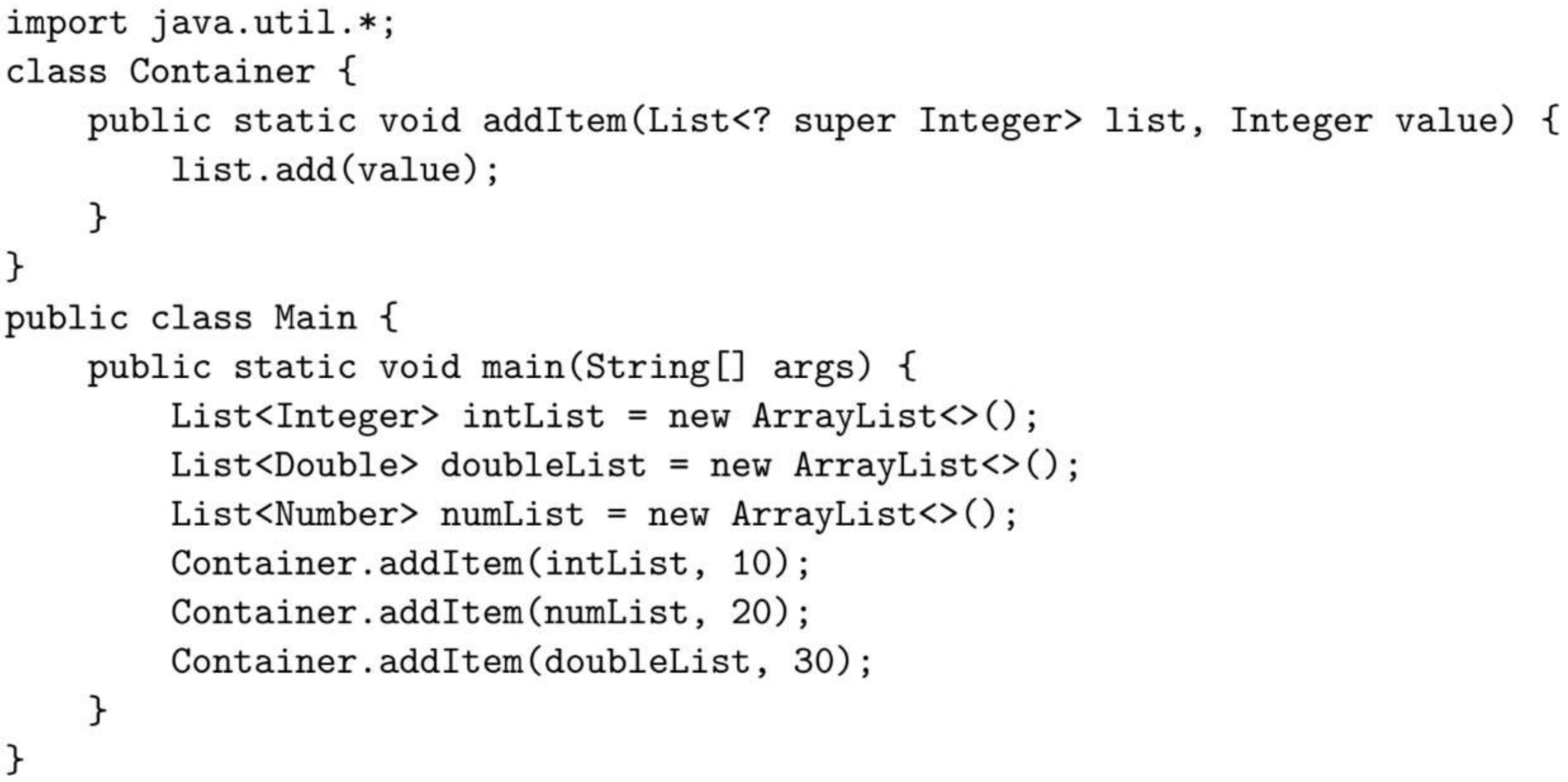
○ Performance Efficiency

○ Type Safety

Yes, the answer is correct.
Score: 1

Accepted Answers:
Dynamic Memory Allocation

10) Consider the following Java code snippet: ***1 point***



```
import java.util.*;
class Container {
    public static void addItem(List<? super Integer> list, Integer value) {
        list.add(value);
    }
}
public class Main {
    public static void main(String[] args) {
        List<Integer> intList = new ArrayList<>();
        List<Double> doubleList = new ArrayList<>();
        List<Number> numList = new ArrayList<>();
        Container.addItem(intList, 10);
        Container.addItem(numList, 20);
        Container.addItem(doubleList, 30);
    }
}
```

○ The code will compile and print the contents of the lists.

○ The code will compile but it will throw a runtime exception.

● The code will fail to compile because List<Double> is incompatible with List<? super Integer>.

○ The code will fail to compile because List<? super Integer> cannot accept Integer values.

Yes, the answer is correct.

Score: 1

Accepted Answers:

The code will fail to compile because List<Double> is incompatible with List<? super Integer>.