**Object-Oriented programming**

Lab #6 – **Generic classes and methods**

**I. Get familiar with generic types** Given the following class

public class MyPair<T, U> { public final T Fst; public final U Snd;

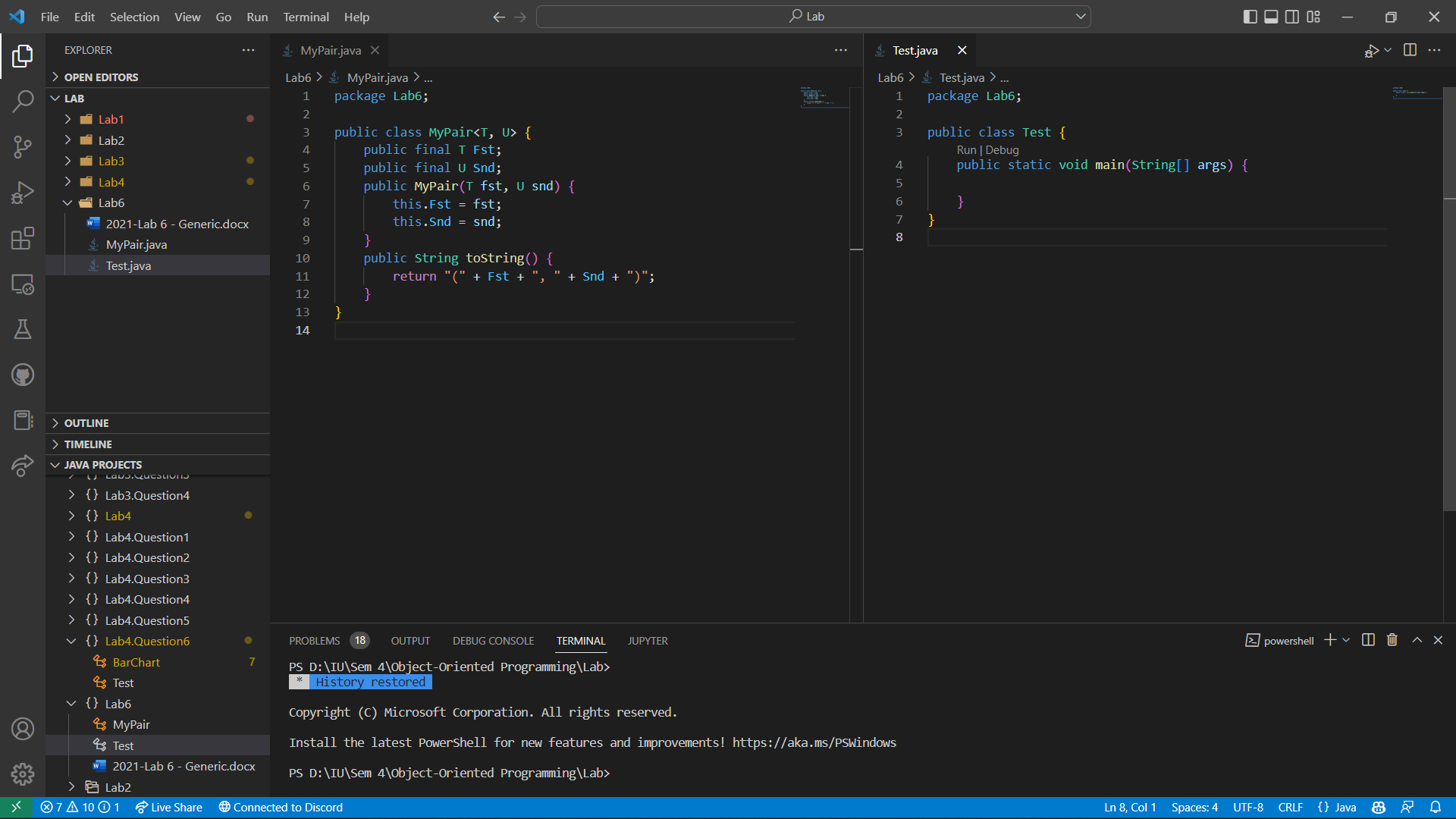
public MyPair(T fst, U snd) { this.Fst = fst; this.Snd = snd;

}

public String toString() { return "(" + Fst + ", " + Snd + ")"; }

}

1. In a new source file, write a Java program that includes this declaration and a class with an empty Main method. Compile it to check that the program is well-formed.



