

PRE-LAB

1.What are the Advantages of Collections and Generics?

Solution:

1. Type Safety: Generics ensure compile-time type safety, preventing runtime type errors. With collections, this means you can define a collection to hold only a specific type of objects, reducing the risk of adding the wrong type of object.
2. Code Reusability: Both collections and generics promote code reusability. Generics allow you to write classes, interfaces, and methods that operate on a specific type, making them more flexible and reusable. Collections provide a set of data structures that can be reused in different parts of the codebase.
3. Performance: Generics in Java can improve performance by eliminating the need for casting and by enabling the compiler to perform more optimizations. Collections also offer efficient data structures that can improve the performance of operations like insertion, deletion, and search.
4. Readability and Maintainability: Generics and collections can make code more readable and maintainable. They provide a way to express the intent of the code more clearly, making it easier for other developers to understand and modify the code.
5. Safety and Integrity: Using generics and collections can help ensure the integrity of your data. Generics prevent the mixing of different types, while collections offer data structures with built-in checks and operations that maintain the consistency of the data.

IN-LAB:

TASK1: To create generic type **CustomArray** – one dimensional array with random index range

CustomArray is a collection – array of random type values with fixed length and with original index that is specified by user.

Example 1: array of 20 elements length, array values – symbols, index starts with 18. Example 2: array of 10 elements length, array values – objects of class Animals, index starts with -5

Values of random type can be located in array, custom first index and the number of elements in array should be specified while creating. The length and range of indexes cannot be changed after creating. Values of array elements can be set while creating the array and later with the help of indexer.

Initial and finite index, array length, array elements in the form of standard array Array that starts with 0 can be obtained from array.

CustomArray should be able to use operator foreach and other constructions that are oriented to the presence of enumerator in class.

The task has two levels of complexity: Low and Advanced.

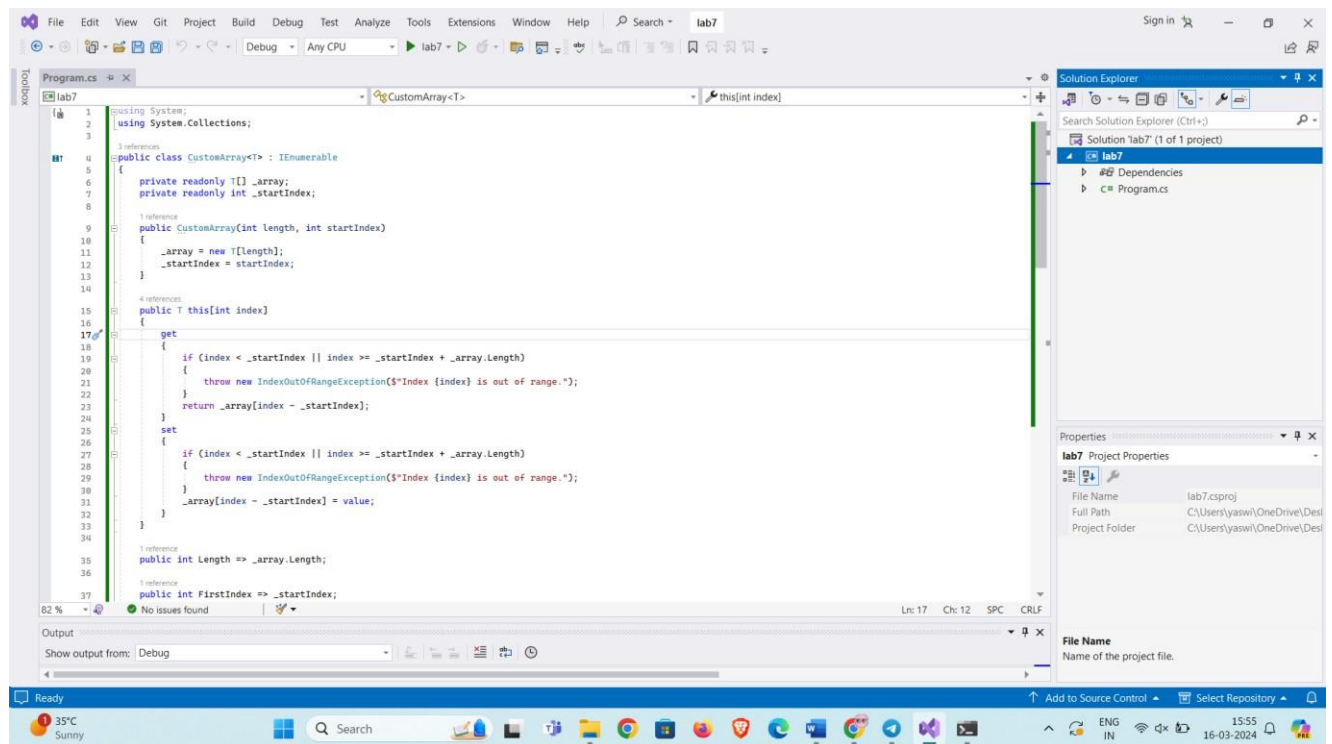
Low level tasks require implementation of the following functionality:

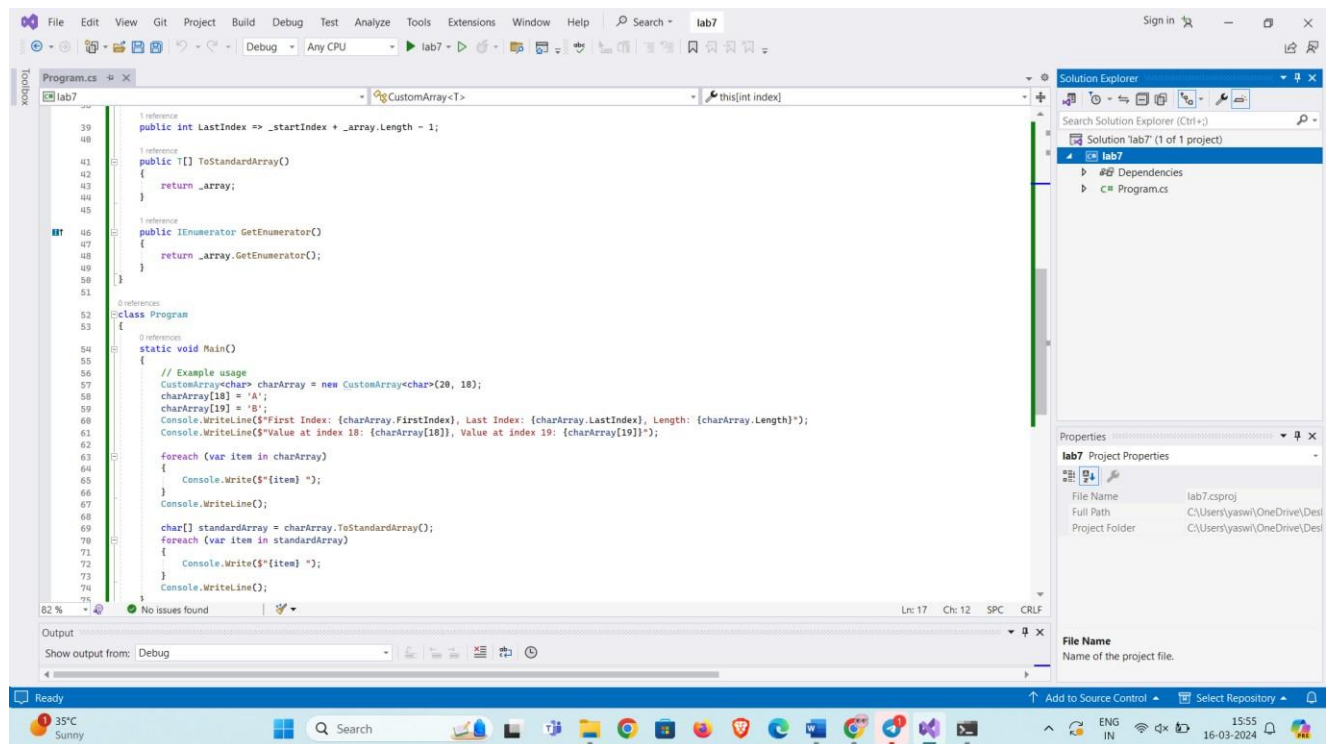
- Creating of empty user array and the one based on standard existing
- Receiving first, last indexes, length, values in form of standard array with 0.
- Access to writing and reading element based on predetermined correct index

Advanced level tasks require implementation of the following functionality:

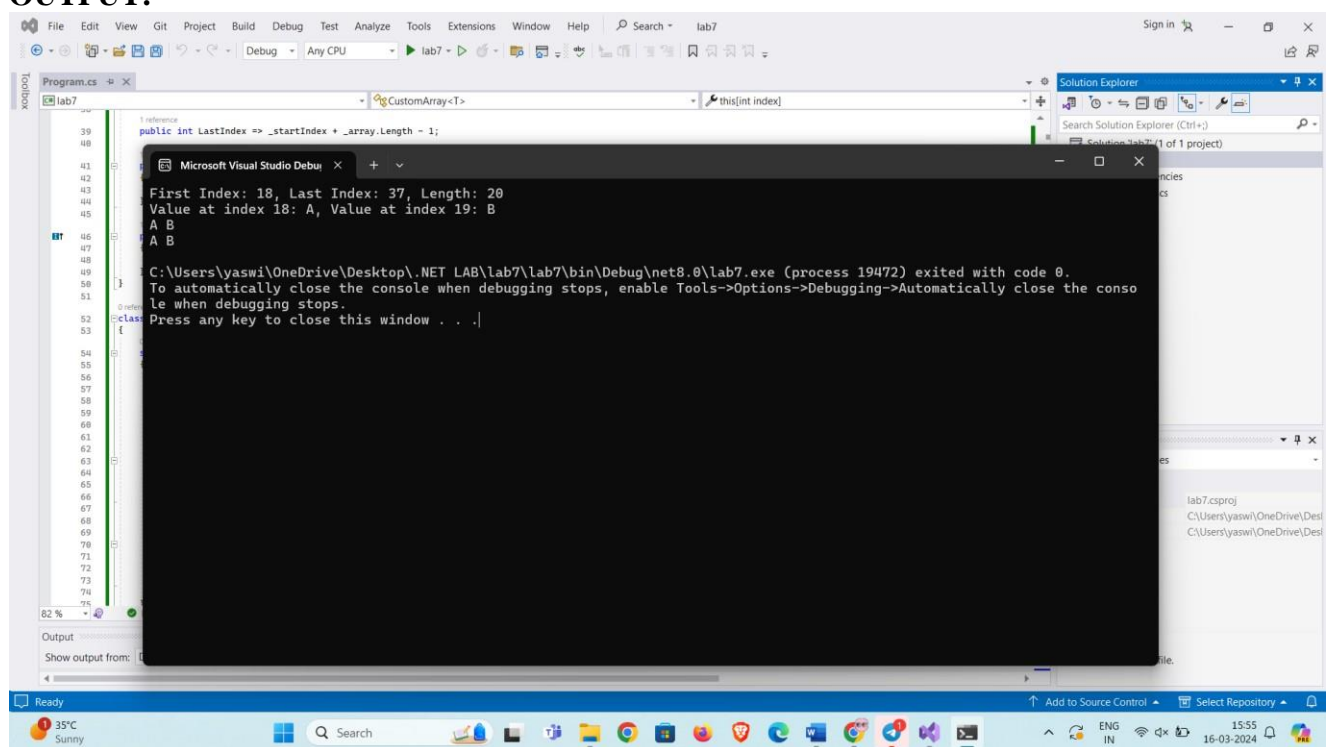
- All completed tasks of Low level
- Creating of array based on values params
- Generating exceptions, specified in xml-comments to class methods
- Receiving numerator from array for operator foreach

Solution:





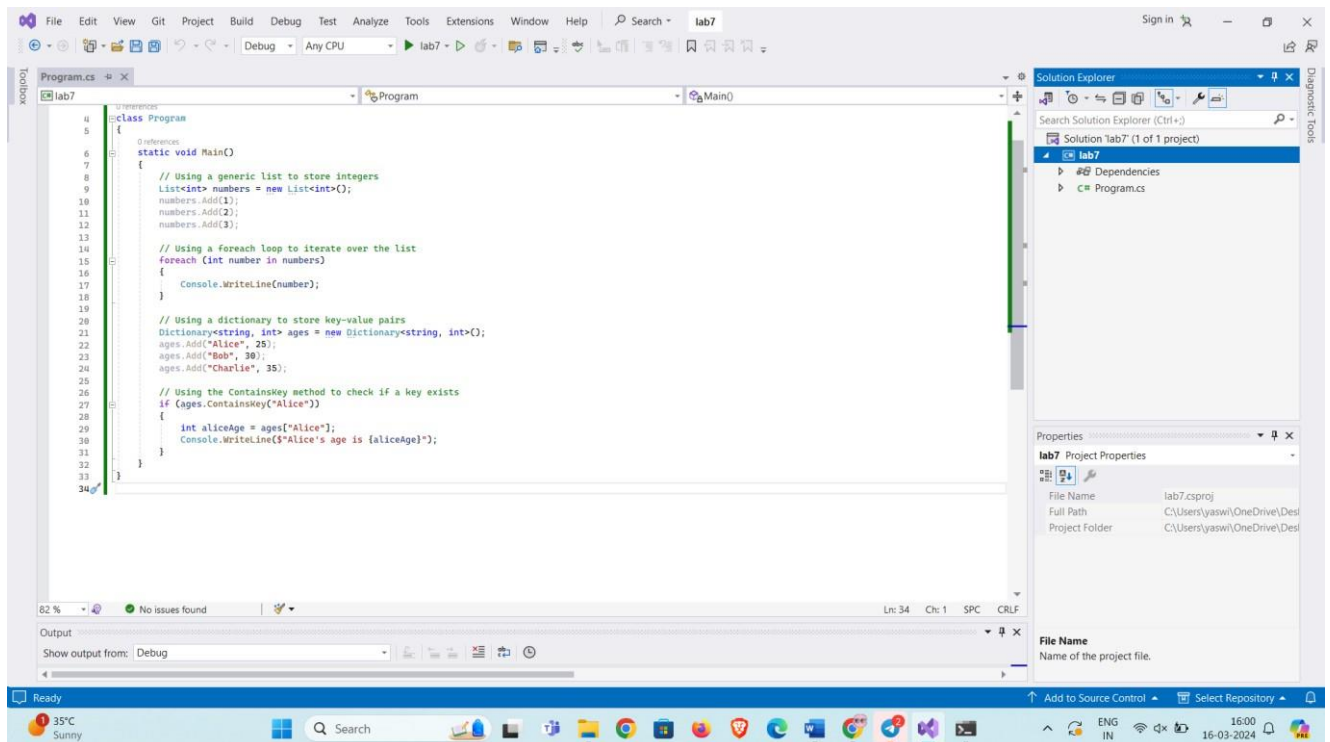
OUTPUT:



POST-LAB

1. Why and How the Collections and Generics are helpful to develop an Application?

Solution:



Output:

