Laboratory Session 1

Introduction

In this session, you will develop and test an actual computer program for the first time. You will be using the C language. As this is a high-level language (which the computer cannot understand as is), you will need to translate the program into a low-level or machine language that the Central Processing Unit (CPU) of your computer can actually execute. A special program called compiler performs this translation.

There are a wide variety of C compilers available for the Windows environment, each with their own particular strengths and weaknesses. You will be using Borland C++ compiler. Details about the compiling process are given in section 4 of this laboratory manual.

This session is also the first to require you to prepare a formal lab report. Section 3 of this laboratory manual provides an example of a report submission. The same format will be required also for the final computer-based exam report.

You will receive individual grades for each lab report and they will contribute to your overall assessment result for this module. It is advisable to write the report as you progress with the work during the lab session.

Create in your home directory (the H: drive) a working directory called for example **em108** and in that folder - a directory called **lab1**. It is strongly recommended to save all the materials related to the current lab session in **lab1** directory.

Copy the template of the lab report in **lab1** directory and open it using Microsoft Word. Keep the Word editor running throughout the lab session, so that you can add to it as you go along. Remember to save every time you add any significant amount of text, otherwise you run the risk of losing this text completely if, for example, there is a power failure in the lab! It is your responsibility to prevent this from happening.

Write your programs using "Textpad". Never use Word for this, as it saves the content in a different format than plain text as required by the C compiler. Save them using name representative for their content. For example, the file that stores the program, which prints the "Hello World" message, can be called "hello.c".

Task 1: Hello World! (40%)

Here is the minimal "Hello World!" program discussed in the lectures

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
  printf("Hello World!\n");
  return(EXIT_SUCCESS);
}
```

Use the text editor - Textpad - to create a file called "hello.c" in your working directory (H://em108/lab1) containing the program listed above.

Compile your program by giving the following command:

bcc32 hello.c

Automatically the compiler creates the compiled executable, calls it "hello.exe" and places it in the same directory. More details about compiling procedure are given in Section 4 of this manual.

Of course, "hello.c" should compile without any problems - if you have entered it correctly. But if any problems are encountered try to determine what was inputted wrong and fix the problem(s). If you have problems, seek help from the demonstrator by raising your hand.

To execute the program generated you simply give its name as a command:

hello.exe

or

hello

Check that the program executes as you expect. If it does not, try to figure out why, and fix the problem(s). Again, if you encounter problems, ask for help from the demonstrators.

If the program executes correctly, then congratulations - you have just entered, compiled, and tested your first C program!

Do not forget to write your report for task 1. A template is provided in Section 3.

Task 2: Branching out! (40%)

Here is the program called "insult.c":

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

int main(void)
{
  char name[100];
```

```
 printf("Welcome to INSULT ...\n");
 printf("Please enter your name:\n");
 scanf("%s", name);

 if(strcmp(name, "John") == 0)
  printf("Hello, %s. Welcome to my program!\n", name);
 else
  printf("I don't like you, %s. Good bye!\n", name);

 printf("Bye from INSULT ...\n");

 return(EXIT_SUCCESS);
}
```

Using this as a basis, or template, develop (plan, code, test, correct) a program called "sport.c" which will behave as follows:

- Displays a suitable welcome/identification message.
- Ask the user to type in his or her favourite sport (one word only).
- If the topic is different than "Soccer", display the message: "What? SPORT? Not good enough! I like Soccer!", where in the place of "SPORT" you display the sport entered by the user.
- If the topic is "Soccer", display the message "Good Choice! I like SPORT, too!" instead. In the place of "SPORT" you display the sport entered by the user ("Soccer" in this case!)
- Finally display some suitable closing message.

Record in your report your experiences in developing this program - especially whatever problems you encounter and how did you solve them.

Test the program with different inputs and record in your report the program's output.

Task 3: Variables, Operators, Expressions and Statements Quiz (20%)

Please finish the quiz "Lab1 Quiz" on the NEWTELP platform.

Conclusion

The major work of the lab session is now complete. The formal lab report, which you have prepared during the session, must be submitted by email. Perform the following steps:

- Read quickly back through the report, correcting any obvious errors, and possibly adding some brief concluding remarks.

- Exit from the text editor, saving the final version of the report

- Go to the submission system and upload the files: one code file for each of the tasks and the report with sections for each task:

- Create an email to yourself attaching the same files. Fill in the subject of the message as "Lab1 report". Attach the source files only: "hello.c" and "sport.c"

- Send the message. This serves as backup.