UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Computer Programming 143

Week 1 — Practical 0

2016

Aim of Practical 0:

Use two assignments (Assignment A and B) to

(a) Introduce students to the available facilities.

(b) Introduction on how to use the Eclipse IDE to create C projects.

"In the beginning the Universe was created. This has made a lot of people very angry and been widely regarded as a bad move." - Douglas Adams

"Everything starts somewhere, although many physicists disagree." - Terry Pratchett, Hogfather

Instructions

1. Attendance is **compulsory** for all the practical sessions of your assigned group. See the study guide for more details.
2. The last section (usually the last 30 minutes) of the practical will be used for a test.
3. If more than three tests have been missed for what ever reason, you will receive an **INCOMPLETE** for the subject. See the study guide for more details.
4. You must do all assignments **on your own**. Students are encouraged to help each other **understand** the problems and solutions, but each should write his/her own code. By simply copying someone else's code or solutions, you will not build an understanding of the work.
5. You are responsible for your own progress. Ensure that you understand the practical work. Check your work against the memorandum that will be posted on Wednesday afternoons on learn.sun.ac.za.
6. Use H:\CP143 as the Eclipse workspace folder for all projects but it is highly suggested that you use a **flash drive to backup** all work done.
7. Create a new project **for each assignment**. See *Creating an Eclipse Project* on page 9 for instructions on how to do just that.
8. Include a comment block at the top of each source file according to the format given. It must include the correct filename and date, your name and student number, the copying declaration, and the title of the source file.
9. **Indent your code correctly!** Making your code readable is not beautification, it is a time and life saving habit. Adhere to the standards (refer to the documents on SUNLearn). You can use Ctrl+A and then Ctrl+I to auto-indent.
10. Comment your code sufficiently well. It is required for you and others to understand what you have done.

Opening the Eclipse IDE

Start Eclipse

- Once logged onto the computer, click on the **Start** button -> **All Programs** -> **Development & Programming** -> **eclipse C++**. Make sure you launch **eclipse C++** and not **Eclipse Java Mars** – the latter is set up for programming in Java.
- A **security warning** might appear. Just click the **Run** button.
- You will be asked to enter a path to a workspace folder. Enter the following to create a workspace for the subject: **`H:\CP143`**.
- If you are opening the workspace for the first time, Eclipse will take you to the **Welcome** screen. Click the workbench icon on the right of the screen to start working:

Assignment 0A

Goal:

Type, compile and run a C-program, which will display text on the screen.
Find and fix programming errors in Eclipse.

1. a) Create a new project with the name **`Assignment0A`** and a source file **`Assignment0A.c`**. To do this follow the process described in the section headed *Creating an Eclipse Project* on page 9.

   b) Change the code created by the template to be similar to the example below. Enter the correct date, your own student number and description. You have to add the libraries and `printf` statement.

```c
/* Filename:       Assignment0A.c
 * Date:           2016/01/01
 * Name:           Doe J.J.
 * Student number: 12345678
 * ------------------------------------------------------
 * By submitting this file electronically, I declare that
 * it is my own original work, and that I have not copied
 * any part of it from another source.
 * ------------------------------------------------------
 * "Hello world!" Program in C
 * ------------------------------------------------------
 */
#include <stdlib.h>
#include <stdio.h>

int main() {
    printf("Hello World!\n"); // Print output
    return 0;             // Terminate program
}
```

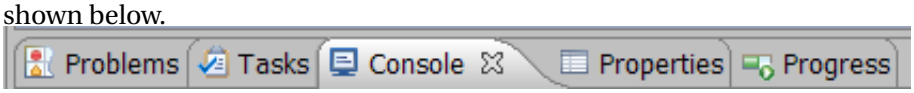
   c) Save the source file by pressing **Ctrl-S**, by selecting **File** -> **Save** from the menu bar, or by clicking on the **Save** icon in the toolbar.

   d) Compile the source file by pressing **Ctrl-B**, or by selecting **Project** -> **Build All** from the menu bar, or by clicking on the **Build** icon in the toolbar.

