

# Workshop #1

*Worth: 0.75% of final grade*

---

## Breakdown

- Part-1 Coding: 10%
- Part-2 Coding: 40%
- Part-2 Reflection: 50%

---

## Submission Policy

- Part-1 is due **1-day** after your scheduled LAB class by the **end of day 23:59 EST (UTC – 5)**
- Part-2 is due **5-days** after your scheduled LAB class by the **end of day 23:59 EST (UTC – 5)**
- Source (.c) and text (.txt) files that are provided with the workshop MUST be used or your work will not be accepted. Resubmission will be required attracting a **15% deduction**
- Late submissions will **NOT** be accepted
- **All work must be submitted by the matrix submitter – no exceptions**
- Reflections will not be read or graded until the coding parts are deemed acceptable and graded.
- All files you create or modify MUST contain the following declaration at the top of all documents:

\*\*\*\*\*

<assessment name example: Workshop - #1 (Part-1)>

Full Name :

Student ID#:

Email :

Section :

Authenticity Declaration:

I declare this submission is the result of my own work and has not been shared with any other student or 3rd party content provider. This submitted piece of work is entirely of my own creation.

\*\*\*\*\*

## Notes

- Due dates are in effect **even during a holiday**
- You are responsible for **backing up your work regularly**
- It is expected and assumed that for each workshop, you will plan your coding solution by using the computational thinking approach to problem solving and that **you will code your solution based on your defined pseudo code algorithm.**

## Late Submission/Incomplete Penalties

If any Part-1, Part-2, or Reflection portions are missing, the mark will be **ZERO**.

---

## Introduction

In this workshop, you will create your first C program, copy your program to the Linux matrix system for submission, interact with matrix in a remote secured shell, and submit your work using the SDDS program submitter. This routine will be repeated for all workshops and assignment(s) going forward.

## Topic(s)

- Writing a simple C program implementing standard output
- Using Visual Studio's Integrated Development Environment (IDE)
- Using PuTTY – a client application supporting the Secured Socket Shell network Protocol (SSH) to interact with a Linux server
- Using WinSCP or FileZilla client application supporting the Secured File Transfer Protocol (SFTP) to copy files to a remote Linux server

## Learning Outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities:

- to use Visual Studio to code, edit and execute a C-language program
  - to login to a remote host using SSH client application
  - to transfer source code files between a local computer and a remote host using an SFTP client application
  - To describe to your instructor what you have learned in completing this workshop
- 

## Part-1 (10%)

### Instructions

Create a C program that displays to the standard output (screen) the following:

```
Workshop 1 Part-1
```

```
=====
```

```
I'm displaying this using the 'printf' stdio  
(standard input output) library function!
```

```
Dear teacher,
```

```
    I promise I will work hard from this day onward.  
    I acknowledge that practice is extremely important,  
    so I will do all workshops, quizzes, and assignments.
```

```
Sincerely,
```

```
First_Name Last_Name  
Student ID# 123456789
```

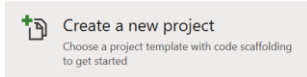
### Note

- The highlighted yellow lines identify where you should replace with YOUR personal information.
- The highlighted yellow lines are **not to be printed in yellow**.

## Visual Studio 2022

Code your program on your local computer:

- Start **Visual Studio 2022** (free Community version)
- Select **“Create a new project”** from the splash screen:

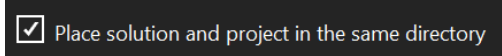


- Type **“Empty Project”** in the new project filter:



- Select the C++ template with Windows and Console on the lower line:



- Click the **NEXT** button
- Enter **“Workshop1”** as the Project Name
- Set the Location where you want to save the project (use the button with the ellipsis “...” to specify a different path from the default)
  - **Note:** If you are using a public computer, it is **strongly** advised you use a USB removable/flash drive
- Check the option to "Place solution and project in the same directory":  

- Click the button: **“Create”**
- From the top menu bar, select **Project -> Add New Item**  
(keyboard shortcut: <CTRL> + <Shift> + A)
- Select **“Code”** under the “Visual C++” tree (left panel)
- Select **“C++ File (.cpp)”** (right panel)
- Enter **“w1p1.c”** as the File Name (bottom panel)
  - **Note:** Make sure the file extension is **ALWAYS** **“.c”** (This forces Visual Studio to use the C compiler)
- Click the button **“Add”**
- Enter your source code in the main window (the contents of file **“w1p1.c”**)
- Compile your code: From the top menu bar, select **Build -> Build Solution**
  - If there are errors, fix your errors and re-attempt building  
(keyboard shortcut: <CTRL> + <Shift> + B)
- If successful (bottom left corner status message), execute your program: From the top menu bar, select **Debug -> Start without Debugging** (keyboard shortcut: <CTRL> + F5)

## Remote Linux System (matrix)

**Prepare your matrix environment for this course and for submitting your workshops:**

- Connect to the Seneca **VPN service** (Global Protect)
- Open an SSH client application like PuTTY
- Connect to the host: **matrix.senecacollege.ca**
- Login using your Seneca **userid** and **password** (be careful: case-sensitive)
- Create a directory named **btp10** where all your workshops will be stored and change into that directory:

```
[... ~]$ mkdir btp100 <ENTER>
[... ~]$ cd btp100 <ENTER>
```

