

ANNEX C

Example of a program that prints colored characters on the console

MS-Windows/DOS version

```
// PROG - MIEIC
// JAS
// Example of a program that prints colored characters on the console (in text mode)

#include <iostream>
#include <ctime>
#include <cstdlib>
#include <windows.h>

using namespace std;

//=====
//COLOR CODES: (alternative: use symbolic const's)
#define BLACK 0
#define BLUE 1
#define GREEN 2
#define CYAN 3
#define RED 4
#define MAGENTA 5
#define BROWN 6
#define LIGHTGRAY 7
#define DARKGRAY 8
#define LIGHTBLUE 9
#define LIGHTGREEN 10
#define LIGHTCYAN 11
#define LIGHTRED 12
#define LIGHTMAGENTA 13
#define YELLOW 14
#define WHITE 15
//=====

void clrscr(void)
{
    COORD coordScreen = { 0, 0 }; // upper left corner
    DWORD cCharsWritten;
    DWORD dwConSize;
    HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_SCREEN_BUFFER_INFO csbi;

    GetConsoleScreenBufferInfo(hCon, &csbi);
    dwConSize = csbi.dwSize.X * csbi.dwSize.Y;

    // fill with spaces
    FillConsoleOutputCharacter(hCon, TEXT(' '), dwConSize, coordScreen, &cCharsWritten);
    GetConsoleScreenBufferInfo(hCon, &csbi);
    FillConsoleOutputAttribute(hCon, csbi.wAttributes, dwConSize, coordScreen, &cCharsWritten);

    // cursor to upper left corner
    SetConsoleCursorPosition(hCon, coordScreen);
}

//=====

// Position the cursor at column 'x', line 'y'
// The coordinates of upper left corner of the screen are (x,y)=(0,0)
void gotoxy(int x, int y)
{
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}

//=====

// Set text color
void setcolor(unsigned int color)
{
    HANDLE hcon = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hcon, color);
}

//=====

// Set text color & background
void setcolor(unsigned int color, unsigned int background_color)
{
    HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    if (background_color == BLACK)
        SetConsoleTextAttribute(hCon, color);
    else
        SetConsoleTextAttribute(hCon, color | BACKGROUND_BLUE | BACKGROUND_GREEN |
                                BACKGROUND_RED);
}

//=====

// Fill the screen with colored numbers
int main()
{
    clrscr();

    srand((unsigned int)time(NULL));

    for (int x = 0; x < 80; x++)
        for (int y = 0; y < 24; y++)
        {
            gotoxy(x, y);
            if (rand() % 2 == 0)
                setcolor(x % 15 + 1);
            else
                setcolor(y % 15 + 1, rand() % 2);
            cout << x % 10;
        }
}
```