   Wait for the compiler to finish.

e) Run the program by pressing **Ctrl-F11**, or by selecting **Run** -> **Run** from the menu bar, or by clicking on the **Run** icon in the toolbar.

The message **`Hello World!`** should appear in the console at the bottom of the screen. Note: you may have to click on the **Console** tab to select the console, as shown below.



The program may also take a long time to execute the first time – wait for the progress bar in the bottom-right of the screen to finish.

2. a) Replace the *printf* statement with the following statement:
```
printf("Hello \nWorld!");
```

   b) Compile and run the program. The output should appear over two lines. The **`\n`** is a command that tells the output stream to insert a carriage return and to start on a new line. Remember this for Assignment B.

3. a) Remove the semicolon at the end of the **`printf`** statement. Save and compile the source file again.

   b) An error message should appear in the console, with some lines highlighted. Also, the **`return`** statement in the source code should be underlined in red, and have a red cross next to it in the left-hand margin. This indicates that your code has an **error**, and therefore can't be compiled or executed.

   c) Double-click the first highlighted line in the console. This should select the line in the source code which caused the error. In this case, the compiler only detects the error when it reaches a **`return`** statement which is not preceded by a semicolon.

   d) Put the semicolon back at the end of the **`printf`** statement. Save and compile the source file again. The error message in the console and the red underline in the code should disappear.

   e) This feature can be used to find errors in your program – look near red-underlined code to find errors, or double click an error message to jump to the error in your code, the fault will often be on the previous line, but not always.

   f) Eclipse can also give **warnings**, which will not prevent your program from compiling and running, but indicate that your code may not work correctly. These are indicated by warning messages in the console, yellow underlines in your code and yellow exclamation marks in the margin. To see an example of a warning delete the `return 0` statement.

4. Ensure that you copy the **Assignment0A** project folder to a flash drive as a backup.

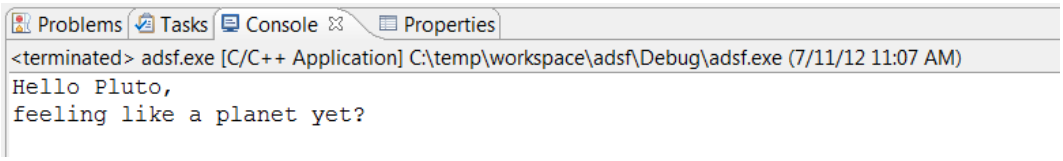
Assignment 0B

Goal:

Type, compile and run a C-program, which will display text on the screen.

1. a) Create a new project with the name **`Assignment0B`** and a source file **`Assignment0B.c`**. To do this follow the process described in the section headed *Creating an Eclipse Project* on page 9. Ensure that the correct project is selected and that the other one is closed before you start programming and compiling.

   b) Write a program that will display the following in the console:



   c) NOTE: Your output must look **EXACTLY** like the above output. That is, the output must be over two lines.

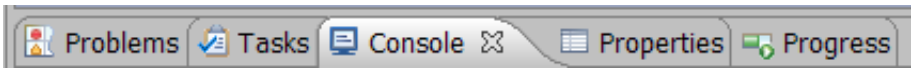
   d) Save the source file by pressing **Ctrl-S**, by selecting **File** -> **Save** from the menu bar, or by clicking on the **Save** icon in the toolbar.

   e) Compile the source file by pressing **Ctrl-B**, or by selecting **Project** -> **Build All** from the menu bar, or by clicking on the **Build** icon in the toolbar.

   Wait for the compiler to finish.

