

# Create a Simple Calculator with a GUI

Company : Saiket Systems Innovate. Elevate. Excel.

Name : Ashwini Kallappa Naik

Intern ID : SKS/A2/C48374

Domain : C/C++ Development Intern

Duration : 1 Month (30-08-2025 to 30-09-2025)

## Explanation

1. **GTK (GIMP Toolkit)** is a popular GUI library for C.
2. We'll use:
  - **GtkEntry** → to show the expression and result.
  - **GtkButton** → for digits (0–9), operators (+, -, \*, /), clear (C), and equals (=).
  - **GtkGrid** → to arrange buttons in a grid layout like a calculator.
3. **Logic:**
  - Button clicks append digits/operators to the entry.
  - = evaluates the expression (we'll use `g_strtod` + basic parsing).
  - C clears the display.

## Full C Code (GTK Calculator)

```
#include <gtk/gtk.h>
#include <string.h>
#include <stdlib.h>

GtkWidget *entry; // To display input/output

// Append text to the entry
void on_button_clicked(GtkWidget *widget, gpointer data) {
    const char *text = (const char *)data;
    const char *current = gtk_entry_get_text(GTK_ENTRY(entry));
    char *new_text = g_strdup_printf("%s%s", current, text);
    gtk_entry_set_text(GTK_ENTRY(entry), new_text);
    g_free(new_text);
}

// Clear the entry
void on_clear_clicked(GtkWidget *widget, gpointer data) {
    gtk_entry_set_text(GTK_ENTRY(entry), "");
}

// Evaluate expression (only supports + - * /)
void on_equal_clicked(GtkWidget *widget, gpointer data) {
    const char *expr = gtk_entry_get_text(GTK_ENTRY(entry));
    double num1, num2;
    char op;
    if (sscanf(expr, "%lf %c %lf", &num1, &op, &num2) == 3) {
        double result = 0;
        switch (op) {
            case '+': result = num1 + num2; break;
            case '-': result = num1 - num2; break;
            case '*': result = num1 * num2; break;
            case '/': result = (num2 != 0) ? num1 / num2 : 0; break;
        }
    }
}
```

```
char res_str[50];
snprintf(res_str, sizeof(res_str), "%g", result);
gtk_entry_set_text(GTK_ENTRY(entry), res_str);
} else {
gtk_entry_set_text(GTK_ENTRY(entry), "Error");
}
}
```

```
int main(int argc, char *argv[]) {
GtkWidget *window, *grid;
gtk_init(&argc, &argv);
```

```
// Create window
window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
gtk_window_set_title(GTK_WINDOW(window), "Simple
Calculator");
gtk_window_set_default_size(GTK_WINDOW(window), 250,
300);
g_signal_connect(window, "destroy",
G_CALLBACK(gtk_main_quit),
NULL);
```

```
// Create grid
grid = gtk_grid_new();
gtk_container_add(GTK_CONTAINER(window), grid);
```

```
// Entry widget
entry = gtk_entry_new();
gtk_editable_set_editable(GTK_EDITABLE(entry), FALSE);
gtk_grid_attach(GTK_GRID(grid), entry, 0, 0, 4, 1);
```

```
// Button labels
const char *buttons[4][4] = {
{"7", "8", "9", "/"},
{"4", "5", "6", "*"},
```

```

{"1", "2", "3", "-"},
{"0", "C", "=", "+"}
};

// Create buttons
for (int i = 0; i < 4; i++) {
for (int j = 0; j < 4; j++) {
GtkWidget *btn = gtk_button_new_with_label(buttons[i][j]);
gtk_grid_attach(GTK_GRID(grid), btn, j, i + 1, 1, 1);

if (strcmp(buttons[i][j], "C") == 0) {
g_signal_connect(btn, "clicked",
G_CALLBACK(on_clear_clicked), NULL);
} else if (strcmp(buttons[i][j], "=") == 0) {
g_signal_connect(btn, "clicked",
G_CALLBACK(on_equal_clicked), NULL);
} else {
g_signal_connect(btn, "clicked",
G_CALLBACK(on_button_clicked), (gpointer)buttons[i][j]);
}
}
}

gtk_widget_show_all(window);
gtk_main();
return 0;
}

```

## How to Compile & Run

1. Install GTK (Linux example for GTK 3):

```
sudo apt-get install libgtk-3-dev
```

2. Save code as calculator.c.

3. Compile with:

```
gcc calculator.c -o calculator `pkg-config --cflags --libs gtk+-3.0`
```

4.Run:

```
./calculator
```

## Output (GUI Layout)

When run, you'll see a **Calculator window** like this:

```
-----  
|      [Entry]      |  
-----  
  
| 7 | 8 | 9 | / |  
| 4 | 5 | 6 | * |  
| 1 | 2 | 3 | - |  
| 0 | C | = | + |  
-----
```

✓ Example:

- Press 7, +, 3, then = → Output is **10**
- Press C → Clears the entry