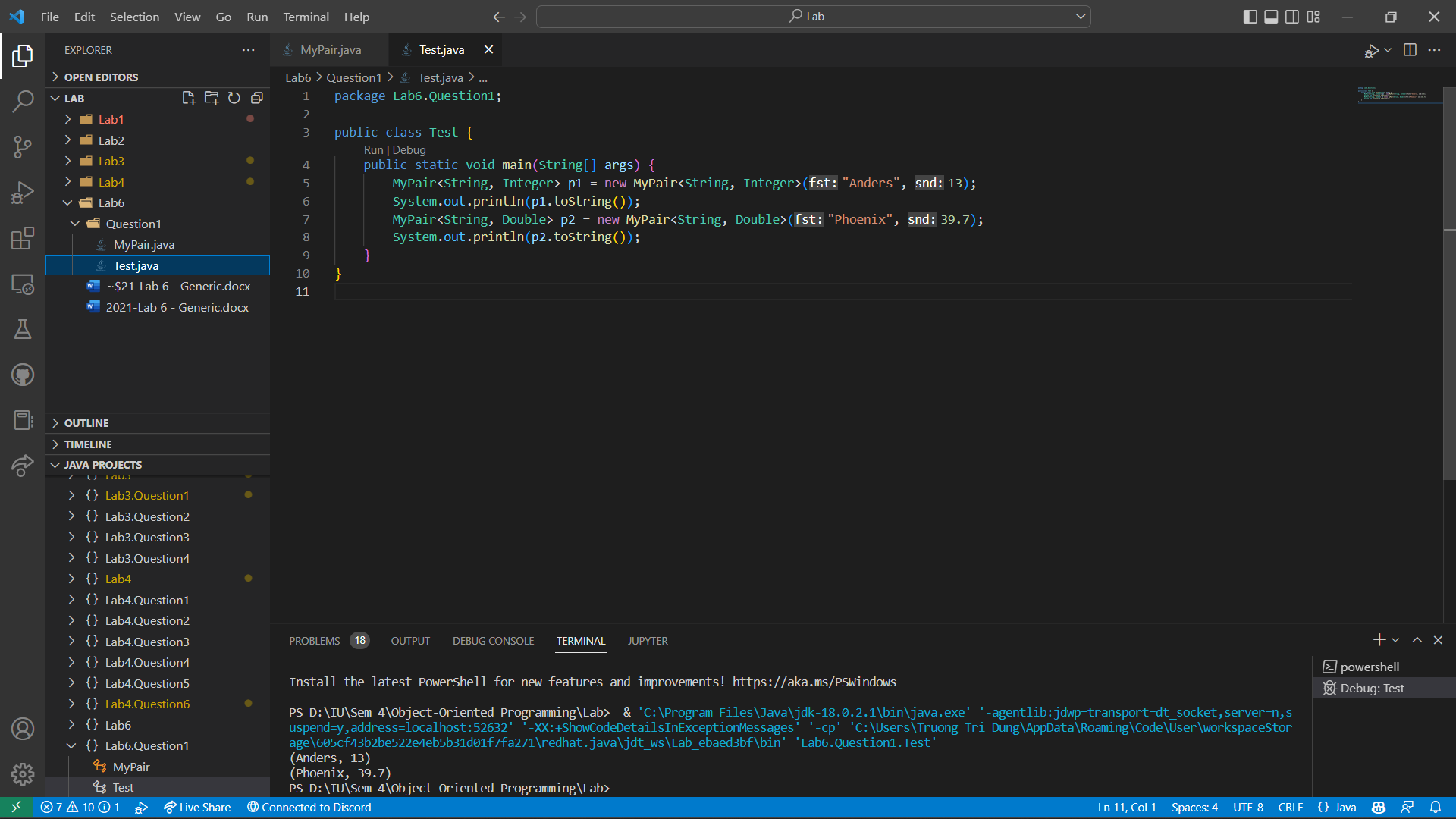
1. Declare a variable of type MyPair<String, Integer> and create some values, for instance new MyPair<String, Integer>("Anders", 13), and assign them to the variable.

