

# Scripting Language Lab Assignment

**Name :** Rahul Kumar

**Section :** A

**Reg No :** 201900095

**Objective:**

Create a calculator app using Angular which is capable of performing following operations:

1. Addition of two numbers
2. Subtraction of two numbers
3. Multiplication of two numbers
4. Division of two numbers
5. Factorial of a number
6. Checking if a given number is Prime or not

**index.html**

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">

<title>Calculator</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1"> <link rel="icon"
type="image/x-icon" href="favicon.ico">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZM
l3AhiU" crossorigin="anonymous">

</head>
<body>
<center><h1>Calculator app</h1></center> <app-root></app-root>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384/bQdsTh/da6pkl1MST/rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2
THRZ" crossorigin="anonymous"></script>
</body></html>
```

### **app-component.html**

```
<app-calculator></app-calculator>
```

### **app-component.ts**

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'calc-app';
}
```

### **app.module.ts**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

import { CalculatorComponent } from './calculator/calculator.component'; import {
  CalculatorKeysComponent } from './calculator-keys/calculator-keys.component';

@NgModule({
  declarations: [
    AppComponent,
    CalculatorComponent,
    CalculatorKeysComponent
  ],
  imports: [
    BrowserModule
```

```

],
providers: [],
bootstrap: [AppComponent]

})

export class AppModule { }
calculator.component.spec.ts

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CalculatorComponent } from './calculator.component';

describe('CalculatorComponent', () => {
  let component: CalculatorComponent;
  let fixture: ComponentFixture<CalculatorComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ CalculatorComponent ]
    })
    .compileComponents();
  });
  beforeEach(() => {
    fixture = TestBed.createComponent(CalculatorComponent); component =
    fixture.componentInstance; fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

calculator.component.ts

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-calculator',

  templateUrl: './calculator.component.html',

```

```
styleUrls: ['./calculator.component.css'] })
```

```
export class CalculatorComponent {
```

```
}
```

### **calculator-keys.component.html**

```
<input type="text" class="calculator-screen" [value]="currentNumber" disabled> <div  
class="calculator-keys">
```

```
<button type="button" (click) = "getfacto()" class="operator" >!</button>
```

```
<button type="button" (click) = "getPrime()" class="operator" >Prime</button>
```

```
>
```

```
<button type="button" (click) = "getOperation('+)" class="operator" value=" + ">+</button>
```

```
<button type="button" (click) = "getOperation('-)" class="operator" value=" - ">-</button>
```

```
<button type="button" (click) = "getOperation('*)" class="operator" value="
```

```
*">x</button>
```

```
<button type="button" (click) = "getOperation('/')" class="operator" value=" /">/</button>
```

```
<button type="button" (click) = "getNumber('7')" value="7">7</button> <button  
type="button" (click) = "getNumber('8')" value="8">8</button> <button type="button"  
(click) = "getNumber('9')" value="9">9</button>
```

```
<button type="button" (click) = "getNumber('4')" value="4">4</button> <button  
type="button" (click) = "getNumber('5')" value="5">5</button> <button type="button"  
(click) = "getNumber('6')" value="6">6</button>
```

```
<button type="button" (click) = "getNumber('1')" value="1">1</button> <button  
type="button" (click) = "getNumber('2')" value="2">2</button> <button type="button"  
(click) = "getNumber('3')" value="3">3</button>
```

```
<button type="button" (click) = "getNumber('0')" value="0">0</button>  
<button type="button" (click) = "getDecimal()" class="decimal" value=".">.</ button>  
<button type="button" (click) = "clear()" class="all-clear" value="all-clear">AC</button>  
<button type="button" (click) = "getOperation('=)" class="equal-sign"  
value="=">=</button>  
</div>
```

### **calculator-keys.component.specs.ts**

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { CalculatorKeysComponent } from './calculator-keys.component';

describe('CalculatorKeysComponent', () => {
  let component: CalculatorKeysComponent;
  let fixture: ComponentFixture<CalculatorKeysComponent>;
  beforeEach(async () => {
    await TestBed.configureTestingModule({ declarations: [ CalculatorKeysComponent ]
  })
  .compileComponents();
});

beforeEach(() => {

  fixture = TestBed.createComponent(CalculatorKeysComponent); component =
  fixture.componentInstance; fixture.detectChanges();
});
it('should create', () => {
  expect(component).toBeTruthy();
});
});
```

### **Calculator-keys.component.ts**

```
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-calculator-keys',
  templateUrl: './calculator-keys.component.html',
  styleUrls: ['./calculator-keys.component.css'] })
export class CalculatorKeysComponent{

  currentNumber = '0';
  firstOperand= 0;
  operator = "";
  waitForSecondNumber = false;

  public getNumber(v: string){
    console.log(v);
    if(this.waitForSecondNumber){
      this.currentNumber = v;
      this.waitForSecondNumber = false;
    }
    else{
```

```
this.currentNumber === '0'? this.currentNumber = v: this.currentNumber +=v;
}
}
```

```
getDecimal(){
if(!this.currentNumber.includes('.')){
this.currentNumber += '.';
}
```

```

}
}
getPrime(){
const num = Number(this.currentNumber);
let flag = 0;
if(num < 2){
    this.currentNumber = "Neither Prime nor Composite"
}
for (let k = 2; k < num; k++){
    if( num % k == 0){
        flag =1;
    }
}
}
```

```
if(flag==0){
    this.currentNumber = "Prime"
}
else{
    this.currentNumber = "Composite"
}
}
```

```
getfacto(){
    const num= Number(this.currentNumber);
    let answer = 1;
    if (num == 0 || num == 1){
        this.currentNumber= "1";
    }
    else{
        for(var i = num; i >= 1; i--){
            answer = answer * i;
        }
        this.currentNumber = String(answer);
    }
}
private doCalulation(op:string , secondOp:number){
    switch (op){
```

```
case '+':  
return this.firstOperand += secondOp;
```

```
case '-':  
return this.firstOperand -= secondOp;
```

```
case '*':  
return this.firstOperand *= secondOp;
```

```
case '/':  
return this.firstOperand /= secondOp;
```

```
case '=':  
return secondOp;
```

```
}  
return secondOp;
```

```
}
```

```
public getOperation(op: string){  
console.log(op);  
if(this.firstOperand === null){  
this.firstOperand = Number(this.currentNumber);  
}  
else if(this.operator){  
const result = this.doCalculation(this.operator , Number(this.currentNumber)) as number  
this.currentNumber = String(result);  
this.firstOperand = result;  
}  
}
```

```
this.operator = op;  
this.waitForSecondNumber = true;  
console.log(this.firstOperand);  
}
```

```
public clear(){  
this.currentNumber = '0';  
this.firstOperand = 0;  
this.operator = "";  
this.waitForSecondNumber = false;  
}  
}
```

## Screenshot

