

## Übung HelloWorld

```
/*=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----
Name          : HelloWorld.c
Author        : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version       : 2012-09-22v1.0
Description   : Hello World Sample!
=====*/

#include <stdio.h>
#include <stdlib.h>

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    printf("Hello World!\n");

    getchar();
    return(EXIT_SUCCESS);
}
```

## Übung 1.1 - for Schleife

```

/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----
Name       : Uebung_1-1.c
Author    : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version   : 2012-09-22v1.0
Description : Uebung_1-1
            Entwurf Übung mit Funktionen und for - Schleife
=====*/

#include <stdio.h>
#include <stdlib.h>

/**
 * Draw "Raute" based on the Parameters
 * Example: drawRaute(3, 5)
 * #####
 * #####
 * #####
 * @param zeile Anzahl Zeilen
 * @param spalte Anzahl Spalten
 */
void drawRaute(int zeile, int spalte);

/**
 * Print out all Numbers from 0..value, ex, printNumbers(4): 0 01, 012, 0123, 01234
 * @param value Wert > 0 bis wohin gezählt werden soll
 */
void printNumbers(int value);

/**
 * Draw a triangle with '*'s
 * @param size Grösser des Dreiecks
 */
void drawTriangle(int size);

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    printf("drawRaute(3, 5)...\n");
    drawRaute(3, 5);

    printf("drawRaute(2, 2)...\n");
    drawRaute(2, 2);

    printf("printNumbers(10)...\n");
    printNumbers(10);

    printf("drawTriangle(6)...\n");
    drawTriangle(6);

    printf("\n\nPress [Enter] to exit...");
    getchar();
    return(EXIT_SUCCESS);
}

void drawRaute(int zeile, int spalte)
{
    int lineCount, rowCount;
    for (lineCount = 0; lineCount < zeile; lineCount++)
    {
        for (rowCount = 0; rowCount < spalte; rowCount++)
        {
            printf("#");
        }
        printf("\n");
    }
    printf("\n");
}

```

```
void printNumbers(int value)
{
    int count, currCount;
    for (count = 0; count <= value; count++)
    {
        for (currCount = 0; currCount < count; currCount++)
        {
            printf("%i", currCount);
        }
        printf("\n");
    }
    printf("\n");
}

void drawTriangle(int size)
{
    int i, j;
    for (i=size; i>=0; i--)
    {
        for (j=i; j>0; j--)
        {
            printf(" ");
        }
        for(j=i; j<size; j++)
        {
            printf("*");
        }
        for(j=i; j<=size; j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

## Übung 1.2 - while Schleife

```

/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----
Name       : Uebung_1-2.c
Author    : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version   : 2012-09-22v1.0
Description : Uebung_1-2
            Entwurf Übung mit Funktionen und while Schleife
=====*/

#include <stdio.h>
#include <stdlib.h>

/**
 * Increment of an Integer between start and end
 * @param start Start Value to increment
 * @param end Limit Value
 */
void increment(int start, int end);

/**
 * Decrement of an Integer between start and end
 * @param start Start Value to decrement
 * @param end Limit Value
 */
void decrement(int start, int end);

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    printf("increment(0, 5)...\n");
    increment(0, 5);

    printf("decrement(10, -3)...\n");
    decrement(10, -3);

    printf("\n\nPress [Enter] to exit...");
    getchar();
    return(EXIT_SUCCESS);
}

void increment(int start, int end)
{
    while(start <= end)
    {
        printf("%d,", start);
        start++;
    }
    printf("\b \n\n"); // "\b " = Backspace & "Space" => remove last ","
}

void decrement(int start, int end)
{
    while(start >= end)
    {
        printf("%d,", start);
        start--;
    }
    printf("\b \n\n"); // "\b " = Backspace & "Space" => remove last ","
}

```

## Übung 1.3 - Integer nach Binary (iToBinary)

```
/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----
Name       : Uebung_1-3.c
Author    : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version   : 2012-09-22v1.0
Description : Uebung_1-2
            Integer nach Binary (iToBinary)
=====*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/**
 * Integer to Binary String Converter
 * @param value Integer to convert
 */
void iToBinary(int value);

