

# LAB 1: RUNNING THE C PROGRAM

## OBJECTIVES FOR THE STUDENT

---

1. To use the Borland C++ software version 5.02 to enter C source code, compile, link and run the C program.  
*[Untuk menggunakan perisian Borland C++ versi 5.02 untuk memasukkan kod sumber C, mengkompil, memaut, dan melaksanakan aturcara C]*
2. To identify the C program structure.  
*[Untuk mengenal pasti struktur aturcara C]*
3. To load, edit and recompile the C program.  
*[Untuk memuat, mengedit dan mengkompil semula aturcara C]*

## ASSUMPTIONS

---

1. Students should have a basic understanding of computer hardware and software.  
*[Pelajar sepatutnya mempunyai pengetahuan asas dalam perkakasan dan perisian komputer]*

## LAB EXERCISES

---

### EXERCISE 1:

#### [LATIHAN 1]

1. Starting the Computer.  
*[Menghidupkan komputer.]*
  - i. Booting is the term used to start up the PC (personal computer) by turning on the power. The boot instructions are placed in ROM by the computer manufacturer and cannot be altered by the user. These instructions first perform a quick check of the basic computer hardware.  
*[But (booting) ialah istilah yang digunakan untuk menghidupkan komputer peribadi (PC) dengan cara memasang kuasa elektrik. Arahan but terletak dalam ROM oleh pembuat komputer dan arahan ini tidak boleh diubahsuai oleh pengguna. Pada permulaannya, arahan ini melaksanakan pemeriksaan pantas terhadap perkakasan komputer yang asas.]*
  - ii. Next, the ROM BIOS checks to see if there is a system disk in drive A. If it is not found, it then checks to see if the system files are on drive C.

Once the system disk is found, a short set of instructions called the boot record is loaded into RAM. The boot record contains the information necessary to load the I/O handler into RAM.

*[Seterusnya, ROM BIOS memeriksa sama ada terdapat cakera sistem pada pemacu A. Jika ia tidak dijumpai, ia akan memeriksa sama ada fail sistem berada di pemacu C. Sebaik sahaja cakera sistem ditemui, satu set arahan pendek yang dinamakan rekod but dimuatkan ke dalam RAM. But rekod ini mengandungi maklumat penting untuk memuatkan pengendali I/O ke dalam RAM.]*

- iii. After the I/O handler is loaded, the system disk is then checked for the presence of the operating system files (i.e. CONFIG.SYS, COMMAND.COM and AUTOEXEC.BAT) and loads them into RAM.  
*[Setelah pengendali I/O dimuatkan, cakera system akan diperiksa bagi kehadiran fail-fail sistem pengoperasian (seperti CONFIG.SYS, COMMAND.COM and AUTOEXEC.BAT) dan memuatkannya ke dalam RAM.]*
  - iv. The PC that uses Windows 98, 2000 or XP, needs the user to enter a password in order to access any program/files in the computer. For the computer in the FSKSM lab, please get the password from the technician in charge.  
*[Komputer peribadi yang menggunakan sistem pengoperasian Windows 98, 2000 atau XP, memerlukan penggunaannya memasukkan kata laluan tertentu sebelum boleh mencapai mana-mana aturcara/fail yang terdapat di dalam komputer tersebut. Bagi komputer di makmal FSKSM, sila dapatkan kata laluan daripada juruteknik yang bertugas.]*
  - v. To boot and enter the password:  
*[Untuk but dan memasukkan kata laluan:]*
  - vi. Turn on all the switches to start your PC.  
*[Pasang semua suis untuk menghidupkan komputer anda]*
  - vii. Wait until a dialog box appears for you to enter the password.  
*[Tunggu sehingga kekotak dialog dipaparkan untuk anda masukkan kata laluan.]*
  - viii. Enter the appropriate password.  
*[Taipkan kata laluan yang bersesuaian.]*
  - ix. Press the Enter key.  
*[Tekan kekunci Enter.]*
2. Loading the Borland C++ software.  
*[Memuatkan perisian Borland C++.]*
- i. All programs to be run by the computer need to be loaded into the computer's memory, i.e RAM.

[Semua aturcara yang hendak dilaksanakan oleh komputer mesti dimuatkan ke dalam ingatan utama komputer, iaitu RAM.]

- ii. To load the Borland C++ software:  
[Untuk memuatkan perisian Borland C++:]
  - a. Click on the Start menu that is located at the bottom left of your screen.  
[Klik pada menu Start yang berada di bahagian bawah sebelah kiri skrin.]
  - b. Choose All Programs.  
[Pilih pada pilihan All Programs.]
  - c. Choose Borland C++ 5.02  
[Pilih pada pilihan Borland C++ 5.02]
  - d. Click on the Borland C++ program.  
[Klik pada aturcara Borland C++]
- iii. The Borland C++ software version 5.02 will be presented as below:  
[Perisian Borland C++ versi 5.02 akan dipamerkan seperti berikut:]

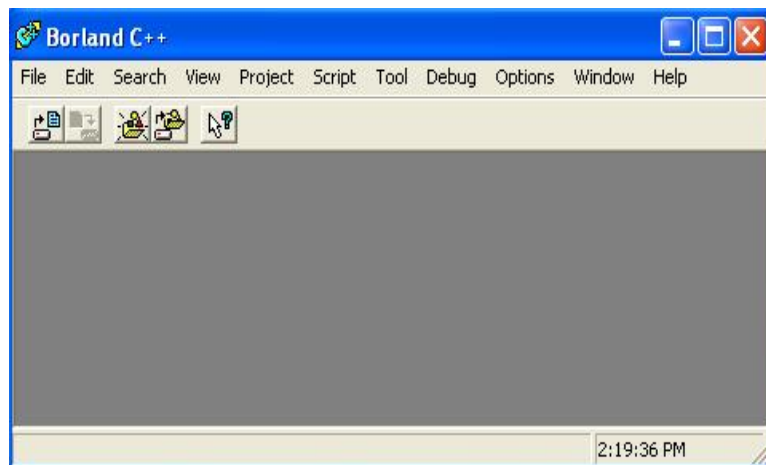


Figure 1.1 :

3. Writing the C source code.  
[Menulis kod sumber C.]
  - i. The Borland C++ software consists of an editor that permits the programmer to type, edit, and save programs in C, as well as C++ language.  
[Perisian Borland C++ mempunyai penyunting yang membenarkan pengaturcara menaip, menyunting/membaiki dan menyimpan aturcara dalam bahasa C dan juga C++]
  - ii. To write a new program, click on the **File** menu, choose **New**, and **Text Edit** option as follows:

[Untuk menulis aturcara yang baru, klik pada menu **File**, pilih **New** dan **Text Edit** seperti berikut:]

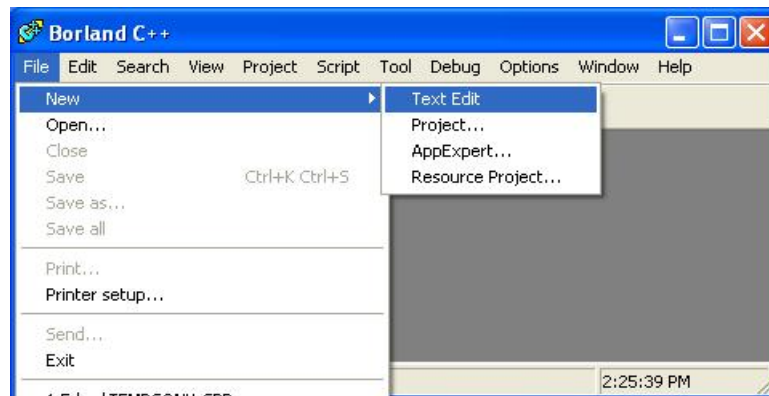


Figure 1.2 :

- iii. The following window will appear.  
 [Tetingkap berikut akan dipamerkan.]

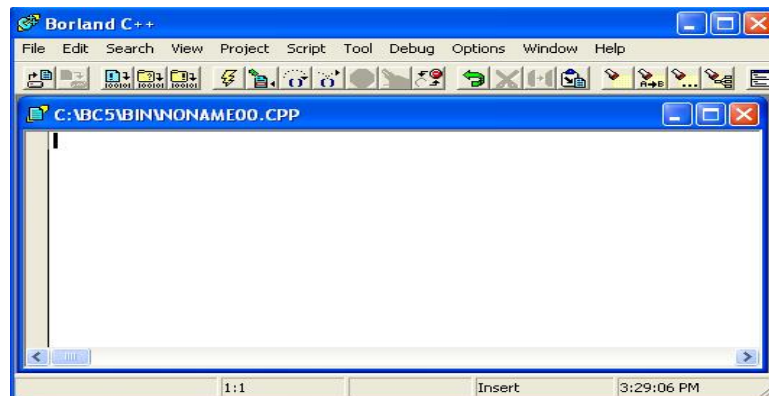


Figure 1.3 :

