**Application Programs using Windows API**

I am Alvin Alexander, eighth standard student of Salvation Army HSS, Kowdiar, Thiruvananthapuram.

In the previous session, we learned about the message loop, the most crucial part of our window. This session, we will delve into components that make our GUI more functional like buttons.

In our main file:

… (Rest of the code)

HWND hWndButton = CreateWindowEx(0, "BUTTON", "Button", WS\_VISIBLE | WS\_CHILD | BS\_PUSHBUTTON, 235, 234, 70, 40, hWnd, NULL, (HINSTANCE)GetWindowLongPtr(hWnd, GWLP\_HINSTANCE), NULL);

In our message loop:

… (Rest of the code)

case WM\_COMMAND:

if (HIWORD(wParam) == BN\_CLICKED) {

MessageBox(NULL, "Message From Button Click!", "Message", MB\_ICONINFORMATION | MB\_OK);

}

return 0;

The code in our main code creates the button window inside of our main window (“(HINSTANCE)GetWindowLongPtr(hWnd, GWLP\_HINSTANCE)”). This function retrieves the HINSTANCE of our window and provides it to the function. “BUTTON” is the class name for any button created. The styles WS\_VISIBLE, WS\_CHILD, and BS\_PUSHBUTTON make the window: visible, a child to the main window and a push button, which can be clicked, respectively. The position and size of the button can be selected as desired (I have used.NET framework to create a window, position buttons to my desire and get the positions relative to the window position very easily and input it to my application). In our message loop, we handle WM\_COMMAND message sent by the button when it is clicked, in this example. The wParam parameter gets the notification code (Here, BN\_CLICKED) in its High Word Position (I highly recommend understanding data types and its bit representations in C, or else “High Word Position” and other terms may be confusing). If we get this notification code, we can run any code. In this example, I made the button click create a MessageBox with the message of my desire.

In the next session, we start an endeavor of creating an arbitrary bank application.