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    int wert;

    printf("Wert als Integer (auch negativ): ");
    scanf("%d", &wert);
    fflush(stdin); //flushes the input stream and gets rid of '\n'

    iToBinary(wert);

    printf("\n\nPress [Enter] to exit...");
    getchar();
    return(EXIT_SUCCESS);
}

void iToBinary(int value)
{
    int cmp = 1;
    int cnt;
    char res[sizeof(int) * 8 + 1]; //get size of Integer * 8 (bits) + 1
    memset(res, 0, sizeof(res)); // Sets buffers to a specified character

    cnt = sizeof(int) * 8;
    while (cnt != 0)
    {
        cnt--;
        res[cnt] = (value & cmp) ? '1' : '0';
        cmp <<= 1;
    }
    printf("Integer: %d\n", value);
    printf("Binary: %s\n", res);
}
}
```

## Übung 1.4 - Text Menu

```

/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----*/

Name       : Uebung_1-4.c
Author     : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version    : 2012-09-22v1.0
Description: Uebung_1-4
            Schreiben Sie das Gerüst zu einer textbasierten Menusteuerung.
=====*/

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

/**
 * Show Main Menu
 */
void showMenuMain();

/**
 * Show SubMenu A, A-A, A-B, A-C
 */
void showMenuSubA();
void showMenuSubA_A();
void showMenuSubA_B();
void showMenuSubA_C();

/**
 * Show SubMenu B
 */
void showMenuSubB();

/**
 * Show SubMenu C
 */
void showMenuSubC();

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    // Show MainMenu
    showMenuMain();

    //printf("\n\nPress [Enter] to exit...");
    //getchar();
    return(EXIT_SUCCESS);
}

void showMenuMain()
{
    char input;

    do {
        // Show MainMenu
        system("cls");
        printf("MainMenu\n-----\n");
        printf("A --> Select menu item A\n");
        printf("B --> Select menu item B\n");
        printf("C --> Select menu item C\n");
        printf("Q --> Exit\n");

        // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'

        // Select SubMenu (or Exit)
        switch(input) {
            case 'A':
                system("cls");

```

```

        showMenuSubA();
        break;
    case 'B':
        system("cls");
        showMenuSubB();
        break;
    case 'C':
        system("cls");
        showMenuSubC();
        break;
    case 'Q':
        break;
    }
} while (input != 'Q');
}

void showMenuSubA()
{
    char input;

    do {
        // Show SubMenu
        system("cls");
        printf("SubMenu A\n-----\n");
        printf("A --> Select submenu item A-A\n");
        printf("B --> Select submenu item A-B\n");
        printf("C --> Select submenu item A-C\n");
        printf("Q --> Back to MainMenu\n");

        // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'

        // Select SubMenu (or Exit)
        switch(input) {
            case 'A':
                showMenuSubA_A();
                break;
            case 'B':
                showMenuSubA_B();
                break;
            case 'C':
                showMenuSubA_C();
                break;
            case 'Q':
                break;
        }
    } while (input != 'Q');
}

void showMenuSubA_A()
{
    char input;

    system("cls");
    printf("SubMenu A-A\n-----\n");
    printf("Q --> Back to SubMenu A\n");
    do { // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'
    } while (input != 'Q');
}

void showMenuSubA_B()
{
    char input;

    system("cls");
    printf("SubMenu A-B\n-----\n");
    printf("Q --> Back to SubMenu A\n");
    do { // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'
    } while (input != 'Q');
}

void showMenuSubA_C()

```

```
{
    char input;

    system("cls");
    printf("SubMenu A-C\n-----\n");
    printf("Q --> Back to SubMenu A\n");
    do { // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'
    } while (input != 'Q');
}

void showMenuSubB()
{
    char input;

    system("cls");
    printf("SubMenu B\n-----\n");
    printf("Q --> Back to MainMenu\n");
    do { // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'
    } while (input != 'Q');
}

void showMenuSubC()
{
    char input;

    system("cls");
    printf("SubMenu C\n-----\n");
    printf("Q --> Back to MainMenu\n");
    do { // get input
        printf("Enter selection: ");
        while (!isalpha(input = toupper(getchar())));
        fflush(stdin); //flushes the input stream and gets rid of '\n'
    } while (input != 'Q');
}
```



## Übung 1.5 - Celsius in Fahrenheit

