

A Borland C

A.1 Creating a new project

The following contains a step by step tutorial on how to create a new project in Borland C++.

1. Open Borland C++ Builder.
2. Click *File/New* and select Console Wizard.
3. Select source type C (instead of C++); you can still write your program in C++. Also be sure to uncheck the "multi-threaded" checkbox.
4. A new window will open with the filename Unit1.c. Delete the source code inside that window and start writing the program afresh. A sample program can be found in section C.5.
5. Include only the following header files in your source code:
iostream.h (C++) or stdio.h (C)
conio.h
cbw.h
6. After you have finished writing your program, save it in some directory on your network drive (probably Z:\). Select *Save as type C*". Save your file with the proper extension, i.e. as ".c" for C and as ".cpp" for C++.
7. Using *Project->Add to Project*, add the library file cbw32bc.lib to the project. Also make sure the file cbw32.dll is in the same folder as the rest of your project files (It does not need to be added to the project). These files are located on the course website.
8. Save the project again on your network drive.
9. Compile the project using *Project->Build XXX*.
10. Run the program from Borland C

A.2 conio.h

This include file contains console versions of common I/O functions. These functions bypass the *stdin* and *stdout* buffers and access the console directly. The following table (A.1) lists the most important functions that you might useful for your programming tasks. There are even more functions, refer to the include file.

Function Prototype Description

`kbhit int kbhit()` Check if a keyboard key was pressed.

`getch int getch()` Read a character directly from the console without echo.

`getche int getche()` Read a character directly from the console with echo.

`ungetch int ungetch(int c)` Put character `c` back into the keyboard buffer.

`putch int putch(int c)` Write character `c` to the console without buffering.

`cgets char* cgets(char* str)` Get a string directly from the console.

`cputs int cputs(const char* str)` Write a null-terminated string directly to the console.

`cprintf int cprintf(const char* str)` Format and print a string directly to the console.

`wherex int wherex()` Cursor position (x coordinate)

`wherey int wherey()` Cursor position (y coordinate)

`gotoxy void gotoxy(int x, int y)` Move cursor to new position (x, y).