Simple Calculator

Create Simple Calculator

This calculator program in C helps the user to enter the Operator (+, -, *, or /) and two values. Using those two values and operand, it will perform Arithmetic Operations. For this C calculator program example, we used the Switch case to check which operand is inserted by the user. Next, based on the Operand result will display. It is the same C program that we used in the first example. But this time, we are performing the arithmetic operation directly inside the Switch case blocks.

Different ways to create a Calculator Program in C

Following are the different ways to write a Calculator Program in the C language.

1. Calculator Program in C using the switch statement.
2. Calculator Program in C using if else if statement.
3. Calculator Program in C using do-while loop and switch statement.
4. Calculator Program in C using function and switch statement.

Swot Analysis

STRENGTHS

The strength to calculate the any numbers with different arithmetic functions.

WEAKNESSES

Do not include any other signs rather than this arithmetic functions (+,-,*,/,%).

OPPORTUNITIES

Using the physical operators this project is huge valuable in market.

THREATS

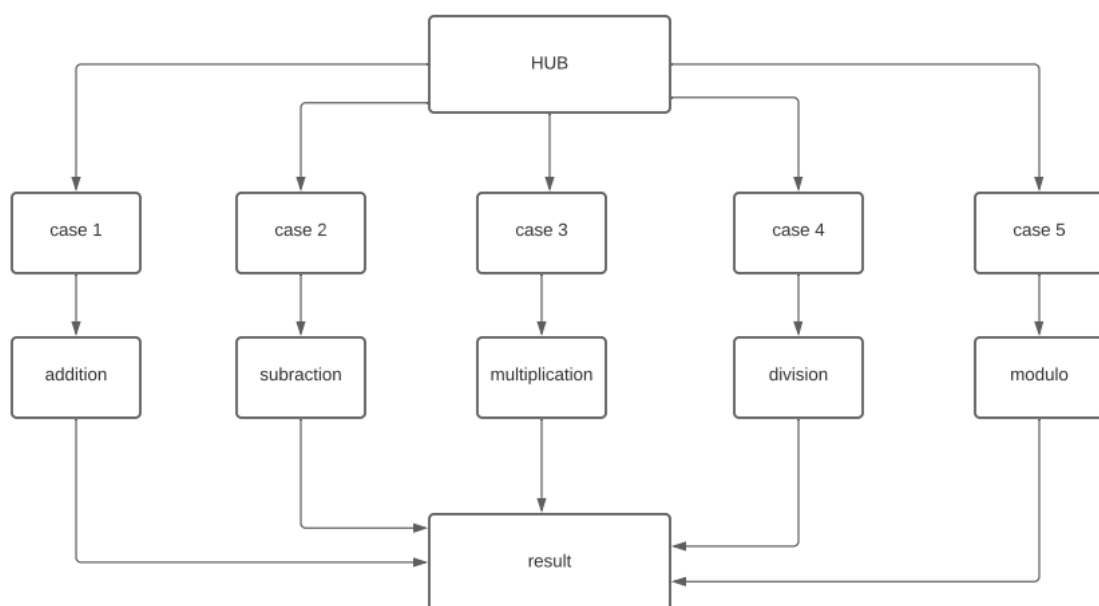The slower growth in technological innovation will also bring a significant threat in the upcoming dynamic world.

HIGH LEVEL REQUIREMENTS:-

1 User should able to view operation list.

2 User should able to select the operation.

3 The system should able perform the given operation.

4. The system should provide correct result.

LOW LEVEL REQUIREMENTS:-

User should identify correctly.

Structure

Flow chart

```
START
   │
   ▼
declare
n1,n2,operator,result
   │
   ▼
Data
   │
   ▼
Data
   │
   ▼
switch operator
```

```
case 1:  ──Yes──▶  result=n1+n2  ──────▶
   │ No
   ▼
case 2:  ──Yes──▶  result=n1-n2  ──────▶
   │ No
   ▼
case 3:  ──yes──▶  result=n1*n2  ──────▶
   │ No
   ▼
case 4:  ──yes──▶  result=n1/n2  ──────▶
   │ No
   ▼
case 5:  ──yes──▶  result=n1%n2  ──────▶
   │ No
   ▼
Default:  ──Yes──▶  write : No Operator or Invalid Operator selected!
```

```
write: result
```

```
write : No Operator
or Invalid Operator
selected!
   │
   ▼
stop
```