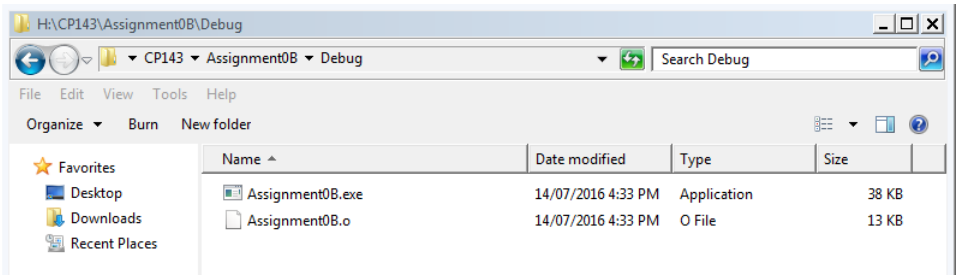
   f) Run the program by pressing **Ctrl-F11**, or by selecting **Run** -> **Run** from the menu bar, or by clicking on the **Run** icon in the toolbar.

   The correct message should appear in the console at the bottom of the screen. Note: you may have to click on the **Console** tab to select the console, as shown below.

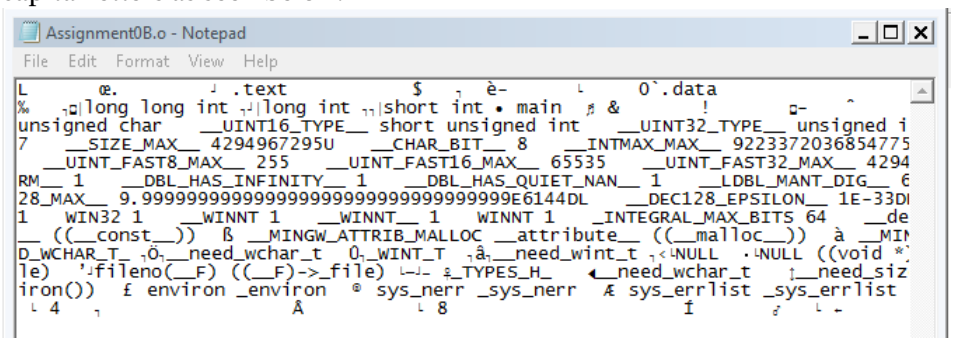


   The program may also take a long time to execute the first time – wait for the progress bar in the bottom-right of the screen to finish.

   g) After successfully running your program, use your file browser (e.g. Windows Explorer) to go to your Eclipse workspace folder, go to **Assignment0B** folder and then to the **Debug** folder. The folder content should appear as follows:

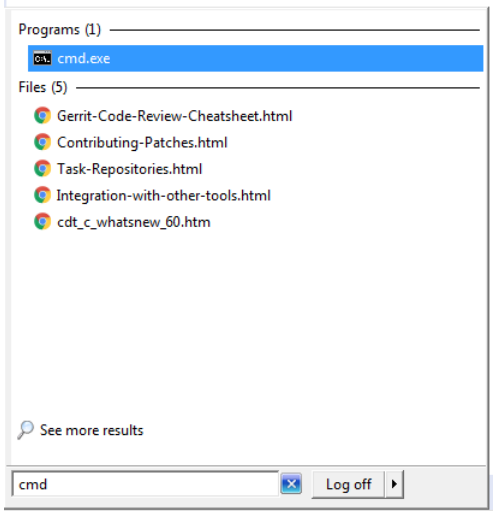


Open the object file (.o) with a text editor like Notepad (e.g. right click on **Assignment0B.o**, choose **Open**, then select **Select a program from a list of installed programs**, select **Notepad**, and click **OK**). The file should have text in capital letters as seen below:

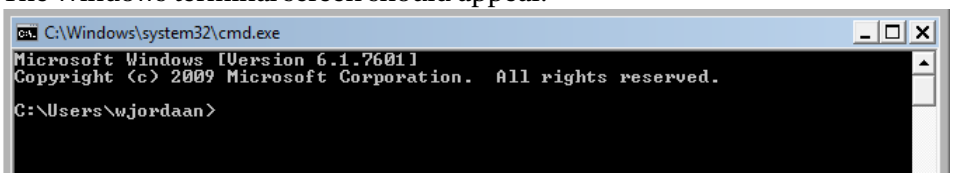


What is the function of an object file, and should it be readable to humans?

