

Mobile Application Development Lab

CSL-341

Lab Journal



Student Name: Hassan Ahmed

Enrollment No: 01-134222-065

Section: BSCS(6-B)

Department of Computer Science
BAHRIA UNIVERSITY ISLAMABAD

Lab 2 – Dart Introduction

Objectives:

Basic syntax of Dart programming language

Tools Used:

VS Code

Submission Date:

Evaluation

Signatures of Lab Instructor

TASK 1:

Find the largest number in a given list.

Solution:

```
1 int largestnum(var a){
2     int search = a[0];
3     for (int i = 1 ; i < a.length; i++){
4         if(a[i]>search){
5             search = a[i];
6         }
7     }
8     return search;
9 }
10 void main(){
11     var a = [10,20,99,30,40];
12     var largestnumber = largestnum(a);
13     print(largestnumber);
14 }
```

Output:

```
99
```

TASK 2:

Use merge sort to sort a List.

Solution:

```
1 void main() {
2     List<int> numbers = [38, 27, 43, 3, 9, 82, 10];
3
4     print("Unsorted List: $numbers");
5     List<int> sortedList = mergeSort(numbers);
6     print("Sorted List: $sortedList");
7 }
8
9 List<int> mergeSort(List<int> list) {
10     if (list.length <= 1) {
11         return list;
12     }
13
14     int mid = list.length ~/ 2;
15     List<int> left = mergeSort(list.sublist(0, mid));
16     List<int> right = mergeSort(list.sublist(mid));
17
18     return merge(left, right);
19 }
20
21 List<int> merge(List<int> left, List<int> right) {
22     List<int> result = [];
```

```
23     int i = 0, j = 0;
24
25     while (i < left.length && j < right.length) {
26         if (left[i] < right[j]) {
27             result.add(left[i]);
28             i++;
29         } else {
30             result.add(right[j]);
31             j++;
32         }
33     }
34
35     while (i < left.length) {
36         result.add(left[i]);
37         i++;
38     }
39
40     while (j < right.length) {
41         result.add(right[j]);
42         j++;
43     }
44 }
```

Output:

```
Unsorted List: [38, 27, 43, 3, 9, 82, 10]
Sorted List: [3, 9, 10, 27, 38, 43, 82]
```

Task 3:

Implement a Stack from Scratch.

Solution:

```
1 - class Stack<T> {
2     List<T> _stack = [];
3     void push(T value) {
4         _stack.add(value);
5     }
6     T? pop() {
7         if (isEmpty()) {
8             print("Stack is empty!");
9             return null;
10        }
11        return _stack.removeLast();
12    }
13    T? peek() {
14        if (isEmpty()) {
15            print("Stack is empty!");
16            return null;
17        }
18        return _stack.last;
19    }
20    bool isEmpty() {
21        return _stack.isEmpty();
22    }
23    int size() {
24        return _stack.length;
25    }
26    void display() {
27        print("Stack: $_stack");
28    }
29 }
30 void main() {
31     Stack<int> stack = Stack<int>();
32     stack.push(10);
33     stack.push(20);
34     stack.push(30);
35     stack.display();
36     print("Top element: ${stack.peek()}");
37     print("Popped element: ${stack.pop()}");
38     stack.display();
39     print("Is stack empty? ${stack.isEmpty()}");
40     print("Stack size: ${stack.size()}");
41 }
```

Output:

```
Stack: [10, 20, 30]
Top element: 30
Popped element: 30
Stack: [10, 20]
Is stack empty? false
Stack size: 2
```