

Assignment 1: the CCTS process

1. **CREATE**: Create a directory, say `ass1`, download all related files into `ass1`, then create `ass1/ass1.c` that satisfies the requirements 😊
2. **COPY**: Copy the whole directory `ass1` to your university's drive `H:`. Note: if you work in lab computers, you don't need to do this step.
3. **TEST**: login into the server `dimefox.eng.unimelb.edu.au`, then on that server, navigate to the directory `ass1`, compile and test your program.
4. **SUBMIT**: while in `dimefox`, submit your `ass1.c`, and verify.

Assignment 1: the CCTS process

1. CREATE: Create a directory, say `ass1`, download all related files into `ass1`, then create `ass1/ass1.c` that satisfies the requirements ☺

In `minGW` or `Terminal` window, when you are in your `COMP20005` (or similar) directory, do:

```
mkdir ass1
```

```
cd ass1
```

then, download all needed files from `LMS` → `Assignment1` to this directory. That includes 12 files listed in point 5 of `LMS` → `Assignment1`.

Now, it's time to build `ass1.c` (you can choose other name).

Assignment 1: the CCTS process

2. COPY: Copy the whole directory ass1 to your university's drive **H:**. Note: if you work in lab computers, you don't need to do this step.

Skip this step if you are working on a LAB's PC. Otherwise, supposing that ass1 is your current directory. You need:

<code>cd ..</code>	make parent of ass1 the current directory
<code>scp -r ass1 XXX@dimefox.eng.unimelb.edu.au:</code>	copy whole directory to your H:

Notes:

- replace XXX by your uni's login name
- there is a colon : at the end of scp
- if you do that outside uni, you need to install VPN (instructions available in LMS)

Assignment 1: the CCTS process

3. TEST:

login into `dimefox`: from your `minGW/Terminal`, type:

```
ssh XXX@dimefox.eng.unimelb.edu.au
```

and answer as needed. You will see your prompt changed to

```
bash $
```

now, use `ls` and `cd` to navigate into your `ass1` directory and compile:

```
bash $gcc -Wall -o ass1 ass1.c
```

Then, test your program against, say, `pedestrians-melbcent-p100.tsv`:

```
bash $./ass1 < pedestrians-melbcent-p100.tsv > o1.txt
```

that will write output to `o1.txt`. You can compare that with Alistair's one by

```
bash $diff o1.txt pedestrians-melbcent-p100-out.txt
```

which lists the differences between 2 files. No output means the 2 files are exactly the same, bravo.

Assignment 1: the CCTS process

4. SUBMIT: while in `dimefox`, submit your `ass1.c`, and verify.

To submit you must be on `dimefox`, and `ass1` must be your current directory. Use the command:

```
bash $ submit comp20005 ass1 ass1.c
```

After that you can verify your submission using:

```
bash $ verify comp20005 ass1 > receipt.txt
```

It's a good idea to open `receipt.txt` with `jEdit` to see its content. You might find some useful information in `receipt.txt`.

Assignment 1: the CCTS process

Today Work: minimal requirement

Create a simple (perhaps empty, perhaps just reading the data), then try all other 3 steps. Make sure that you can submit, at least from a lab PC.

Then, incrementally **CREATE** your ass1.c, do **COPY-TEST-SUBMIT** after every major development.

Assignments: advices

- *Be active in the subject's Discussion Forum!*
- *Visit LMS → Assignment 1 frequently!*
- *Make as many submissions as you want, only the last one (before deadline) counts.*
- *To simplify, do submit at uni. If you want to submit from home, beware that it might be complicated!*
- *Read the specifications carefully.*
- *Test your program carefully, at least with all supplied data. Do the testing on dimefox.*
- *Read the marking rubric carefully and try to maximize your marks!*
- *Skim the sample solution to 2015 (in LMS.Assignment1, point 6), focusing on main.c. You can learn something from there.*
- ***START EARLY, START RIGHT NOW!***