

Arithmetic Calculator

Introduction:

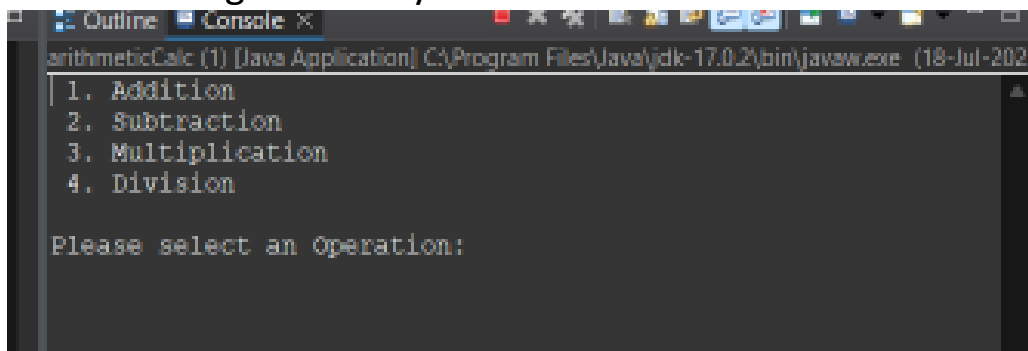
This is an Arithmetic calculator fully written in JAVA. User can do basic operations like Addition, Subtraction, Multiplication and Division.

Installation Guide:

1. GitHub link: <https://github.com/baljeet-singh97/JAVA-FSD>
2. Download the entire project as Zip in local system.
3. import the project in **Eclipse IDE**
4. Go to src/arithmeticCalculator/arithmeticCalc.java and run the code

How to use:

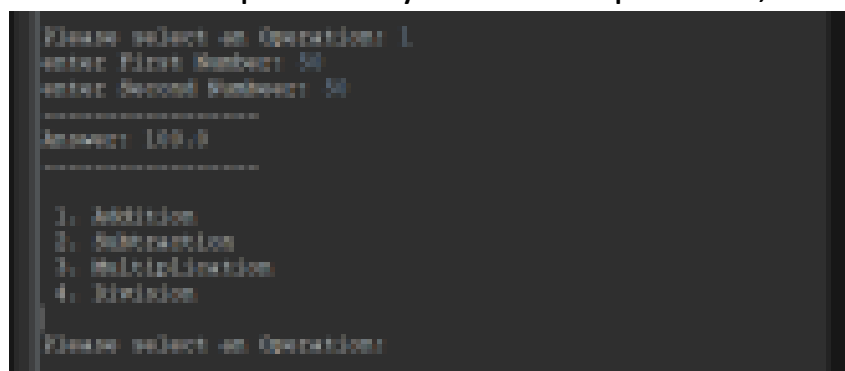
1. After running the code you will see the window like:



```
arithmeticCalc (1) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (18-Jul-2023)
1. Addition
2. Subtraction
3. Multiplication
4. Division

Please select an Operation:
```

2. Enter the number of the operation you want to perform, and press Enter.



```
Please select an Operation: 1
enter First Number: 50
enter Second Number: 50
Answer: 100.0

1. Addition
2. Subtraction
3. Multiplication
4. Division

Please select an Operation:
```

eg: Addition pressing 1

3. After pressing the desired operation number user will be asked to enter the values to perform desired operation and after that user will get output and program runs again.

Code Description:

Main Function:

- Creates an object of Scanner class

```
Scanner sc = new Scanner(System.in);
```

- taking input from the user

```
int operation = sc.nextInt();
```

- used switch case to navigate to desired Class

```
switch(operation)
```

- created object of class addition in case 1

case 1:

```
addition objA = new addition();
```

- **calling the class addition, same called all the classes for all operations**

```
objA.add();
```

```
break;
```

- Called Main function again at the end of every operation so user can start new operation by default

```
System.out.print("\n");
```

```
arithmeticCalc.main(null);
```

Classes:

1. Class userInput

Defined this class to take input from the user and store the values and called the same class every time for every operation.

- Used public Access Specifiers so that we can access the values anywhere
- 'Double' to get float values and return float values and bigger values also
- Two methods used inB() & inA()
- Returned values of 'a' and 'b' to the calling method respectively

```
public double inB()
```

```
{
```

```
    System.out.print("enter Second Number: ");
```

```
    double b = sc.nextDouble();
```

```
    return b;
```

```
} .
```

2. Class addition

Method used add()

- Created object for the class userInputs and called both 'a' and 'b' and adding both the numbers.

```
userInputs objUi = new userInputs(); .
```

```
double a = objUi.inA() .
```

```
double b = objUi.inB(); .
```

```
double ans = a+b; .
```

3. Class Subtraction

Method used subtract()

- Created object for the class userInputs and called both 'a' and 'b' and subtracting both the numbers.

```
userInputs objUi = new userInputs(); .
```

```
double a = objUi.inA() .
```

```
double b = objUi.inB(); .
```

```
double ans = a-b; .
```

4. Class multiplication

Method used multiply()

- Created object for the class userInputs and called both 'a' and 'b' and multiplying both the numbers.
- Used format to fix the value after decimal upto 3 values after decimal

```
System.out.format("%.3f\n",ans) .
```

```
userInputs objUi = new userInputs(); .
```

```
double a = objUi.inA() .
```

```
double b = objUi.inB(); .
```

```
double ans = a*b; .
```

5. Class division

Method used divide()

- Created object for the class userInputs and called both 'a' and 'b' and mdivide both the numbers.
- Used explicit type casting, if there is no reminder then converted double to int.
- Used format to fix the value after decimal upto 3 values after decimal

```
System.out.format("%.3f\n",ans)
```

```
userInputs objUi = new userInputs();
```

```
double a = objUi.inA()
```

```
double b = objUi.inB();
```

```
double ans = a/b;
```

```
if(a%b == 0) int value = (int)ans;
```