```
/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----*/

Name       : Uebung_1-5.c
Author    : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version   : 2012-09-22v1.0
Description : Umrechnung Fahrenheit -> Celsius mit Funktion
=====*/

#include <stdio.h>
#include <stdlib.h>

/**
 * Convert Fahrenheit -> Celsius
 * @param value Fahrenheit
 * @return Value in Celsius
 */
float fahrenheitToCelsius(float fahrenheit);

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    float f, c;
    int start = 0;
    int end = 210;
    int step = 15;

    printf("Fahrenh.\tCelsius\n");
    for(f = start; f <= end; f += step) {
        c = fahrenheitToCelsius(f);
        printf("%d\t\t%f\n", (int)f, c);
    }

    printf("\n\nPress [Enter] to exit...");
    getchar();
    return(EXIT_SUCCESS);
}

float fahrenheitToCelsius(float fahrenheit)
{
    return (5.0f / 9.0f) * (fahrenheit - 32.0f);
}
```

## Übung 1.6 - ggT mit while Schleufe

```
/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----*/

Name       : Uebung_1-6.c
Author    : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version   : 2012-09-23v1.0
Description : Grösster gemeinsamer Teiler (ggT) mit while Schleufe
=====*/

#include <stdio.h>
#include <stdlib.h>

/**
 * Berechnet den GGT von zwei Zahlen
 * @param a erste Zahl
 * @param b zweite Zahl
 */
int ggT(int a, int b);

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    int ersteZahl, zweiteZahl, ggtZahl;

    printf("Erste Zahl: ");
    scanf("%d", &ersteZahl);
    printf("Zweite Zahl: ");
    scanf("%d", &zweiteZahl);

    ggtZahl = ggT(ersteZahl, zweiteZahl);
    printf("GGT von %d und %d ist: %d\n", ersteZahl, zweiteZahl, ggtZahl);

    fflush(stdin); //flushes the input stream and gets rid of '\n'
    printf("\n\nPress [Enter] to exit...");
    getchar();
    return(EXIT_SUCCESS);
}

int ggT(int a, int b)
{
    while (a != b) {
        if (a > b) {
            a -= b;
        }
        else {
            b -= a;
        }
    }
    return a;
}
```

## Übung 1.7 - Wörter zählen

```
/**=====
Hochschule Luzern - Technik & Architektur           Mikrocontroller - HS2012
Copyright 2012 Felix Rohrer                         http://hslu.ximit.ch
-----
Name       : Uebung_1-7.c
Author    : Felix Rohrer <felix.rohrer@stud.hslu.ch>
Version   : 2012-09-23v1.0
Description : Wörter Zählen
            Trennzeichen: ' ' (space), '\t' (tab) und '\n' (enter)
=====*/

#include <stdio.h>
#include <stdlib.h>

#define IN 1
#define OUT 2

/**
 * The main procedure to run the application.
 * @param argc Number of command line arguments
 * @param argv The forwarded command line arguments
 * @return Application return (error) code
 */
int main(int argc, char** argv)
{
    int c; // current Char
    int counter = 0; // Word Counter
    int state = OUT; // In or Out

    while ((c = getchar()) != EOF)
    {
        if ((c == ' ') || (c == '\t') || (c == '\n'))
        {
            state = OUT;
        }
        else if (state == OUT)
        {
            state = IN;
            counter++;
        }
    }
    printf("Anzahl Woerter: %d\n", counter);

    fflush(stdin); //flushes the input stream and gets rid of '\n'
    printf("\n\nPress [Enter] to exit...");
    getchar();
    return(EXIT_SUCCESS);
}
```