```python
class Calculator:
    def add(self, a, b):
        return a + b
    def subtract(self, a, b):
        return a - b
    def multiply(self, a, b):
        return a * b
    def divide(self, a, b):
        if b == 0:
            raise ZeroDivisionError("Cannot divide by zero.")
        return a / b
def get_number(prompt):
    while True:
        try:
            return float(input(prompt))
        except ValueError:
            print("Invalid input. Please enter a numeric value.")
def get_operation():
    valid_operations = {'+': 'add', '-': 'subtract', '*': 'multiply', '/':
'divide'}
    while True:
        operation = input("Enter an operation (+, -, *, /): ")
        if operation in valid_operations:
            return valid_operations[operation]
        else:
            print("Invalid operation. Please choose one of +, -, *, or /.")
def main():
    calculator = Calculator()
    while True:
        try:
            num1 = get_number("Enter the first number: ")
            num2 = get_number("Enter the second number: ")
            operation = get_operation()
            method = getattr(calculator, operation, None)
            if method is None:
                raise KeyError("Invalid operation key.")
            result = method(num1, num2)
            print(f"The result of {num1} {operation} {num2} is: {result}")
            if input("Do you want to perform another calculation? (y/n):
").lower() != 'y':
                print("Goodbye!")
                break
        except ZeroDivisionError as e:
            print(f"Error: {e}. Please try again.")
        except ValueError as e:
            print(f"Error: {e}. Please provide valid numbers.")
        except KeyError as e:
            print(f"Error: {e}. Please provide a valid operation.")
```

```
        except TypeError as e:
            print(f"Unexpected error: {e}. Please try again.")
        except Exception as e:
            print(f"An unexpected error occurred: {e}. Please try again.")
main()
```