```
// C++ program to create calculator using
// switch statement
#include <iostream>
using namespace std;

// Driver code
int main()
{
    char op;
    float num1, num2;

    // It allows user to enter operator
    // i.e. +, -, *, /
    cin >> op;

    // It allow user to enter the operands
    cin >> num1 >> num2;

    // Switch statement begins
    switch (op) {
    // If user enter +
    case '+':
        cout << num1 + num2;
        break;

    // If user enter -
    case '-':
        cout << num1 - num2;
        break;

    // If user enter *
    case '*':
        cout << num1 * num2;
        break;
    // If user enter /
    case '/':
        cout << num1 / num2;
        break;

    // If the operator is other than +, -, * or /,
    // error message will display
    default:
        cout << "Error! operator is not correct";
    }
    // switch statement ends

    return 0;
}
```

Certainly! Let’s go through the provided C++ program step-by-step to understand how it functions as a simple calculator using a `switch` statement:

Program Overview

This C++ program allows the user to perform basic arithmetic operations (addition, subtraction, multiplication, and division) based on the operator they provide. The `switch` statement is used to handle different operations based on the user’s input.

Detailed Explanation

1. **Header and Namespace**:

```
   #include <iostream>
   using namespace std;
```

   - Includes the `iostream` library for input and output operations.
   - Uses the `std` namespace to avoid prefixing `std::` before standard library functions and objects.

2. **Main Function**:

~~~~
   int main()
   {
~~~~

   - The starting point of the program.

3. **Variable Declarations**:

~~~~
      char op;
   float num1, num2;
   ```
~~~~

   - `op`: Variable to store the operator (e.g., `+`, `-`, `*`, `/`).
   - `num1` and `num2`: Variables to store the two numbers on which the operation will be performed.

4. **Input for Operator**:

~~~~
     cin >> op;
   ```
~~~~

   - Prompts the user to enter an arithmetic operator. This value is read into the variable `op`.

5. **Input for Operands**:

~~~~
     cin >> num1 >> num2;
   ```
~~~~

   - Prompts the user to enter two numbers. These values are read into `num1` and `num2`.

6. **Switch Statement**:

~~~~
     switch (op) {
   ```
~~~~

   - Evaluates the value of `op` and executes the corresponding `case` block based on the operator entered.

7. **Case Blocks**:

~~~~
      case '+':
       cout << num1 + num2;
       break;
~~~~

~~~
```
~~~

- If the operator is `+`, it performs addition (`num1 + num2`) and prints the result.
- `break;` ends the current case block and exits the `switch` statement.

```
   case '-':
     cout << num1 - num2;
     break;
```

- If the operator is `-`, it performs subtraction (`num1 - num2`) and prints the result.

```
   case '*':
     cout << num1 * num2;
     break;
```

- If the operator is `*`, it performs multiplication (`num1 * num2`) and prints the result.

```
 case '/':
     cout << num1 / num2;
     break;
```

- If the operator is `/`, it performs division (`num1 / num2`) and prints the result.
- **Note**: This code doesn’t handle division by zero, which can cause a runtime error if `num2` is `0`.

8. **Default Case**:

```
 default:
     cout << "Error! operator is not correct";
```

- If the user enters an operator that is not recognized (not `+`, `-`, `*`, or `/`), the `default` case executes and displays an error message.

9. **End of Main Function**:

   return 0;
   }
   ```
   - Ends the `main` function and returns `0`, indicating that the program executed successfully.

### Summary

The program functions as follows:
1. The user inputs an arithmetic operator and two numbers.
2. Based on the operator, the `switch` statement performs the corresponding arithmetic operation.
3. It handles basic operations (addition, subtraction, multiplication, division) and displays the result.
4. If the operator is invalid, it displays an error message.

This simple calculator provides a straightforward example of using a `switch` statement to handle multiple conditions based on user input.
   ```