A screenshot of a computer

Description automatically generated with medium confidence

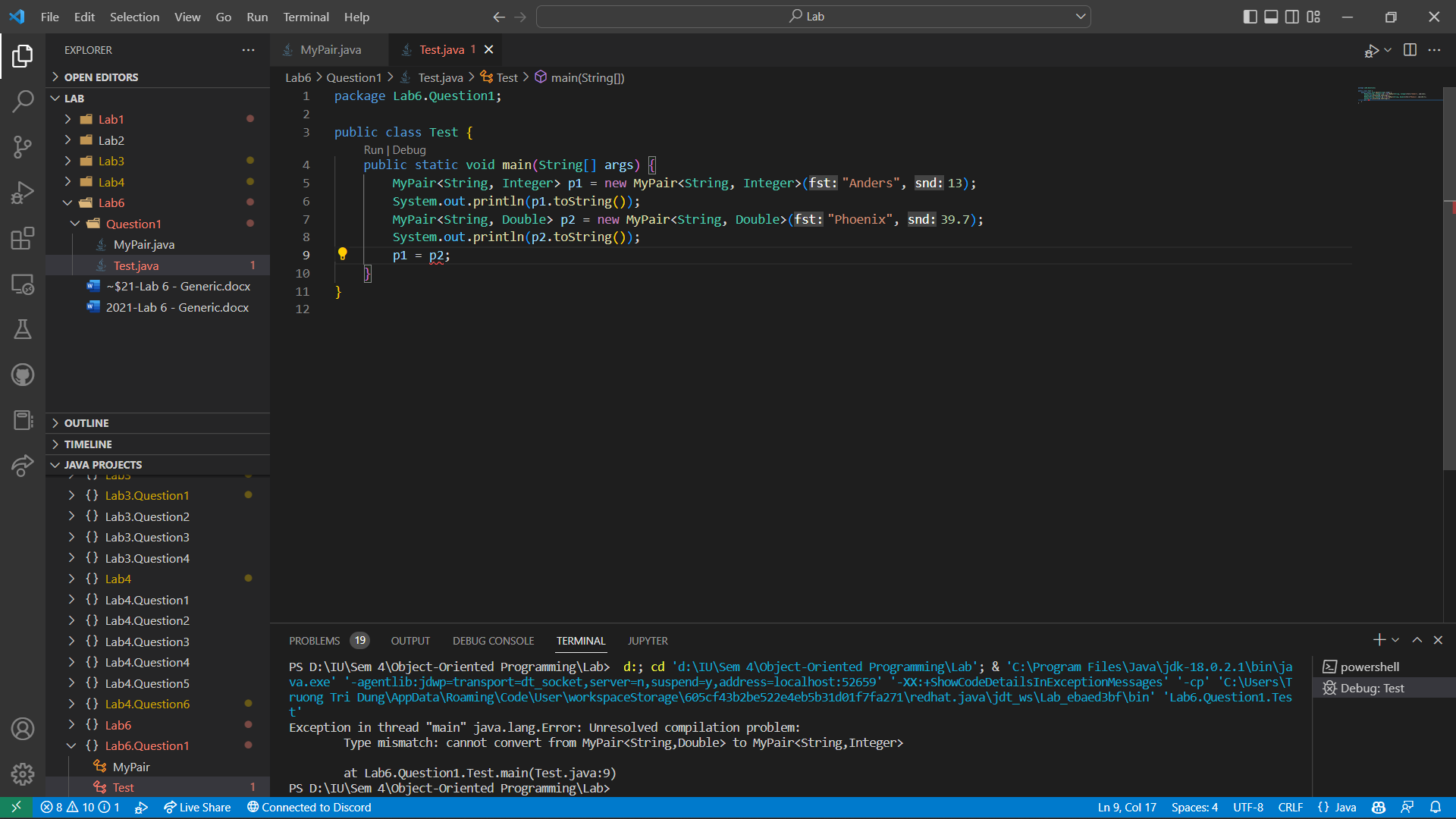
1. Declare a variable of type MyPair<String, Double>. Create a value such as new