- iv. Type the following C statements. Do not type the line number.  
 [Taipkan pernyataan C seperti berikut. Jangan taipkan nombor baris.]

```

1 //Program 1.1
2 //Nama: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio.h>
6
7 int main(void)
8 {
9     printf ("Hai!! Apa khabar?");
10    getch();
11    return 0;
    }
    
```

4. Saving the C source file.  
[Menyimpan fail sumber C.]
- i. The program that you just wrote using the C editor is called the source program. This source program resides in the computer's main memory (RAM). You need to save the source program into the secondary storage (such as the hard disk, diskette, or pen drive) as a backup so that you can use the program file at other time.  
[Aturcara yang telah ditulis menggunakan editor C dikenali sebagai aturcara sumber. Aturcara sumber ini berada di dalam ingatan utama computer (RAM). Anda perlu menyimpan aturcara sumber ini ke dalam storan sekunder (seperti cakera keras, disket, atau pemacu pen) sebagai sokongan supaya anda boleh menggunakan fail aturcara ini pada masa yang lain.]
  - ii. To save a source program into a new folder in the hard disk.  
[Untuk menyimpan aturcara sumber ke dalam folder baru di cakera keras:]
    - a. Click at the **File** menu. Choose **Save As**.  
[Klik pada menu **File**. Pilih pilihan **Save As**.]

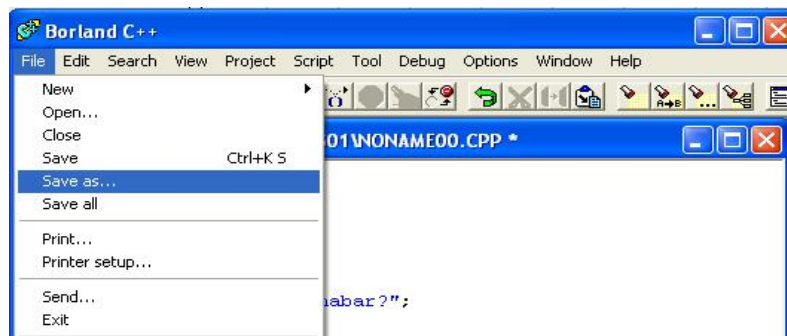


Figure 1.4 :

- b. To create a new folder so that it will be located under the Bin folder, click on the icon **Create New Folder** (as circled) in the **Save File As** dialog box.  
[Untuk mencipta folder baru supaya berada di bawah folder Bin, klik pada ikon **Create New Folder** seperti yang dibulatkan pada bahagian kotak dialog **Save File As** berikut:]

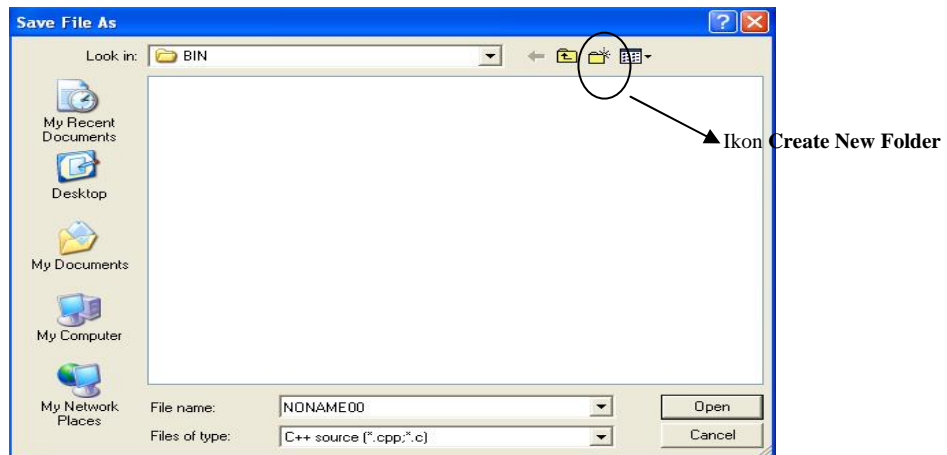


Figure 1.5 :