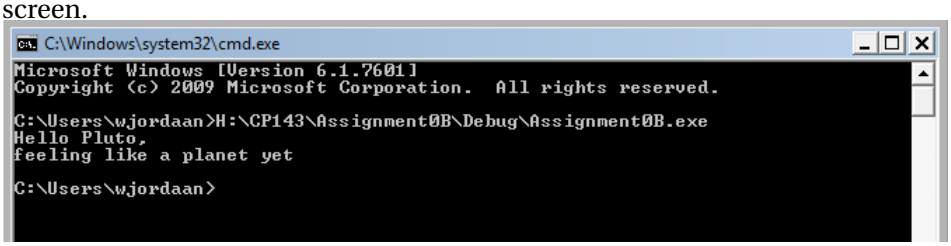
h) Keep the file browser (e.g. Windows Explorer) open. Click on the **Start** button, type `cmd` in the *Search programs and files* text box and press **Enter**.



The Windows terminal screen should appear.



Go to your file browser, click on the executable file **Assignment0B.exe**, and drag it into the Windows terminal screen. The path to this file should appear in the terminal screen. Press Enter and your program should run within the terminal screen.



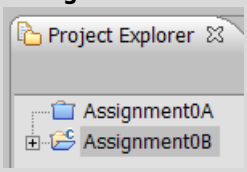
i) Congratulations you have created your first stand-alone application. This .exe is independent of Eclipse and can be used as any other program you have on your computer.

j) Ensure that you copy the **Assignment0B** project folder to a flash drive as a backup.

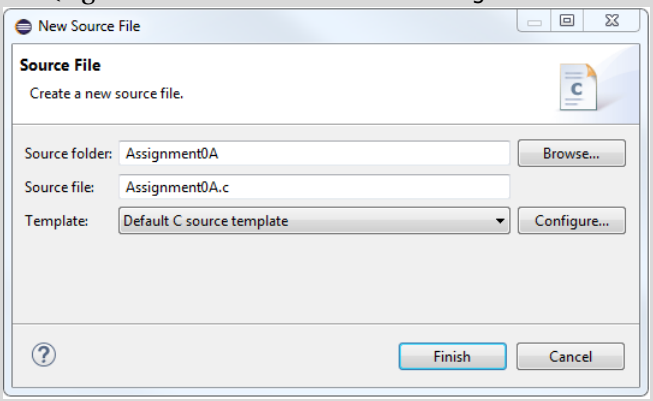
Creating an Eclipse Project

1. Create a new project.
   Note: You will do this process twice in this practical: once for each question.

   - Click **File** -> **New** -> **C Project**
   - Enter **`Assignment0A`** or **`Assignment0B`** as the project name, depending on which question you are doing.
   - Under **Project Type**, select **Executable** -> **Empty Project**.
   - Under **Toolchains**, select **MinGW GCC**. This selects the appropriate compiler for your project.
   - Click **Finish**.
   - This will create an empty C project named **`Assignment0A`** or **`Assignment0B`** as the project name, depending on which question you are doing.
   - The new project should be visible in the **Project Explorer** on the left of the screen. If there is more than one project in the workspace, make sure the one you want to work on is the only one that is open. Close all other open projects by right-clicking on each and clicking **Close**. In the example below, **`Assignment0A`** is closed, and **`Assignment0B`** is open.

   - To create a source file (.c file) within the project, right-click on the project in the **Project Explorer**, choose **New** -> **Source File**. In the **Source** field type in the project name with the addition of `.c` (e.g. in this case either `Assignment0A.c` or `Assignment0B.c`).

   - Open the newly created source file (.c) and populate the comment block to contain the file name, date, name of programmer, student number, declaration of own work and a description of what this source file does or contain (refer to the code extract in Assignment0A).

2. You can now start programming