S&T Batch-1 C Programming Lab 1

Dev C Compiler, scanf(), printf()

24 Jan 2025 Friday 6:50pm at S13-M-09 – Computer Lab

Bring along a thumb drive to the lab.

This document is produced by Hu Jialun

You never trust a computer!
- A/Prof Tay Seng Chuan

Introduction

Dev-C++ is a simple integrated development environment (IDE) for the C and C++ programming languages suitable for beginners. This compiler is largely modernised by Embarcadero Technologies. In this Computer Bridging Course, we are going to use only C language portion and make use of the compiler and a few debugging features. Our purpose is not to become experts in any IDE but to learn the C Programming Language.

Legend

Monospaced text is verbatim. In other words, look for or input the exact same text without changing any single character. This is typically used for code and other machine-readable text. <Text in angle brackets> is meant to be replaced with your own information. For example, NUSSTU\E<your 7 digits> may be replaced by NUSSTU\E15xxxxx

Bold text indicates UI controls on your screen, and the **>** symbol indicates menu hierarchy. For example, **File>Save** points you to the "Save" item under the "File" menu indicates keystrokes on your keyboard. As an example, + means to press the Control key and the C key together.

Reminders

- Memorise your own NUSNET ID and password for logging in to Windows on lab computers.
- Smartphone/tablet with Microsoft Authenticator setup for 2FA (Two factor authentication).
- Read this pdf file. Save paper - Think before you print.
- Use a thumb drive to save your work.

Logging in

The lab computers are connected to and authenticated via NUS intranet domains. To log in, use your NUSNET ID (the same one you use for NUS_STU WiFi and NUS Webmail, starting with E and followed by 7 digits, like E15xxxxx) and password.

Username: NUSSTU\E<your 7 digits>
Password: <Your password>

Error

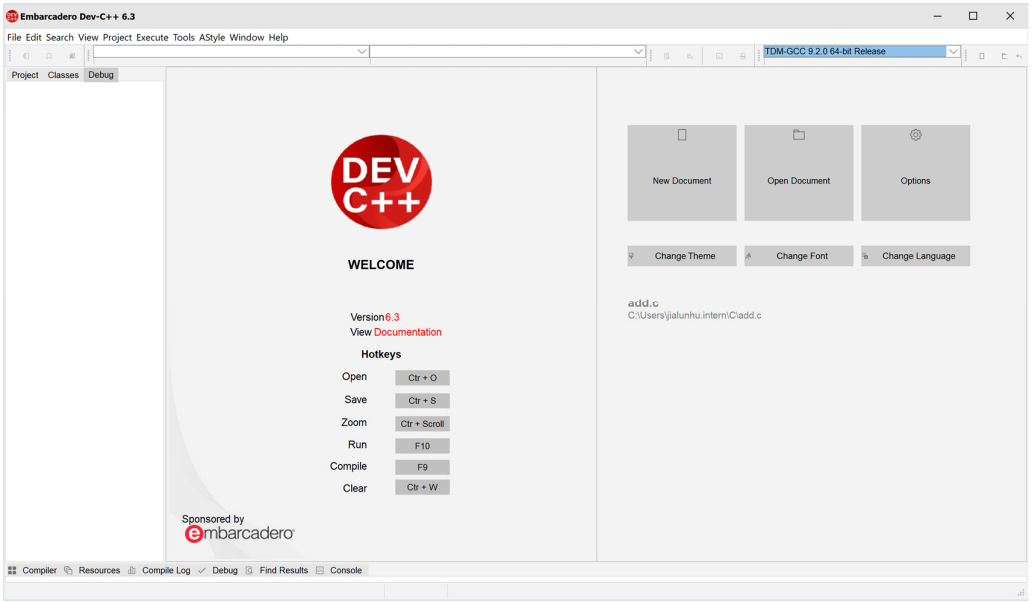
TODO: Most likely due to wrong password. You should try again.

让我们说中文?

