

```

class Calculator:
    def add(self, num1, num2):
        return num1 + num2

    def subtract(self, num1, num2):
        return num1 - num2

    def multiply(self, num1, num2):
        return num1 * num2

    def divide(self, num1, num2):
        if num2 == 0:
            return "Cannot divide by zero"
        return num1 / num2

    def is_odd_even(self, num):
        return "Even" if num % 2 == 0 else "Odd"

    def is_positive_negative(self, num):
        return "Positive" if num >= 0 else "Negative"

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

def main():
    calculator = Calculator()
    while True:
        print("\nCalculator Features:")
        print("1. Add Two Numbers")
        print("2. Subtract Two Numbers")
        print("3. Multiply Two Numbers")
        print("4. Divide Two Numbers")
        print("5. Check if a Number is Odd or Even")
        print("6. Check if an Integer is Positive or Negative")
        print("7. Exit the Application")

        choice = input("Enter your choice (1-7): ")

        if choice == '1':
            num1 = get_integer_input("Enter the first number: ")
            num2 = get_integer_input("Enter the second number: ")
            result = calculator.add(num1, num2)
            print(f"Result: {result}")
        elif choice == '2':
            num1 = get_integer_input("Enter the first number: ")
            num2 = get_integer_input("Enter the second number: ")
            result = calculator.subtract(num1, num2)
            print(f"Result: {result}")
        elif choice == '3':
            num1 = get_integer_input("Enter the first number: ")
            num2 = get_integer_input("Enter the second number: ")
            result = calculator.multiply(num1, num2)
            print(f"Result: {result}")
        elif choice == '4':
            num1 = get_integer_input("Enter the first number: ")
            num2 = get_integer_input("Enter the second number: ")
            result = calculator.divide(num1, num2)
            print(f"Result: {result}")
        elif choice == '5':
            num = get_integer_input("Enter a number: ")
            result = calculator.is_odd_even(num)
            print(f"The number is {result}")
        elif choice == '6':
            num = get_integer_input("Enter a number: ")
            result = calculator.is_positive_negative(num)
            print(f"The number is {result}")
        elif choice == '7':
            print("Exiting the application. Goodbye!")

```

```
        break
    else:
        print("Invalid choice. Please enter a valid option (1-7).")

if __name__ == "__main__":
    main()
```

1. Import the Required Classes and Functions:

```
class Calculator: # Class definition for the Calculator
```

Here, we define a class called `Calculator`. In object-oriented programming, a class is a blueprint for creating objects. In this case, `Calculator` is a blueprint for creating calculator objects.

2. Define Calculator Methods:

Within the `Calculator` class, we define various methods to perform calculations and checks. These methods include:

```
add(self, num1, num2): Adds two numbers.
```

```
subtract(self, num1, num2): Subtracts two numbers.
```

```
multiply(self, num1, num2): Multiplies two numbers.
```

```
divide(self, num1, num2): Divides two numbers.
```

```
is_odd_even(self, num): Checks if a number is odd or even.
```

```
is_positive_negative(self, num): Checks if a number is positive or negative.
```

These methods encapsulate the behavior of a calculator, and they are defined within the `Calculator` class so that calculator objects can use them.

3. get_integer_input Function:

```
def get_integer_input(prompt): # Function for getting valid integer input from the user
```

This is a separate function that we introduced for getting valid integer input. It helps ensure that only valid integers are accepted as input.

4. Main Function:

```
def main(): # The main function where the program execution starts
```

The `main` function is the entry point of the program. It orchestrates the user interaction, menu choices, and calculation operations.

5. Creating a Calculator Object:

```
calculator = Calculator()
```

Here, we create an instance (object) of the `Calculator` class. This `calculator` object is used to access and utilize the methods defined in the `Calculator` class. It represents a virtual calculator that the user interacts with.

6. Menu and User Interaction:

Within the `main` function, there is a while loop that repeatedly displays a menu to the user and takes their input. It provides options for performing various calculations or exiting the application.

When the user selects an option (e.g., addition), the corresponding method of the `calculator` object is called to perform the calculation.

For numeric inputs, we use the `get_integer_input` function to ensure that only valid integers are accepted, and it handles input validation.

7. Object-Oriented Principles:

The code follows object-oriented principles by encapsulating related functionality (calculator operations) within a class (`Calculator`) and creating instances of that class (`calculator` object) to perform specific tasks.

In summary, the code leverages classes and objects to create a modular and organized structure for a basic calculator application. The `Calculator` class defines methods for calculations, and the `calculator` object is used to invoke these methods based on user input. Additionally, the `get_integer_input` function provides input validation, ensuring that only valid integers are accepted for calculations.