Name : Faricha Aulia

Absen : 08

Class : 1I IT

Lab Unit 1

1. In step no 1, why all types of data can be accommodated into an Arraylist?
   Answer: Because, the array list does not need to be declared with the number of elements. However, it can add any number of elements to the limit of an integer value that is an index or machine memory capacity.
   Example : ArrayList<T> array_name = new ArrayList<>();
   That is, we declare an array list of T (this T can be replaced with any TypeWrapper type you want such as Integer, String, Double, etc.). The constructor can contain an initial capacity value, but this is optional.
2. Modify the program code in step no 1 so that only one type of data is accommodated or specific data types!
   Answer: ArrayList identifier = new ArrayList();

Lab Unit 2

1. What is the difference between the push() and add() methods on the fruits object?
   Answer:
   - Push will give them a hint that they are using the object.
   - The add method only adds data

Lab Unit 3

1. In the add() method that uses unlimited arguments, what concept does it use? And what are the advantages?
   Answer:
2. In the linearSearch() method above, please replace it with the binarySearch() method from the collection!

```
Answer: class BinarySearch {

    int binarySearch(int arr[], int l, int r, int x)

    {
        if (r >= l) {
            int mid = l + (r - l) / 2;
            if (arr[mid] == x)
                return mid;
            if (arr[mid] > x)
                return binarySearch(arr, l, mid - 1, x);
                return binarySearch(arr, mid + 1, r, x);
        }
        return -1;
    }

    public static void main(String args[])
    {
        BinarySearch ob = new BinarySearch();
        int arr[] = { 2, 3, 4, 10, 40 };
        int n = arr.length;
        int x = 10;
        int result = ob.binarySearch(arr, 0, n - 1, x);
        if (result == -1)
            System.out.println("Element not present");
        else
            System.out.println("Element found at index "
                               + result);
    }
}
```

3. Add an ascending or descending sorting method to the class!
   Answer:

```
int a[] = new int[n];
        System.out.println("Enter all the elements:");
        for (int i = 0; i < n; i++)
        {
            a[i] = s.nextInt();
        }
        for (int i = 0; i < n; i++)
        {
            for (int j = i + 1; j < n; j++)
            {
                if (a[i] > a[j])
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.print("Ascending Order:");
        for (int i = 0; i < n - 1; i++)
        {
            System.out.print(a[i] + ",");
        }
        System.out.print(a[n - 1]);
```