Experiment No: 10

AIM: To learn recursion and generics.

Date:

CO mapped: CO-4

Objectives:

a) To develop a deep understanding of recursion and generics in programming. Mastery of recursion will enable the development of elegant and efficient algorithms for solving complex problems. Understanding generics will facilitate the creation of flexible, reusable, and type-safe code in various programming languages.

b) Learning recursion and generics is crucial for building efficient algorithms and writing more versatile and type-safe code in software development. Achieving this objective will help you become a more proficient and well-rounded programmer.

Background:

Recursion in java is a process in which a method calls itself continuously. A method in java that calls itself is called the recursive method.
Java Generics programming is introduced in J2SE 5 to deal with type-safe objects. It makes the code stable by detecting the bugs at compile time. Before generics, we can store any type of object in the collection, i.e., non-generic. Now generics force the java programmer to store a specific type of object.

Practical questions:

1. Write a recursive method that converts a decimal number into a binary number as a string. The method header is: public static String dec2Bin(int value)

Write a test program that prompts the user to enter a decimal number and displays its binary equivalent.

2. Write the following method that returns a new ArrayList. The new list contains the non-duplicate elements from the original list.

public static <E>ArrayList<E>removeDuplicates(ArrayList<E> list)

3. Implement the following method using binary search.

public static <E extends Comparable<E>>

intbinarySearch(E list, E key)

Observations: Put Output of the program

Conclusion: (Sufficient space to be provided)

Quiz:

1. What is recursion in Java, and how does it differ from iteration in solving problems?
2. What are the advantages and disadvantages of using recursion in Java?
3. What are generics in Java, and why are they used for creating parameterized types?
4. How to define Generic class? What are restrictions of generic programming?
5. Can you provide an example of a generic class in Java, such as a generic ArrayList?

Suggested Reference:

1. https://www.tutorialspoint.com/java/
2. https://www.geeksforgeeks.org/
3. https://www.w3schools.com/java/
4. https://www.javatpoint.com/