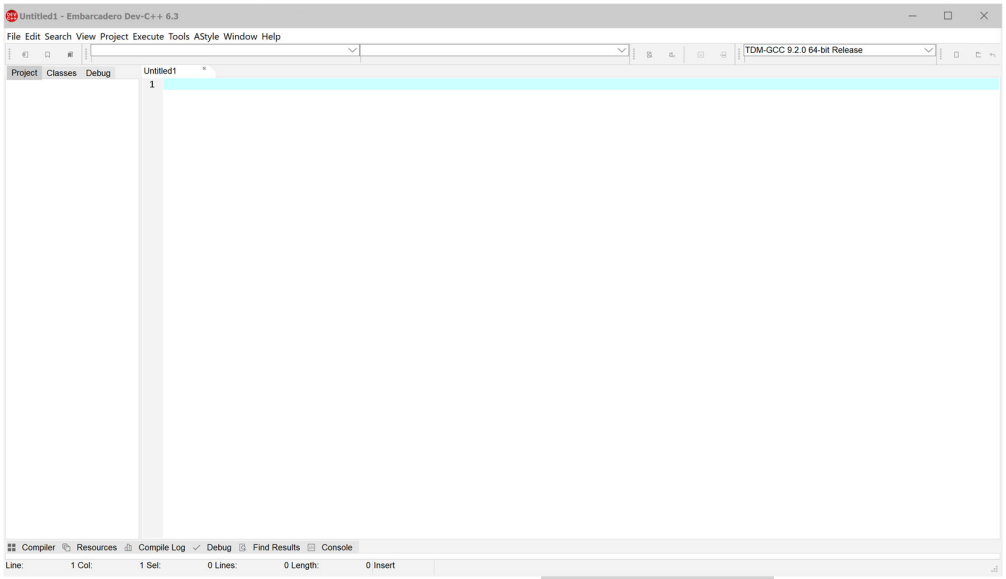
DO NOT change the display language of the computer from English. This setting is global and persistent and will affect subsequent lab users.

Launching Embarcadero Dev-C++

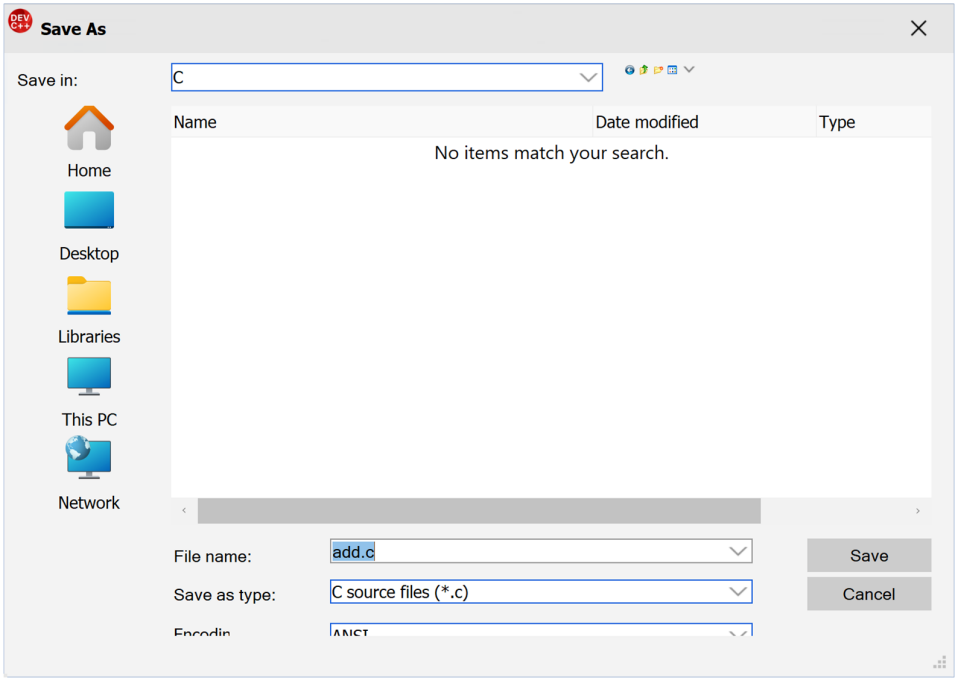
After logging in, you should find a shortcut to "Embarcadero Dev-C++" on your desktop. Double-click on it to open. Alternatively, use the search box on the taskbar to search for the name, and click on the entry to launch the compiler.



Click on **New Document** to create a new file. You now have a transient file named Untitled 1.