MyPair<String, Double>("Phoenix", 39.7) and assign it to the variable.



1. Can you assign a value of type MyPair<String, Double> to a variable of type

MyPair<String, Integer>? Should this be allowed?



Cannot assign a value of type MyPair<String, Double> to a variable of type

MyPair<String, Integer> because the compiler cannot convert from MyPair<String, Double> to MyPair<String, Integer>. It shouldn’t be allowed because data type diversity is characteristic of Generics type but in some case, it should be allowed, sush as the case we convert from any type to type <String>. That is easier to string handling.

1. Declare a variable grades of type MyPair<String, Integer>[], create an array of length 5 with element type MyPair and assign it to the variable. Create a few MyPairs and store them into grades[0], grades[1] and grades[2].

A screenshot of a computer

Description automatically generated with medium confidence

1. Use the foreach statement to iterate over grades and print all its elements. What are the values of those array elements you did not assign anything to?

A screenshot of a computer

Description automatically generated with medium confidence

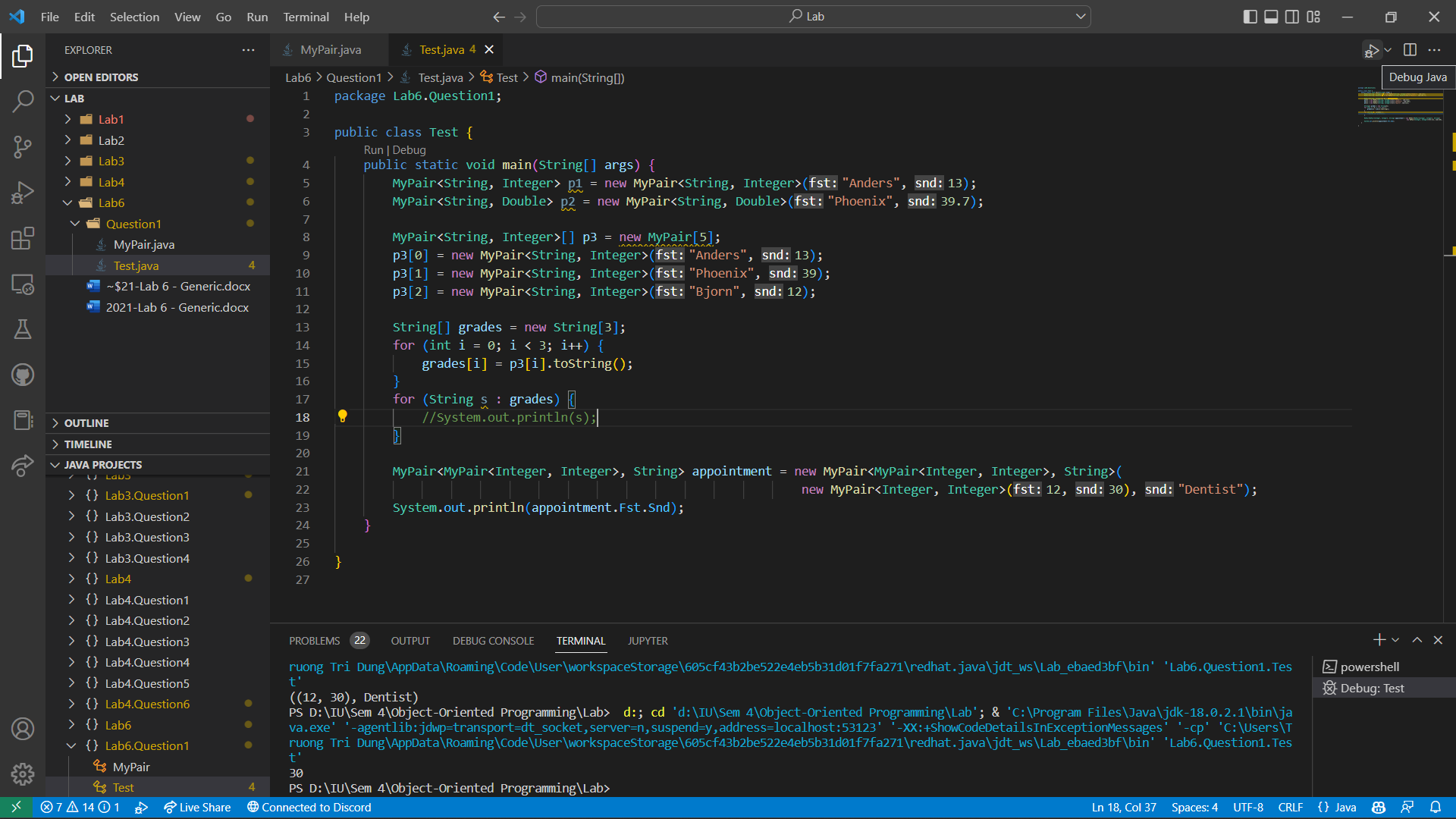
The value is null if we did not assign anything to

1. Declare a variable appointment of type

MyPair<MyPair<Integer, Integer>, String>

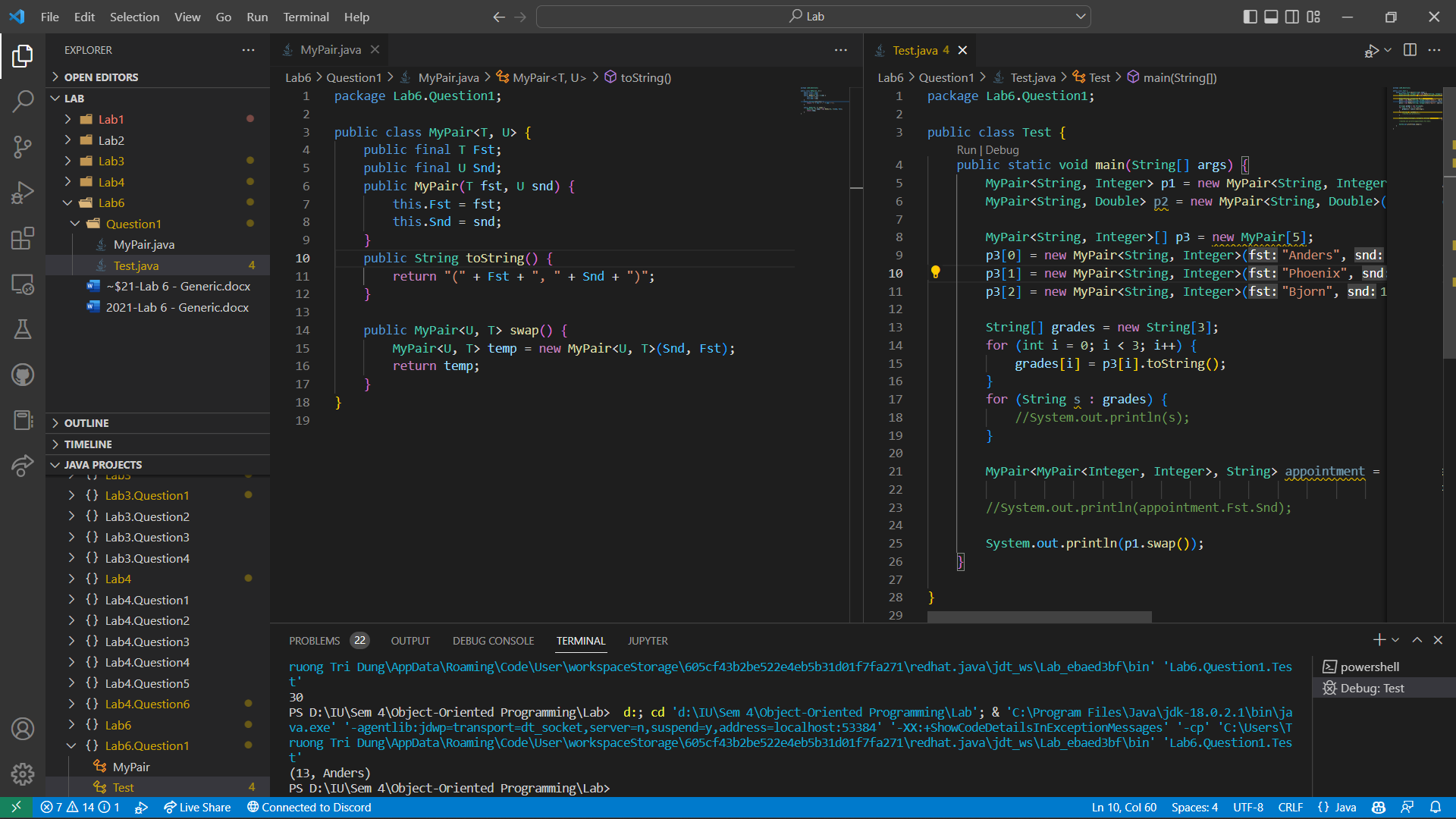
and create a value of this type and assign it to the variable.

What is the type of appointment.Fst.Snd? This shows that a type-argument may itself be a constructed type.



The type of appointment.Fst.Snd is interger. We call appointment.Fst it return MyPair<Integer, Integer> then, we call appointment.Fst.Snd same as MyPair<Integer, Integer>.Snd it return Interger

1. Declare a method Swap() in MyPair<T, U> that returns a new value of type MyPair in which the components have been swapped.



**II. Differences between Object, generic and generic raw types**

In JAVA, there is a Map data structure that helps developers to link a key to a value.

Read about Map in

* <https://www.geeksforgeeks.org/map-interface-java-examples/>
* <https://www.geeksforgeeks.org/java-util-hashmap-in-java/>
* <https://www.tutorialspoint.com/java/java_hashmap_class.htm>

As we want to reimplement the Map class, implement a class named MyMap to manage (store and get back) any object by its ID.