- c. In the **New Folder**, type the folder name keep all of your program files in that folder.  
*[Di bagian **New Folder**, taipkan nama folder yang akan menyimpan fail-fail aturcara anda]*
- d. Click twice at your folder and type the name of your source program at the File name input box, as Hello1.c.  
*[Klik dua kali pada folder anda itu dan kemudian taipkan nama fail aturcara sumber anda sebagai Hello1.c dan kotak input **File Name**]*

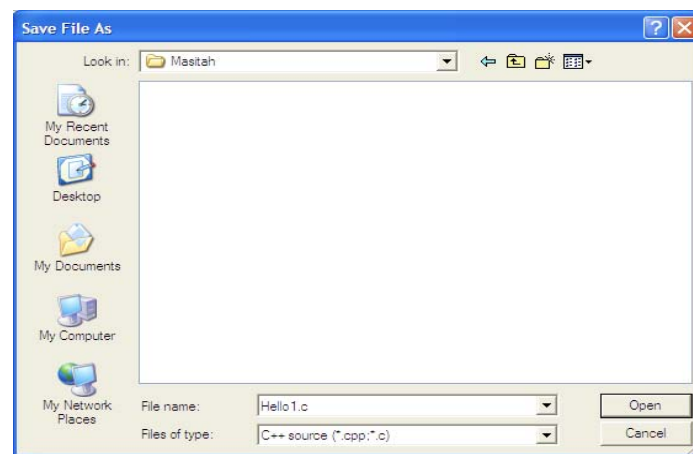


Figure 1.6 :

- e. Click on the button **Open** to save the file.  
*[Klik pada bebutang **Open** untuk menyimpan fail.]*
5. Compiling the C program.  
*[Mengkompil aturcara C.]*
- i. After you created a program, you can compile it so that the source code will be translated into object code. The C compiler will try to detect any syntax errors in your program.

[Setelah siap menaip aturcara, anda boleh mengkompil aturcara tersebut untuk menterjemahkan kod sumber kepada kod objek. Pengkompil bahasa C akan cuba mengesan sebarang ralat sintak yang terdapat di dalam aturcara.]

- ii. To compile a C program:  
[Untuk mengkompil aturcara C]
  - a. Click on the **Project** menu and select the **Compile** option, or simultaneously press the **Alt** key and the **F9** function key. The following screen will appear:  
[Klik pada menu **Project** dan pilih pilihan **Compile** atau tekan pada bebutang **Alt** dan kekunci fungsi secara serentak skrin berikut akan dipaparkan.]

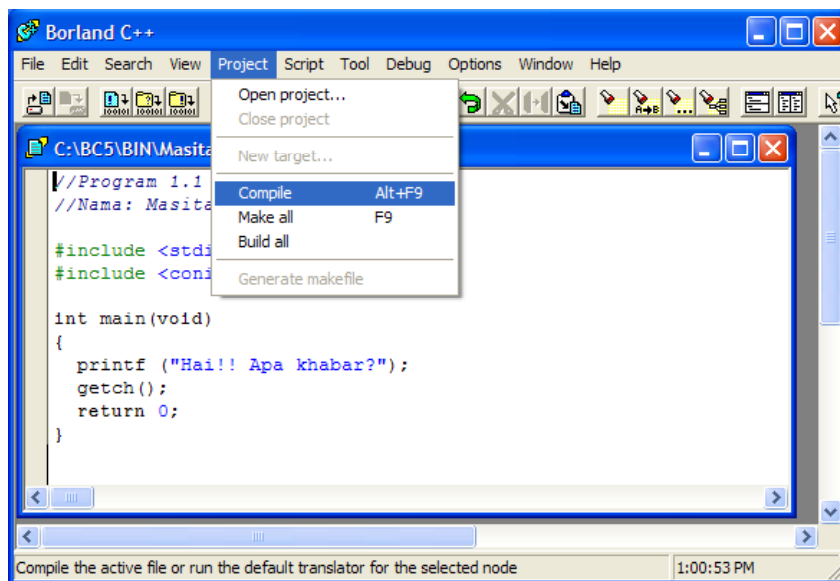


Figure 1.7 :

- b. If there is no syntax error, the compilation process is considered successful.  
[Sekiranya tiada ralat sintaks ditemui, proses pengkompilan dianggap berjaya]