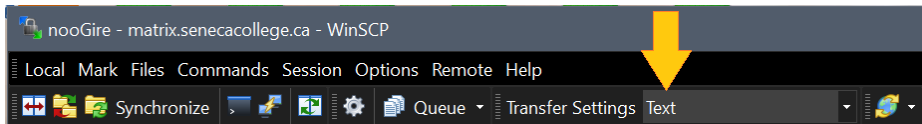
- Create another directory inside ipc144 called “w01” to store workshop #1 files:

```
[... btp100]$ mkdir w01 <ENTER>
[... btp100]$ cd w01 <ENTER>
```

- Remain connected to matrix and leave this application open

### Copying your source code file to matrix:

- Be sure you are still connected to the Seneca **VPN service** (Global Protect)
- Open an SFTP client application like WinSCP or FileZilla
- Connect to the host: **matrix.senecacollege.ca**
- Login using your Seneca **userid** and **password** (be careful: case-sensitive)
- Transfer your source file “**w1p1.c**” from your local computer to the remote matrix directory w01 that is inside directory btp100 (~/btp100/w01)
  - Avoid unexpected errors; ensure your transfer mode is set to “**Text**” (**NOT** Auto):



- After copying the file to matrix, you may close the SFTP application

### Test your code on matrix:

- Return to your SSH client application (PuTTY)
- Be sure your current directory is the w01 directory (inside btp100)
- Confirm your transferred file exists in this directory by listing the contents:

```
[... w01]$ ls <ENTER>
```

- Compile and run your solution on matrix:

```
[... w01]$ gcc -Wall w1p1.c -o w1 <ENTER>
[... w01]$ w1 <ENTER>
```

- **Make sure the output is EXACTLY as the sample output illustrates earlier in this workshop**

---

## Part-1 Submission

1. Upload (file transfer) your source file “**w1p1.c**” to your matrix account
2. Login to matrix in an SSH terminal and change directory to where you placed your workshop source code.
3. Manually compile and run your program to make sure everything works properly:

```
gcc -Wall w1p1.c -o w1 <ENTER>
```

*If there are no errors/warnings generated, execute it: **w1** <ENTER>*

4. Run the submission command below (replace **profname.proflastname** with **your professors** Seneca userid and replace **NAA** with your section):

```
~profName.proflastname/submit 100w1/NAA_p1 <ENTER>
```

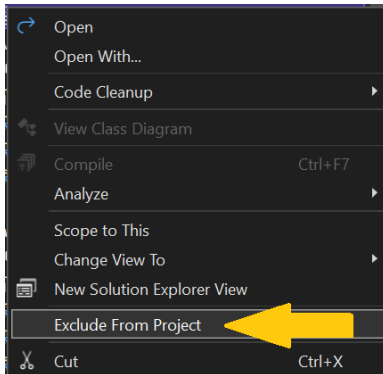
5. Follow the on-screen submission instructions
-

## Part-2 (40%)

### Instructions

Prepare your Visual Studio project environment to use a new source code file for Part-2.

- Right-click on the **w1p1.c** source code file, and from the context menu options, select "**Exclude from project**"



- From the top menu bar, select **Project -> Add New Item** (keyboard shortcut: <CTRL> + <Shift> + A)
- Select "**Code**" under the "Visual C++" tree (left panel)
- Select "**C++ File (.cpp)**" (right panel)
- Enter "**w1p2.c**" as the File Name (bottom panel) and remember to make it a **.c** file extension!
- Click the button "**Add**"
- Copy the contents from your Part-1 program ("**w1p1.c**") and paste it into the new file "**w1p2.c**"
- Modify the "**w1p2.c**" program to display the following:

### Workshop 1 Part-2

=====

Using tab specifiers...

```
<t>      Tab-1<t>Tab-2<t>Tab-3
<t>      -----<t>====<t>-----
```

```
\ this is a back-slash character!
% this is a percent sign character!
" this is a double-quote character!
```

My favourite quotes are:

- <t>** "Skill is only developed by hours and hours of work."  
**<6t>** -Usain Bolt
- <t>** "It's not about having time. It's about making time."  
**<6t>** -unknown
- <t>** "All of us do not have equal talent. But, all of us  
**<t>** have an equal opportunity to develop our talents."  
**<6t>** -A.P.J. Abdul Kalam

### Note

- The highlighted yellow **<t>** indicates you should apply **one** printf ASCII **TAB** encoded character
- The highlighted yellow **<6t>** indicates you should apply **six** printf ASCII **TAB** encoded characters

## Reflection (50%)

### Instructions

- Record your answers to the below questions in the provided “**reflect.txt**” text file
- In 3-4 sentences describe in **your own words**:
  1. Do you have any previous programming experience? If so, what language(s)? If not, what are your initial thoughts about the C language and this course?
  2. Describe your understanding of the compiling process of turning a C text file into an executable program.
  3. What do you hope to achieve in this course? Describe your goals and expectations.

---

### Academic Integrity

**It is a violation of academic policy to copy content from the course notes or any other published source (including websites, work from another student, or sharing your work with others).**

**Failure to adhere to this policy will result in the filing of a violation report to the Academic Integrity Committee.**

---

### Part-2 Submission

1. Upload your source file “**w1p2.c**” to your matrix account
2. Upload your reflection file “**reflect.txt**” to your matrix account (to the same directory)
3. Login to matrix in an SSH terminal and change directory to where you placed your workshop source code.
4. Manually compile and run your program to make sure everything works properly:

```
gcc -Wall w1p2.c -o w1 <ENTER>
```

*If there are no errors/warnings generated, execute it: **w1** <ENTER>*
5. Run the submission command below (replace **profname.prolastname** with **your professors** Seneca userid and replace **NAA** with your section):

```
~profName.prolastname/submit 100w1/NAA_p2 <ENTER>
```
6. Follow the on-screen submission instructions