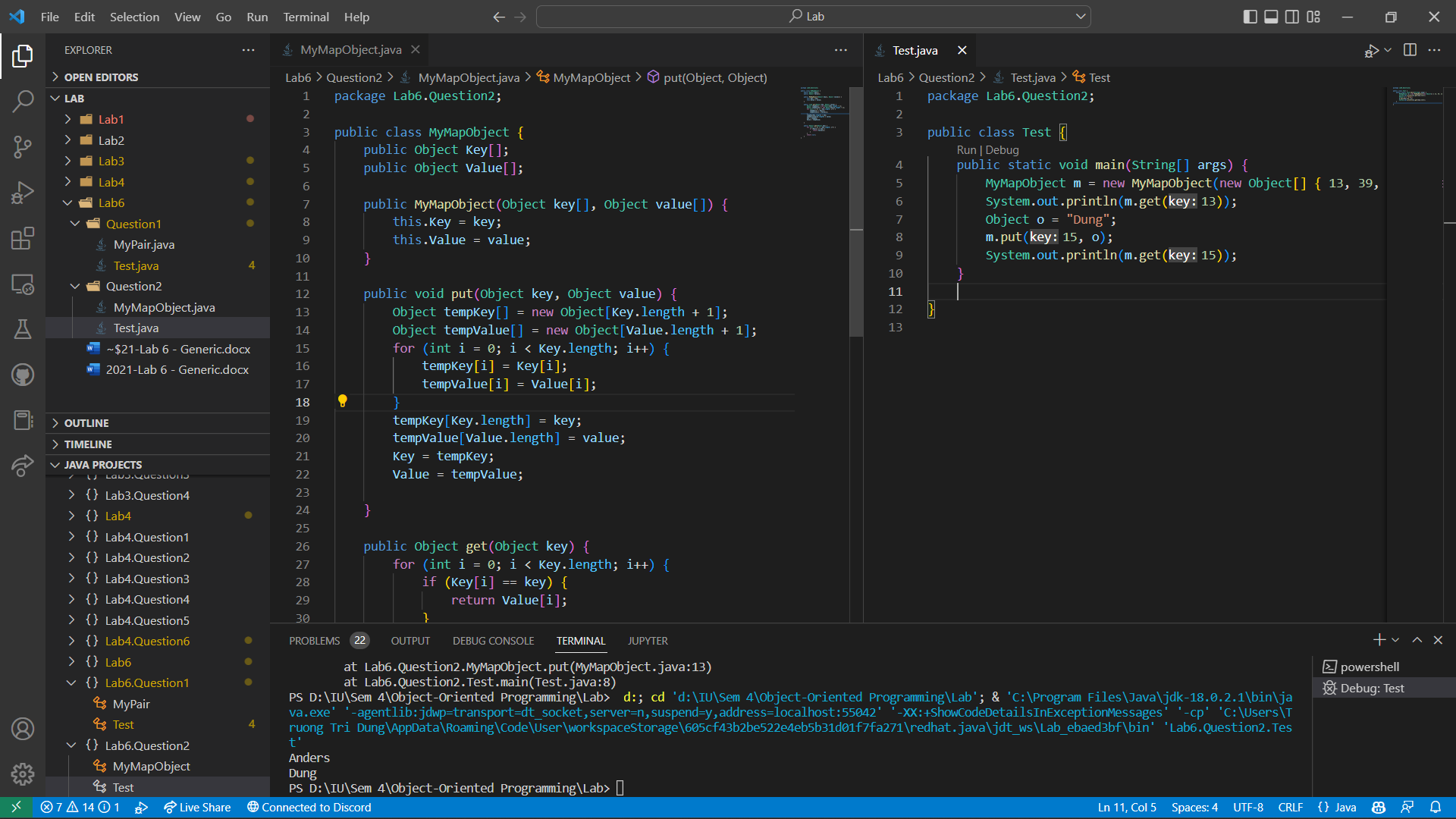
* User can put an object ***obj*** to a Map ***m*** by calling

*m.put(obj.getID(), obj);*

* User can get back an object from the map ***m*** by invoking

# m.get(id);

1. Implement the MyMap in two different ways:
   1. Use Object as the type for both the Key and the Value parameters of the ***put*** and ***get*** methods