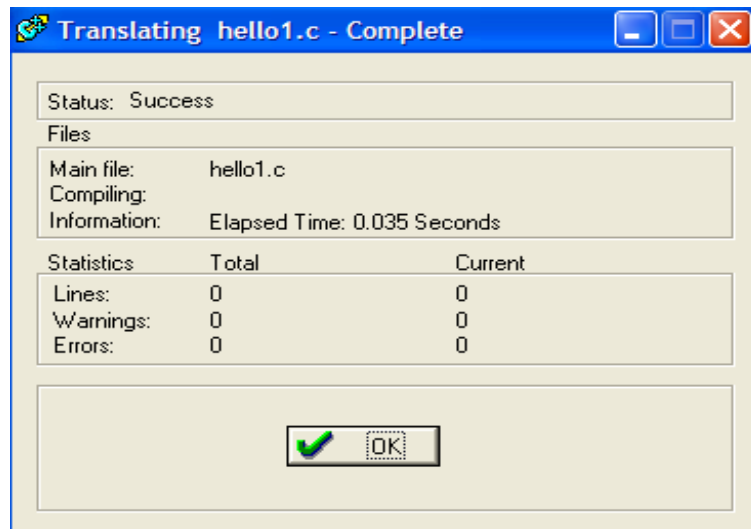


Figure 1.8 :

- c. If there are syntax errors, an error message will be displayed and the programmer will edit the code with errors and recompile the program until successful.  
*[Jika terdapat ralat sintaks, mesej ralat akan dipamerkan dan pengaturcara perlu memperbaiki arahan yang terdapat ralat dan mengkompil semula aturcara tersebut sehingga berjaya.]*
- iii. Click on the **OK** button.  
*[Klik pada bebutang **OK**]*
6. Running the C program.  
*[Melaksanakan aturcara C.]*
- i. The program that has successfully been compiled, can be executed (run) so that the result will appear.  
*[Aturcara yang telah berjaya dikompil perlu dilaksanakan (run) supaya boleh menghasilkan keputusan.]*
- ii. However, if error occurs, an error message will be displayed and the programmer need to edit and recompile, as well as rerun the program to produce the output.  
*[Bagaimanapun, jika terdapat ralat, mesej ralat akan dipamerkan dan pengaturcara perlu membaikinya dan mengkompil semula serta melaksana semula aturcara tersebut untuk menghasilkan output.]*
- iii. To run the program:  
*[Untuk melaksanakan aturcara:]*
- a. Click on the **Debug** menu and choose **Run**, or press the **Ctrl** and the **F9** function key simultaneously.  
*[Klik pada menu **Debug** dan pilih **Run**, atau tekan pada bebutang **Ctrl** dan kekunci fungsi **F9** secara serentak]*



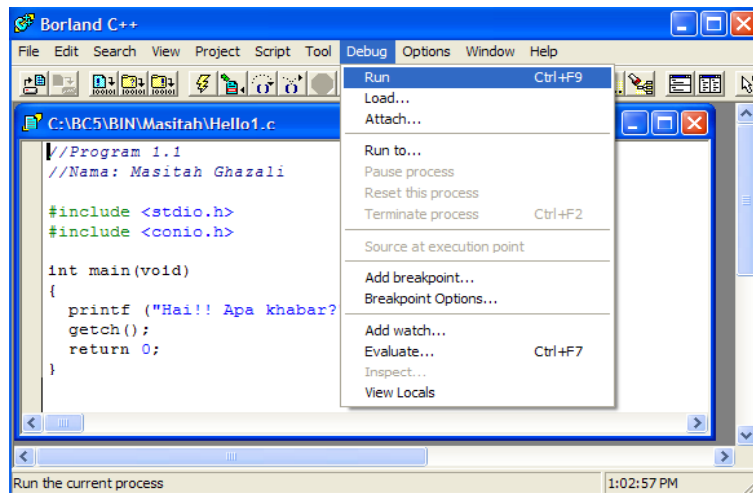


Figure 1.9 :

- b. The output is displayed in a DOS window, as shown below:  
 [Output akan dipaparkan pada tettingkap DOS seperti berikut:]

