

Task 2: Build a Simple Calculator

DESCRIPTION:

- This is the description of Task-2 of my python internship at Happieloop.
- Here, the task is to develop a **Simple Calculator** that performs basic arithmetic calculations.
- I have created a file **task2.py** and developed python code according to the requirements.
- The purpose of program that I have developed is to performs the following operations:
 - i) Addition
 - ii) Subtraction
 - iii) Multiplication
 - iv) Division
- I have defined a class **Calculator** and 4 functions **addition, subtraction, multiplication, division** in the class and developed codes in the functions to implement these 4 operations.
- This code also checks and for the following errors and handles them. They are:
 - i) TypeError
 - ii) ValueError
 - iii) Division by Zero Error
- For example, If a string is given as input in place of integer, it will be an invalid input. In that case an error

will be raised up. The other, In the same if denominator is 0 in the case of division operation, it would also be an error.

- My code will raise up such errors and performs calculation operation if no such error occurs during the intake of inputs and gives output.
- My code can be used to calculate these 4 different operations between 2 integers.
- We should also give the operator required for the operation we need to perform.
- Based on the input of the operator the operation will be performed.
- If another symbol is given rather than +, -, *, / then it will show some error.
- Let's see some examples, how my code can be implemented by taking different inputs to the code.

CODE:

```
class Calculator:
    def __init__(self, a, b):
        self.a = a
        self.b = b

    def addition(self):
        try:
            return self.a + self.b
        except TypeError as e:
            print(f'Error: {e}')
            return None

    def subtraction(self):
```

```
try:
    return self.a - self.b
except TypeError as e:
    print(f"Error: {e}")
    return None
```

```
def multiplication(self):
    try:
        return self.a * self.b
    except TypeError as e:
        print(f"Error: {e}")
        return None
```

```
def division(self):
    try:
        if self.b == 0:
            raise ZeroDivisionError("Division by 0 is not
possible")
        return self.a / self.b
    except (TypeError, ZeroDivisionError) as e:
        print(f"Error: {e}")
        return None
```

```
def get_input(prompt):
    while True:
        try:
            return int(input(prompt))
        except ValueError:
            print("Invalid input. Please enter a valid integer.")
```

```
a = get_input("Enter the first number: ")
```

```
b = get_input("Enter the second number: ")
operator = input("Select your required operator (+, -, *,
/): ")
```

```
c = Calculator(a, b)
```

```
if operator == '+':
    print(c.addition())
elif operator == '-':
    print(c.subtraction())
elif operator == '*':
    print(c.multiplication())
elif operator == '/':
    print(c.division())
else:
    print("Invalid operator selected.")
```

- Now let me show some of the sample inputs and corresponding outputs in different cases.

OUTPUTS:

Enter the first number: 50

Enter the second number: 10

Select your required operator (+, -, *, /): *

500

Enter the first number: 25

Enter the second number: 5

Select your required operator (+, -, *, /): +

30

Enter the first number: 55

Enter the second number: 0

Select your required operator (+, -, *, /): /

Error: Division by 0 is not possible

None

Enter the first number: 12

Enter the second number: 3

Select your required operator (+, -, *, /): /

4.0

Enter the first number: 12

Enter the second number: b

Invalid input. Please enter a valid integer.

Enter the second number: 6

Select your required operator (+, -, *, /): *

72

Enter the first number: 34

Enter the second number: bbv

Invalid input. Please enter a valid integer.

Enter the second number: 44

Select your required operator (+, -, *, /): *

1496

Enter the first number: 75627893412

Enter the second number: 87343296

Select your required operator (+, -, *, /): *

6605589480140765952