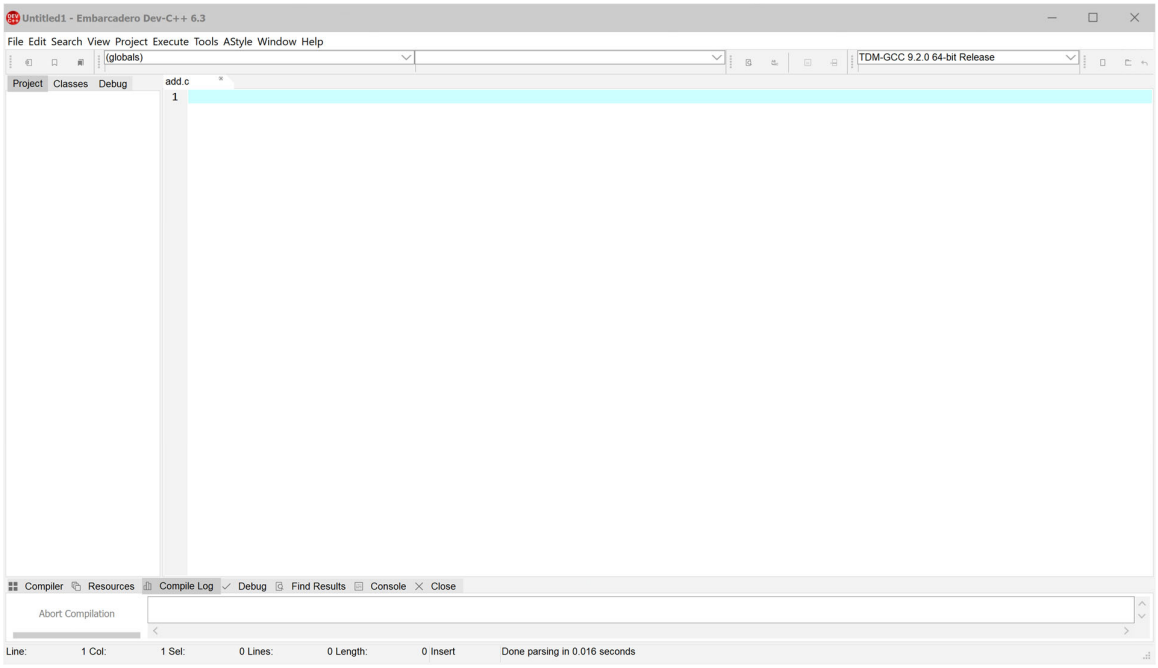


Save it onto the disk with **File>Save As**. Choose C source files (*.c) for "save as type". You can input add.c as the filename. Save the file somewhere you can find, such as inside your desktop folder. Click **Save** afterwards.



You should now be greeted by an empty file named add.c in the editor, as reflected on the tab title. [1]

[1]This may resemble Notepad because it really just is. Most source code files are no different from .txt files other than the file extension, and IDE is basically Notepad on steroids, packed with additional features to work with the text files known as source code.



Attention

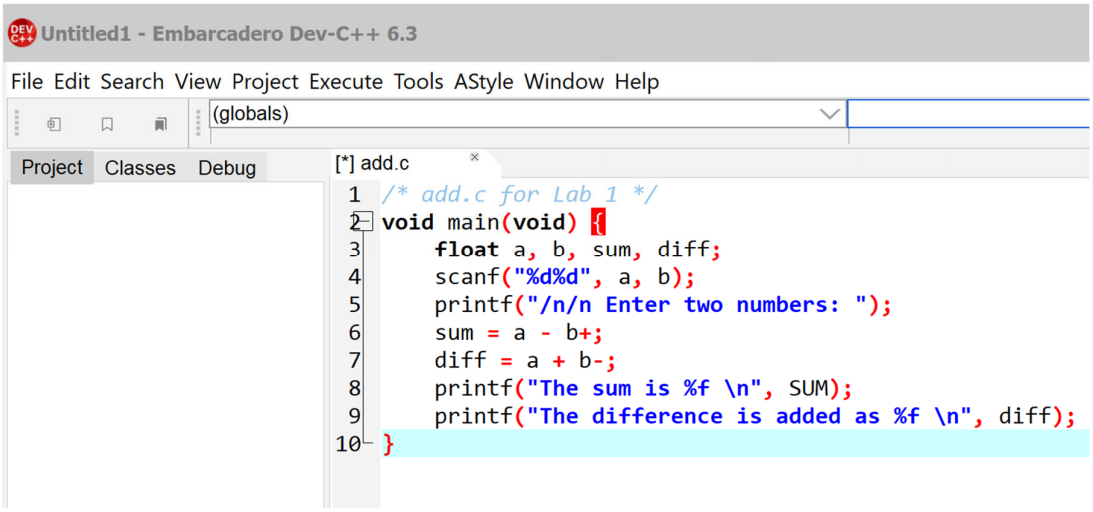
Save your files regularly.