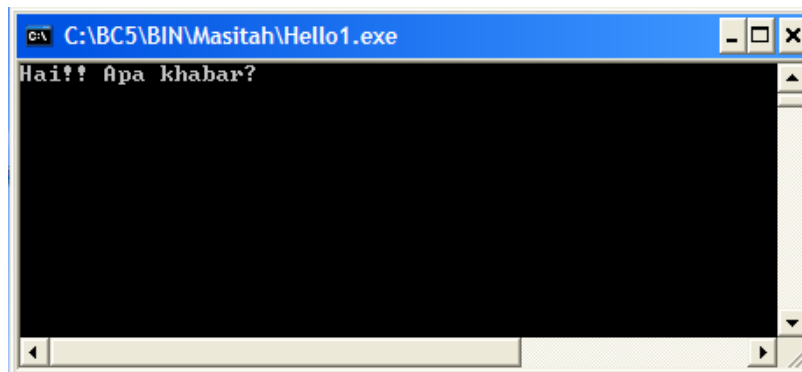


Figure 1.10 :

7. Close the C program.  
 [Tutup aturcara C.]
- i. The opened program source can be closed and reopened whenever required.  
 [Aturcara sumber yang sedang dibuka boleh ditutup dan boleh membukanya semula apabila diperlukan.]
  - ii. Step for closing the opened source program:  
 [Cara untuk menutup aturcara sumber yang sedang dibuka:]
    - a. Click the **File** menu and chose **Close**, or click the cross image at the top right hand corner the window.  
 [Klik pada menu **File** dan pilih **Close**, ataupun klik pada tanda pangkah di bahagian atas sebelah kanan tettingkap aturcara sumber tersebut.]

**EXERCISE 2:****[LATIHAN 2]**

Often you need to load existing C source file, edit the content and compile it.

[*Seringkali anda perlu memuatkan fail sumber C sedia ada, mengedit kandungannya serta menkompilnya.*]

1. Load the Borland C++ software as in Exercise 1 Part 2 (see page 3).  
[*Muatkan perisian Borland C++ seperti Latihan 1 Bahagian 2. (Lihat halaman 3)*]
2. Open the existing C source file as follows:  
[*Buka sumber fail C yang wujud seperti berikut:*]
  - i. Click on the **File** menu and choose **Open** or click on the **Open a file** icon as shown in figure 1.11 below.  
[*Buka fail sumber C sedia ada seperti berikut pada menu File dan memilih Open atau dengan mengklik pada ikon Open a file seperti yang ditunjukkan oleh rajah 1.11 berikut.*]

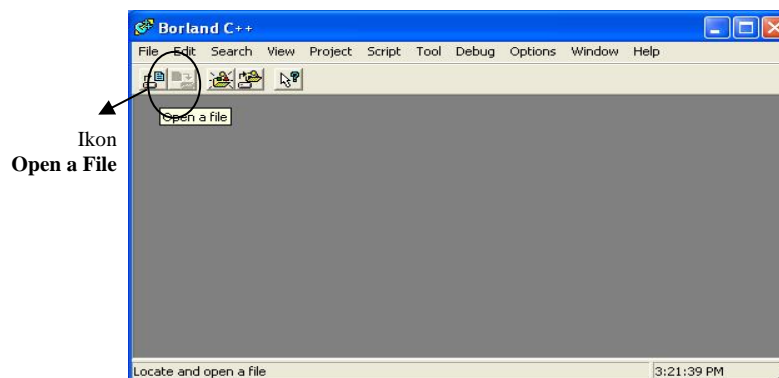


Figure 1.11 :

- ii. Once the dialog box appears, type the appropriate file name as shown in figure 1.12 below.  
[*Apabila kekotak dialog dipamerkan, taipkan nama fail seperti tertera dalam tajah 1.12 berikut.*]

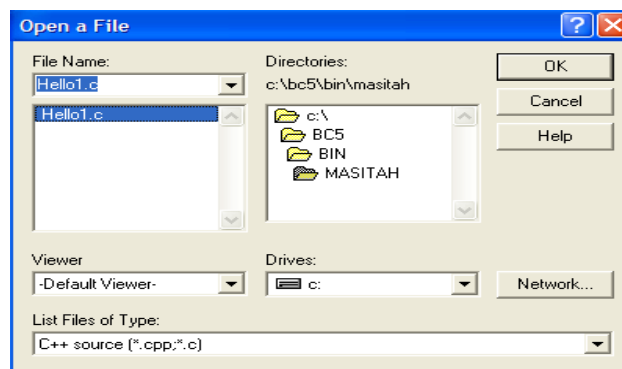


Figure 1.12 :