* 1. Use generic type

Graphical user interface, text

Description automatically generated

1. With your implementations, write a main function to
   1. Test these two implementations

A screenshot of a computer

Description automatically generated

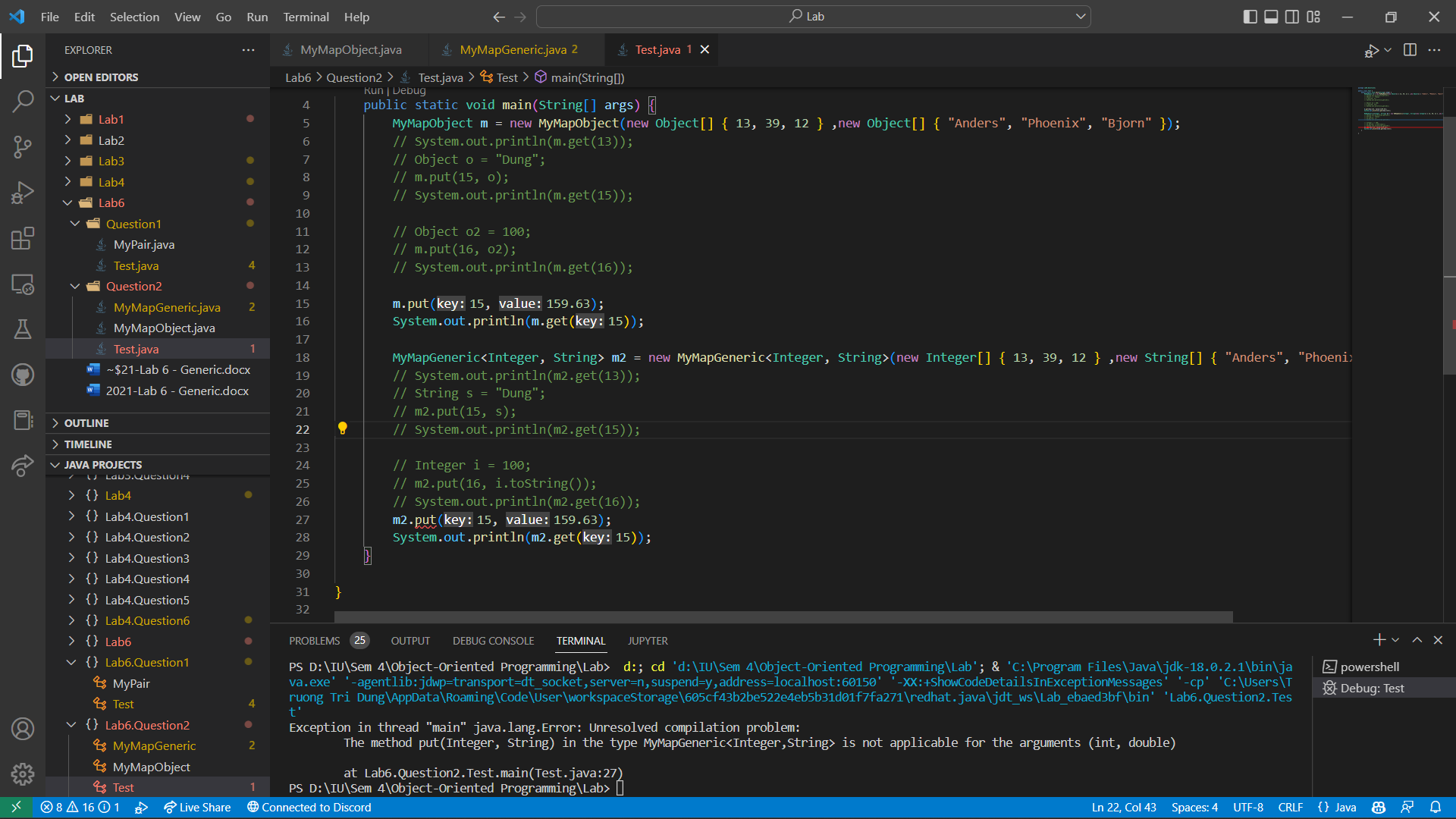
* 1. To show advantage of generic type over Object

A screenshot of a computer

Description automatically generated

The advantage of generic is well-defined data type. We need to convert any type to possible type before put them into generic type.

* 1. To show advantage of parameterized type over generic raw type



The advantage of parameterized type is we can assign any type of data into the class

Reference: textbook “Java How to Program”, chapter 20