First Program

Input the program below into the text editor the way you do it in Notepad. The syntax highlighting may be different.

```
/* add.c for Lab 1 */
void main(void)
{
   float a, b, sum, diff;

    scanf("%d%d", a, b);
    printf("/n/n Enter two numbers: ");
    sum = a - b+;
    diff = a + b-;
    printf("The sum is %f \n", SUM);
    printf("The difference is added as %f \n", diff);
}
```

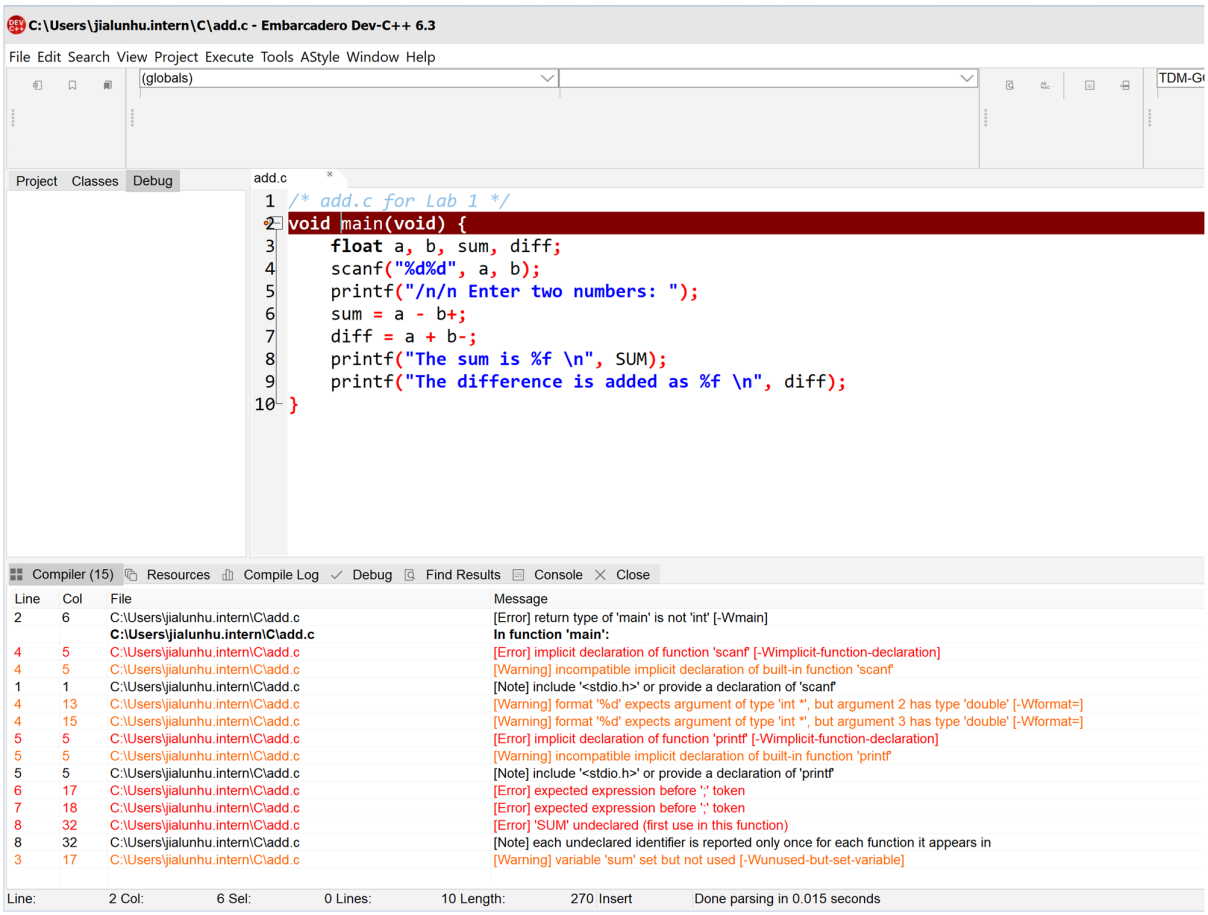


To compile, use **Execute>Compile** ().

Fix Compilation Errors and Warnings

The compiler helps you identify syntax errors and potentially erroneous constructs aka warnings. It is almost always correct and you should **always** fix **all** errors and warnings unless you have a good reason to do otherwise (unlikely at this level).

Double-clicking on an error listing brings your cursor to the exact position of that error. Start with the first error, as later errors may have been caused by the first error itself.



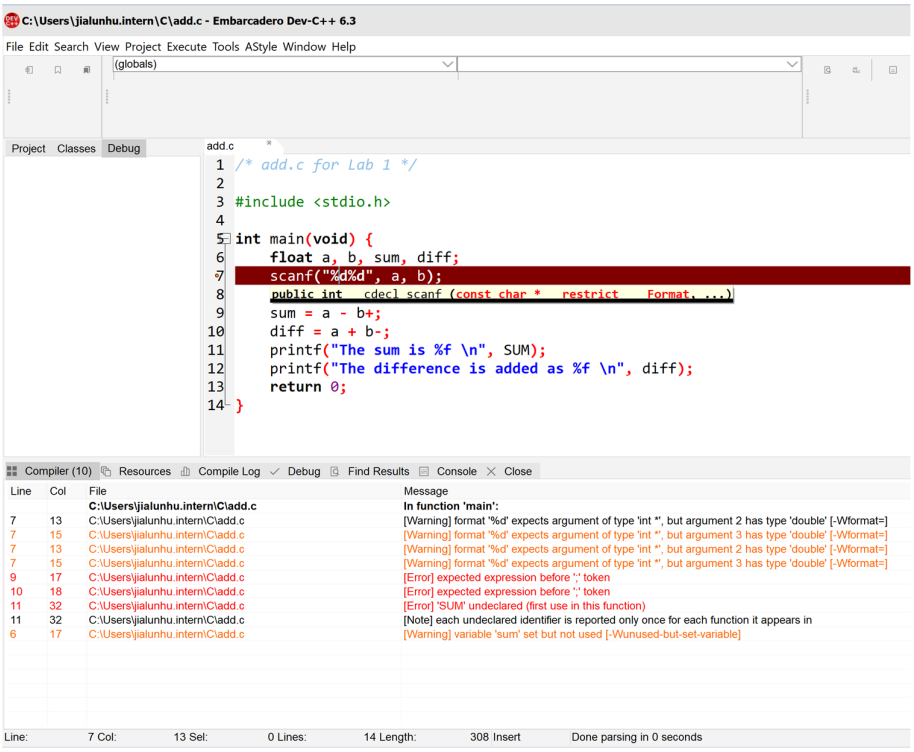
The first one is straightforward: the `main` function is supposed to return a `int`. We change its return type to `int` and add `return 0;` to signify success at the end [2].

The next error [3], "implicit declaration of function", indicates that the compiler had not seen the function declaration by the time you attempted to invoke the function.

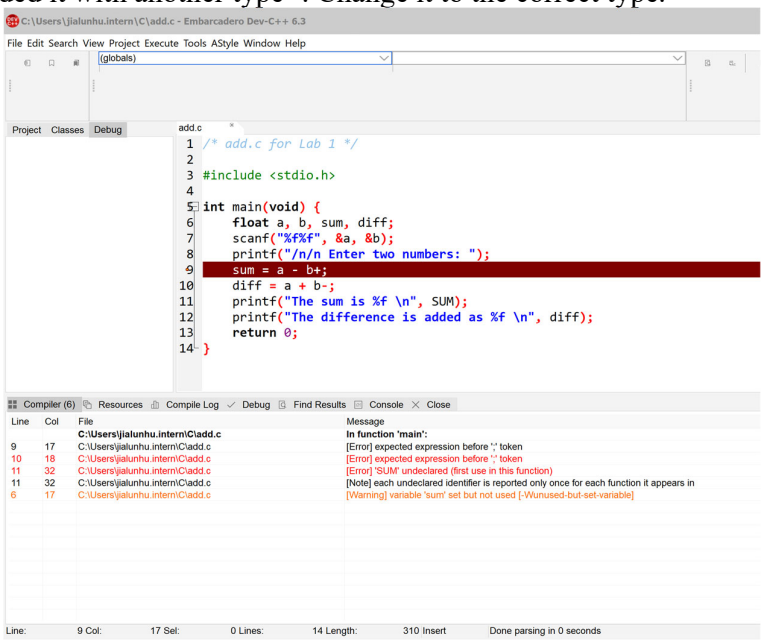
It usually means that you either mistyped the function name, or forgot to include the header in which the function is declared. In this case, it is because we did not include `stdio.h` upfront.