- iii. Add a new line of code at line 9:  
[*Tambah satu barisan kod baru pada baris 9:*]

```
1 //Program 1.2
2 //Nama pengaturcara: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio.h>
6
7 int main(void)
8 {
9     printf ("Hai!! Apa khabar?\n");
10    printf ("Ucapan dari Masitah Ghazali\n");
11    getch();
12    return 0;
}
```

3. Compile the C program as in Exercise 1 Part 5. (See page 7)  
[*Mengkompil aturcara C seperti Latihan 1 Bahagian 5. (Lihat halaman 7)*]
4. Run the C program as in Exercise 1 Part 6. (See page 9)  
[*Melaksana aturcara C seperti Latihan 1 Bahagian 6. (Lihat halaman 9)*]
5. Show the output of program 1.2 above.  
[*Tunjukkan output bagi aturcara 1.2 di atas.*]
6. Close the C program as in Exercise 1 Part 7. (See page 10)  
[*Tutup aturcara C seperti Latihan 1 Bahagian 7. (Lihat halaman 10)*]
7. Shut down the computer system.  
[*Tutup sistem komputer.*]
  - i. Close all application programs before shutting down the computer.  
[*Pastikan semua perisian telah ditutup sebelum menutup komputer anda.*]
  - ii. Click on the **Start** menu and choose **Turn Off Computer**.  
[*Klik pada menu **Start** dan pilih **Turn Off Computer**.*]
  - iii. Click pada ikon **Turn Off** as in figure 1.13.  
[*Klik pada ikon **Turn Off** seperti rajah 1.13.*]



Figure 1.13 :

**EXERCISE 3:****[LATIHAN 3]**

1. The annotation of program 1.1 is as in page 5 will be as follows:  
[Penerangan bagi aturcara 1.1 seperti di halaman 5 adalah seperti berikut:]

```

1 //Program 1.1
2 //Nama: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio.h>
6
7 int main(void)
8 {
9     printf ("Hai!! Apa khabar?\n");
10    getch();
11    return 0;
12 }
```

- i. Line 1 and 2 are **comments**. A **comment** is a piece of descriptive text (letters, number or special symbols) which explains some aspect of a program. Program comments are considered as **non-executable** statements that are totally ignored by the compiler and are only intended for human readers. In program 1.1 above, the symbol // are used to begin the comment in a line.  
[Baris 1 dan 2 merupakan suatu **komen**. **Komen** teks (huruf, nombor atau simbol-simbol) yang menerangkan mengenai aspek-aspek aturcara. Komen dalam aturcara merupakan penyata **bukan pelaksana** (non-executable statement) yang akan diabaikan oleh pengkompil C dan ditunjukkan hanya untuk pembaca aturcara sahaja. Pada aturcara 1.1 di atas, penyata komen dimulai dengan simbol // ]
- ii. Lines 4 and 5 are the **preprocessor directive** #include that include the contents of the library file, namedly `stdio.h` and `conio`, in the program. `stdio.h` is a standard C header file and contains definitions for input and output, such as `printf` and `scanf`. While

the `conio` is another header file that contains functions, such as `getch()`, that accepts an input data from the keyboard.

[Baris 4 dan 5 ialah **arahan prapemproses** yang akan menyertakan kandungan bagi fail perpustakaan bernama `stdio.h` dan `conio`, ke dalam aturcara. `stdio.h` ialah fail kepala C yang piawai dan mengandungi penakrifan input dan output seperti `printf` dan `scanf`. Manakala `conio` pula ialah satu lagi fail kepala yang mengandungi fungsi-fungsi, seperti `getch()`, yang menerima data masukan daripada papan kekunci.]

- iii. Line 7 defines a function called **main** that begins the program execution. The brackets without any word indicate the main function has no parameters. The return type for main is `int` (i.e., an integer number).

[Baris 7 menakrifkan satu fungsi bernama **main** yang memulakan pelaksanaan aturcara. Tanda dua kurungan tanpa sebarang perkataan di antaranya menunjukkan fungsi main tidak mempunyai sebarang parameter. Jenis data yang dikembalikan oleh fungsi main ialah `int` (iaitu, suatu nombor integer)]

- iv. Line 8 is an open brace that marks the beginning of the body of main function.

[Baris 8 merupakan pembuka kurungan berkeluk yang menandakan permulaan badan fungsi main.]

