Mobile Application Development Lab

CSL-341

Lab Journal

Student Name: Hafiz Muhammad Abdullah
Enrollment No: 01-134222-052
Class and Section: BSCS(5-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:

```
int largestnum(var a){
  int search = a[0];
  for (int i = 1 ; i < a.length; i++){
    if(a[i]>search){
      search = a[i];
    }
  }
  return search;
}
void main(){
  var a = [10,20,99,30,40];
  var largestnumber = largestnum(a);
  print(largestnumber);
}
```

Output:

```
99
```

TASK 2:

Use merge sort to sort a List.

Solution:

```
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;
  }

  int mid = list.length ~/ 2;
  List<int> left = mergeSort(list.sublist(0, mid));
  List<int> right = mergeSort(list.sublist(mid));

  return merge(left, right);
}

List<int> merge(List<int> left, List<int> right) {
  List<int> result = [];
```

```
  int i = 0, j = 0;

  while (i < left.length && j < right.length) {
    if (left[i] < right[j]) {
      result.add(left[i]);
      i++;
    } else {
      result.add(right[j]);
      j++;
    }
  }

  while (i < left.length) {
    result.add(left[i]);
    i++;
  }

  while (j < right.length) {
    result.add(right[j]);
    j++;
  }

```

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:

```dart
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()) {
      print("Stack is empty!");
      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 element: ${stack.pop()}");
  stack.display();
  print("Is stack empty? ${stack.isEmpty()}");
  print("Stack size: ${stack.size()}");
}
```

Output:

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