[2]Technically the C standard permits the return statement to be omitted from `main`, and `return 0;` is implicit, but this is an obscure special case only for `main`. Do not omit it in your exam!

[3]Functions without declarations used to be allowed historically to save some bytes in the 1970s and the `int` return type is assumed, but it has been made an error since C99 and you definitely should avoid this nowadays.



Following up, a warning about format specifiers. As the message says, %d corresponds to int *, but we provided it with another type [4]. Change it to the correct type.



The next does not really tell much and is a bit cryptic. Such messages typically mean that the compiler cannot make sense of the syntax we have written, and normally the culprit comes

[4]The message says double instead of float due to default argument promotions of variadic functions.

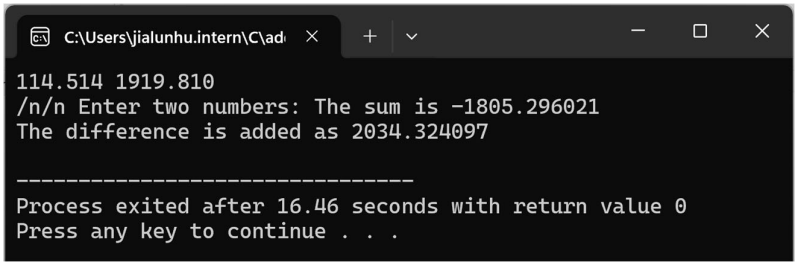
immediately before the place of error. In this case, it is because of the redundant trailing +. The error message actually means that the compiler expected another operand after the + operator, but encountered the end-of-statement marker ; before it could find one.

Refer to https://nus-cs1010.github.io/2425-s1/guides/messages.html for a list of common compilation errors and warnings.

Execution
After compilation passes without errors, you can run the resulting executable (.exe file) by **Execute>Run** (). A terminal window will pop up for your program to read input (scanf etc) and print output (printf etc).

As a shortcut, you can also use **Execute>Compile & Run** () to compile then run the program in one go.

Input two numbers and press . You should see something like this.



Assignment 1

Write a program named compute.c to ask the user to enter eight integers, and print the following contents on the screen:

- Sum of the square of each odd term (1st, 3rd, 5th and 7th)
- Sum of the square of each even term (2nd, 4th, 6th and 8th)
- Difference between the above two sums

The example screen output is shown below:

> Enter Eight Integers: 8 3 6 2 4 5 9 1
> Sum of the square of each odd term: 197
> Sum of the square of each even term: 39
> Difference of two sums: 158

Hint
The following instructions will help you in writing the program.

1. Print on the screen the message asking the user to enter 8 integers (use printf).
2. Scan in 8 integers from the keyboard (use scanf).

3. Compute sum of the square of each odd term, sum of the square of each even term, and the difference between the two sums.
4. Print the complete output on the screen.

Assignment 2

Write a program named as ask.c that informs the user to type an integer from the keyboard, and the program will check whether the integer is even or odd as shown on the screen:

```
Enter a positive integer not greater than 2147483647: 70
>> 70 is even.
Enter a positive integer not greater than 2147483647: 253
>> 253 is odd.
Enter a positive integer not greater than 2147483647: -5
>> Data is not accepted. Please obey the instruction.
```

Cleanup

No submission is required for practical sessions. Although I will explain the solution on the next session, you should try your best to complete in at your own time.

Save your two .c files somewhere accessible for future reference, and I would sugges you save them in your thumb drive, or you send them to your NUS email.

You need not save other files such as .obj and .exe files, as they are artifacts built from the .c sources.

Log out of your NUS account from the lab computer before you leave. This is done as follows:

1. Select the Windows icon on the taskbar
2. Select the shutdown button
3. Select Shutdown