Linux version (using ANSI escape sequences)

```
// PROG - MIEIC
// JAS
// Example of a program that prints colored characters on the console (in text mode)
// LINUX version, using ANSI escape sequences

#include <iostream>
#include <string>
#include <sstream>
#include <unistd.h>
using namespace std;
//=====
// TEXT COLOR CODES
#define NO_COLOR      "\033[0m"
#define BLACK         "\033[0;30m"
#define RED           "\033[0;31m"
#define GREEN         "\033[0;32m"
#define BROWN        "\033[0;33m"
#define BLUE          "\033[0;34m"
#define MAGENTA       "\033[0;35m"
#define CYAN          "\033[0;36m"
#define LIGHTGRAY     "\033[0;37m"
#define DARKGRAY      "\033[1;30m"
#define LIGHTRED      "\033[1;31m"
#define LIGHTGREEN     "\033[1;32m"
#define YELLOW        "\033[1;33m"
#define LIGHTBLUE     "\033[1;34m"
#define LIGHTMAGENTA  "\033[1;35m"
#define LIGHTCYAN     "\033[1;36m"
#define WHITE         "\033[1;37m"
// BACKGROUND COLOR CODES
#define BLACK_B       "\033[0;40m"
#define RED_B         "\033[0;41m"
#define GREEN_B       "\033[0;42m"
#define YELLOW_B      "\033[0;43m"
#define BLUE_B        "\033[0;44m"
#define MAGENTA_B     "\033[0;45m"
#define CYAN_B        "\033[0;46m"
#define WHITE_B       "\033[1;47m"
//=====
// Position the cursor at column 'x', line 'y'
// The coordinates of upper left corner of the screen are (x,y)=(0,0)
void gotoxy(int x, int y)
{
    ostringstream oss;
    oss << "\033[" << y << " << " << " << x << "H";
    cout << oss.str();
}
//=====
// Clear the screen
void clrscr(void)
{
    cout << "\033[2J";
    gotoxy(0, 0);
}
//=====
// Set text color
void setcolor(string color)
{
    cout << color;
}
//=====
// Set text color & background
void setcolor(string color, string background_color)
{
    cout << color << background_color;
}
//=====
// Testing program
int main()
{
    clrscr();
    cout << RED << "Text in RED" << NO_COLOR << endl;
    cout << LIGHTRED << "Text in LIGHTRED" << NO_COLOR << endl;
    cout << BLUE << "Text in BLUE" << NO_COLOR << endl;
    cout << GREEN << "Text in GREEN" << NO_COLOR << endl;
    cout << RED << WHITE_B << "Text in RED on WHITE background" << NO_COLOR << endl;
    cout << RED << BLACK_B << "Text in RED on BLACK background" << NO_COLOR << endl;
    cout << "\nPress <enter> to continue ..."; cin.get();

    cout << "Cursor is going to move to (20,3). Press <enter> to continue ..."; cin.get();
    gotoxy(20, 3);

    cout << "Screen is going to be cleaned. Press <enter> to continue ..."; cin.get();
    clrscr();

    // alternatively ...
    setcolor(LIGHTBLUE, WHITE_B); // OR cout << LIGHTBLUE << WHITE_B;
    cout << "From now on\n";
    cout << "everything is written LIGHTBLUE\n";
    cout << "on WHITE background\n";
    setcolor(NO_COLOR); // OR cout << NO_COLOR;
    cout << "End of program\n";
}
}
```

Linux version (using NCURSES library)

```
// PROG - MIEIC
// ZP
// Example of a program that prints colored characters on the console (in text mode)

#include <iostream>
#include <ctime>
#include <ncurses.h>

using namespace std;

/**
 * Called after using colored text
 */
void finish()
{
    resetterm();
    cout << std::endl;
}

/**
 * This method must be called before using colored text
 */
void init()
{
    short f, b;

    initscr();
    start_color();
    use_default_colors();

    // initialize default colors
    // COLORS is defined by Curses as the total number of colors
    for (f = 0; f < COLORS; ++f)
    {
        init_pair(f, f, -1);
        for (b = 0; b < COLORS; ++b)
            init_pair((short)(f * COLORS + b), f, b);
    }
    atexit(finish);
}

/**
 * Clear the entire screen
 */
void clrscr()
{
    clear();
    refresh();
}

/**
 * Output colored text to screen
 * @param text The text to output
 * @param x column in the screen
 * @param y row in the screen
 * @param fg_color Foreground color
 * @param bg_color Background color
 */
void write_text(const string& text, unsigned int x, unsigned int y, int fg_color, int bg_color = -1) {
    move(y, x);

    if (bg_color == -1)
        attron(COLOR_PAIR(fg_color));
    else
        attron(COLOR_PAIR(fg_color * COLORS + bg_color));

    addstr(text.c_str());
    refresh();
}

// Fill the screen with colored numbers
int main()
{
    srand((unsigned int)time(nullptr));
    char more;

    init();

    do
    {
        clrscr();
        for (unsigned int x = 0; x < 80; x++)
            for (unsigned int y = 0; y < 24; y++)
            {
                // COLOR_GREEN, COLOR_RED, ... are defined by Curses
                // taking values from 0 to COLORS (also defined there)
                write_text(to_string(x % 10), x, y, rand() % COLORS, COLOR_GREEN);
            }
        move(25, 0);
        refresh();
        cout << "more? (Y/N)";
        cin >> more;
    } while (toupper(more) == 'Y');
}
```

- Colors defined in **ncurses.h**: COLOR_BLACK, COLOR_RED, COLOR_GREEN, COLOR_YELLOW, COLOR_BLUE, COLOR_MAGENTA, COLOR_CYAN, COLOR_WHITE. Reference: <http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/color.html>