Class 2:

Class Code: Blink LED Using Push Button

STM32F446RE Board

Code:-

```
/**
 ******************************************************************************
 * @file           : main.c
 * @author         : Sagar More
 * @brief          : Class 2 - Blink LED on STM32F446RE by pressing Push Button
 *
 * On-board Push Button:
 *      - Port: GPIOC
 *      - Pin: 13
 *      - Bus: AHB1
 *
 * On-board LED:
 *      - Port: GPIOA
 *      - Pin: 5
 *      - Bus: AHB1
 ******************************************************************************
 */

/*
 Register Memory Mapping
 -----------------------------------------
 RCC Base Address: 0x40023800

 RCC_AHB1ENR Offset:     0x30
 RCC_AHB1ENR:      0x40023830

 GPIOA Base Address (LED): 0x40020000

 GPIOA_MODER Offset:             0x00
 GPIOA_MODER:              0x40020000

 GPIOA_ODR   Offset:             0x14
 GPIOA_ODR:                0x40020014

 GPIOC Base Address (Switch): 0x40020800

 GPIOC_MODER Offset:                0x00
 GPIOC_MODER:                0x40020800

 GPIOC_IDR   Offset:                0x10
 GPIOC_IDR:                  0x40020810
*/

```

```c
#include <stdint.h>
#include <stdbool.h>

/* ------------------------------------------------------------------
 * Memory-Mapped Registers
 * ------------------------------------------------------------------*/
#define RCC_AHB1ENR   (*(volatile uint32_t *)0x40023830) // Clock enable register

#define GPIOA_MODER   (*(volatile uint32_t *)0x40020000) // GPIOA mode register
#define GPIOA_ODR     (*(volatile uint32_t *)0x40020014) // GPIOA output data register

#define GPIOC_MODER   (*(volatile uint32_t *)0x40020800) // GPIOC mode register
#define GPIOC_IDR     (*(volatile uint32_t *)0x40020810) // GPIOC input data register

/* ------------------------------------------------------------------
 * Function Declarations
 * ------------------------------------------------------------------*/
void ledHigh(void);
void ledLow(void);
```

```c
/* ------------------------------------------------------------------
 * Main Function
 * ------------------------------------------------------------------*/
int main(void)
{
    /* --------------------------------
     *  Enable Clock for GPIOA & GPIOC
     * --------------------------------
     * RCC_AHB1ENR:
     * Bit 0 -> GPIOAEN
     * Bit 2 -> GPIOCEN
     * 0x5 = (1 << 0) | (1 << 2)
     */
    RCC_AHB1ENR |= (0x5U << 0);

    /* --------------------------------
     * Configure GPIOA PIN5 as Output
     * --------------------------------
     * MODER[11:10] = 01 (Output Mode)
     */
    GPIOA_MODER |=  (1U << 10);  // Set MODER10 = 1
    GPIOA_MODER &= ~(1U << 11);  // Set MODER11 = 0

```

```
    /* ------------------------------
     * Configure GPIOC PIN13 as Input
     * ------------------------------
     * MODER[27:26] = 00 (Input Mode)
     */
    GPIOC_MODER &= ~(1U << 26);  // Clear MODER26
    GPIOC_MODER &= ~(1U << 27);  // Clear MODER27

    // Flag to track LED state
    volatile bool ledFlag = false;

    /* ------------------------------
     * Super Loop
     * ------------------------------*/
    while (1)
    {
        /* ------------------------------
         * Read Button State (PC13)
         * ------------------------------
         * Active-low button:
         *   - 0 = Pressed
         *   - 1 = Released
         */

        if ((GPIOC_IDR & (1U << 13)) == 0)
        {
            ledFlag = true;   // Button pressed
        }
        else
        {
            ledFlag = false;  // Button released
        }

        /* ------------------------------
         * Control LED based on Flag
         * ------------------------------*/
        if (ledFlag)
        {
            ledHigh();  // Turn LED on
        }
        else
        {
            ledLow();   // Turn LED off
        }
    }
}

/* ---------------------------------------------------------------------
 * LED Control Functions
 * ---------------------------------------------------------------------*/

/* Turn LED ON (PA5 = 1) */
void ledHigh(void)
{
    GPIOA_ODR |= (1U << 5);
}

/* Turn LED OFF (PA5 = 0) */
void ledLow(void)
{
    GPIOA_ODR &= ~(1U << 5);
}
```