***Mobile Application Development Lab***

***CSL-341***

***Lab Journal***



**Name:** Mohammad Sarim

**Class:** BSCS-6B

**Enrolment Number:** 01-134222-084

**Course:** Mobile App Development

**Department of Computer Science**

**BAHRIA UNIVERSITY ISLAMABAD**

# Lab 2 – Dart Introduction

## Objectives:

Basic syntax of Dart programming language

## Tools Used:

VS Code

### Submission Date:

**GITHUB Link:** [**Sarim139/MAD-LAB: These are the lab tasks which I have been given to complete in a semester.**](https://github.com/Sarim139/MAD-LAB)

# TASK 1:

Find the largest number in a given list.

**CODE:**

void main() {

List<int> numbers = [10, 45, 78, 23, 89, 12, 67];

int largest = numbers[0]; // Assume first element is the largest

for (int num in numbers) {

if (num > largest) {

largest = num;

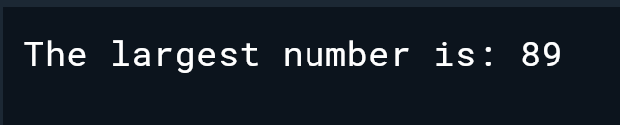
}

}

print("The largest number is: $largest");

}

**OUTPUT:**

****

**Solution:**

# TASK 2:

Use merge sort to sort a List.

**CODE:**

void main() {

List<int> numbers = [38, 27, 43, 3, 9, 82, 10];

print("Unsorted List: $numbers");

List<int> sortedList = mergeSort(numbers);

print("Sorted List: $sortedList");

}

List<int> mergeSort(List<int> list) {

if (list.length <= 1) {

return list; // Base case: a single element is already sorted

}

// Split the list into two halves

int mid = list.length ~/ 2;

List<int> left = list.sublist(0, mid);

List<int> right = list.sublist(mid);

// Recursively sort both halves

left = mergeSort(left);

right = mergeSort(right);

// Merge the sorted halves

return merge(left, right);

}

List<int> merge(List<int> left, List<int> right) {

List<int> result = [];

int i = 0, j = 0;

// Merge both lists in sorted order

while (i < left.length && j < right.length) {

if (left[i] < right[j]) {

result.add(left[i]);

i++;

} else {

result.add(right[j]);

j++;

}

}

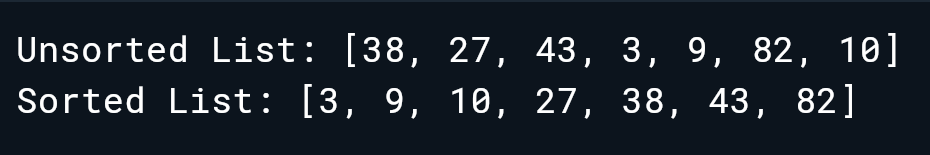
// Add remaining elements (if any)

result.addAll(left.sublist(i));

result.addAll(right.sublist(j));

return result;

}



# Task 3:

Implement a Stack from Scratch.

**CODE:**

class Stack<T> {

List<T> \_stack = [];

void push(T value) {

\_stack.add(value);

}

T? pop() {

if (isEmpty()) {

print("Stack is empty!");

return null;

}

return \_stack.removeLast();

}

T? peek() {

if (isEmpty()) return null;

return \_stack.last;

}

bool isEmpty() {

return \_stack.isEmpty;

}

int size() {

return \_stack.length;

}

void display() {

print("Stack: $\_stack");

}

}

void main() {

Stack<int> stack = Stack<int>();

stack.push(10);

stack.push(20);

stack.push(30);

stack.display();

print("Top Element: ${stack.peek()}");

print("Popped: ${stack.pop()}");

stack.display();

print("Stack Size: ${stack.size()}");

}