- v. Line 9 is a **statement**. The end of a statement is always marked with a semicolon (;). This statement causes the **string** "Hai!! Apa khabar?" to be sent to the `printf` **output stream**. A string is any sequence of characters enclosed in double-quotes. The last character in this string (`\n`) is a newline character which returns the cursor to the new line. A stream is an object which performs input or output. `printf` is the standard output stream in C (standard output usually means your computer monitor screen).

[Baris 9 ialah suatu pernyataan. Pernyataan lazimnya diakhiri dengan satu tanda semikolon (;). Pernyataan ini menyebabkan rentetan "Hai!! Apa khabar?" dihantar kepada aliran (stream) output `cout`. Suatu rentetan ialah sebarang turutan aksara yang dikurungkan dengan tanda petik berkembar "quote". Aksara terakhir pada rentetan ini (`\n`) merupakan satu aksara baris baru yang akan membawa kursor kembali ke baris baru. Aliran (stream) ialah suatu objek yang akan melaksanakan input atau output. `printf` ialah aliran output piawai bagi C. Dalam kes ini, kesan operasi ini ialah rentetan "Hai!! Apa khabar?\n" dihantar kepada `printf`, menyebabkannya dipaparkan ke skrin computer].

- vi. Line 10 is another statement that is a function `getch()` from the library `conio`. `getch()` will prompt and wait for the user enters any key at the keyboard.

[Baris 10 adalah satu lagi pernyataan iaitu fungsi bernama `getch()` daripada perpustakaan `conio`. `getch()` akan menggesa dan menunggu sehingga pengguna menekan sebarang kekunci pada papan kekunci]

- vii. Line 11 is the **return** statement that the control of the program to the operating system. It also determines the value of the return type that match the main function.

[Baris 11 ini ialah pernyataan **return** yang akan mengalihkan kawalan daripada aturcara kepada sistem pengoperasian. Ia juga akan menentukan nilai bagi fungsi yang akan dipulangkan dan perlu berpadanan dengan jenis data pulangan bagi fungsi main.]

- viii. Line 12 is a close brace that marks the end of the body of main.

[Baris 12 merupakan penutup kurungan berkeluk yang menandakan akhir bahagian badan fungsi main.]

2. Given program 1.3 below:

[Diberi aturcara 1.3 berikut:]

```
1  /* Program 1.3
2  Nama pengaturcara: Masitah Ghazali
3  Tujuan aturcara : Masukkan nama dan tahun lahir
4  serta kira usia pengguna pada tahun ini */
5
6  #include <stdio.h>    //Pre-processor directive
7  #include <conio.h>   //Pre-processor directive
8
9  int main()
10 {
11     const int TahunIni=2010;
12     char nama[20];
13     int tahunlahir, umur;
14
15     printf ("Masukkan nama anda: ");
16     scanf ("%s", &nama);
17
18     printf ("Masukkan tahun anda dilahirkan: ");
19     scanf ("%d", &tahunlahir);
20
21     umur = TahunIni - tahunlahir;
22
23     printf ("Nama anda: %s", nama);
24     printf ("\nUsia anda: %d", umur, "tahun.");
25     getch();
26     return 0;
27 }
```

- i. Type, save, compile, and run program 1.3 above. What will be the output of the program for the given input?

[*Taip, simpan, kompil, dan laksanakan aturcara tersebut. Apakah output aturcara di atas bagi input yang dimasukkan?*]

- ii. Briefly annotate what each line of code means.  
[*Terangkan secara ringkas maksud setiap baris kod.*]

3. Given program 1.4 below:  
[*Diberi aturcara 1.4 berikut:*]

```
1 //Program 1.4
2 //Nama pengaturcara: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio>
6
7 int main (void)
8 {
9     int workDays;
10    float workHours, payRate, weeklyPay;
11
12    workDays = 5;
13    workHours = 6.5;
14    payRate = 20.50;
15    weeklyPay = workDays * workHours * payRate;
16    printf ("Weekly Pay = %f", weeklyPay);
17    printf ("\n");
18    getch();
19    return 0;
20 }
```

- i. Type, save, compile, and run the program. What will be the output of the program?  
[*Taip, simpan, kompil, dan laksanakan aturcara tersebut. Apakah output bagi aturcara di atas?*]
- ii. Briefly annotate what each line of code means.  
[*Terangkan secara ringkas maksud setiap baris kodnya.*]