Software Design & Architecture

LAB 01

(lect01: principles of good programming)

Today's lab

- Many topics covered in the lecture
- ==> many small exercises in this lab

Refer to lectures, reading materials and any further hints in each lab

- Whatever you don't complete today, you are *expected* to complete in your private study time as part of this unit

- Where appropriate, solutions will be made available

- Sometimes we will do group work, peer reviews etc

Version Control (15 – 45 minutes)

1. If not done so already, complete LAB 5 "Introduction to Git" from Full-Stack Web Development course
   - Snapshot/copy is available also from SDA moodle

2. Use your knowledge of 'git' to create your own github repository for this course containing files needed for lab01
   - See next slides…

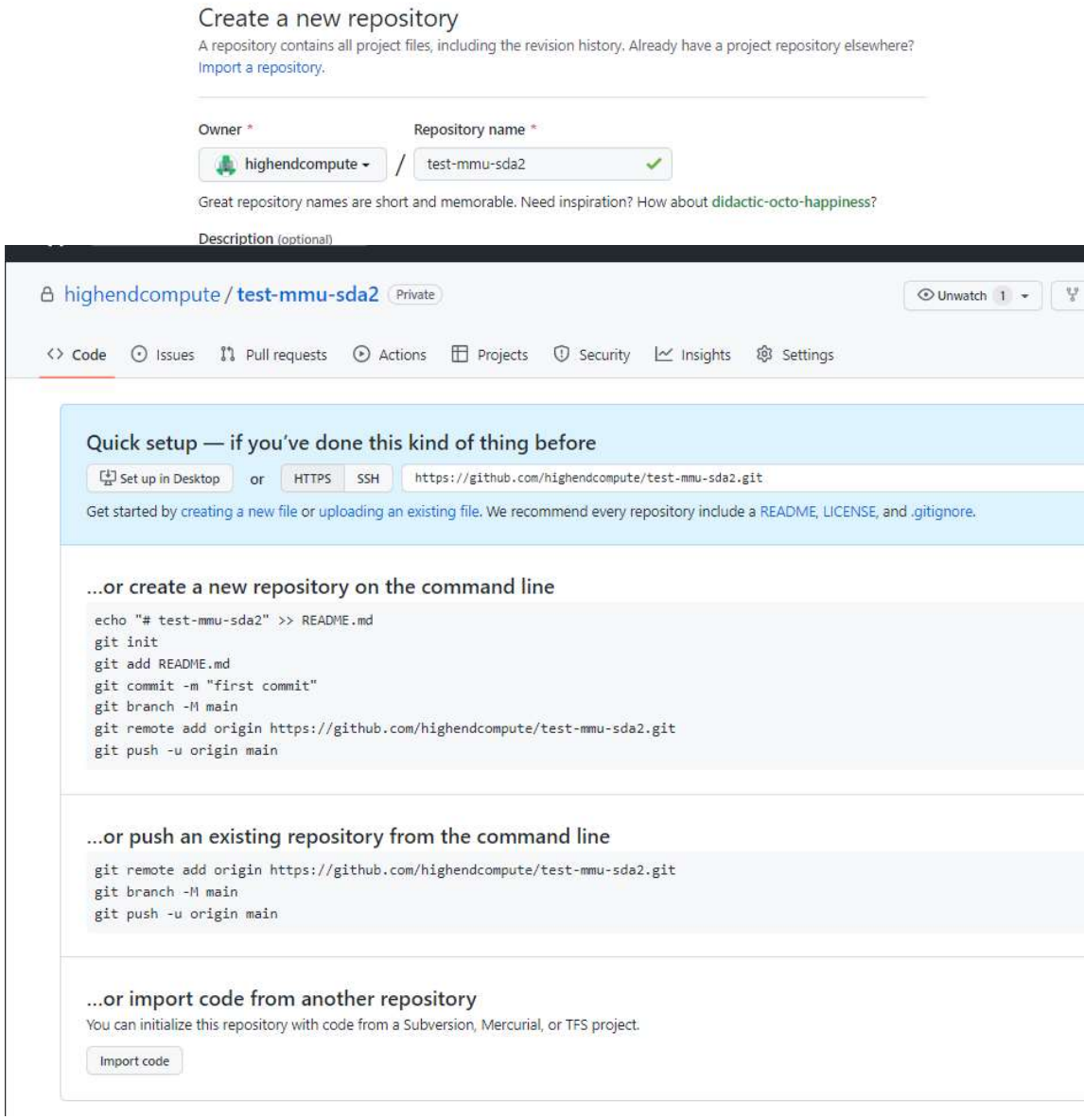
Git

1. In a new directory on your lab machine, initialise a new local git repo
2. Pull from the SDA github repo
   https://github.com/mkbane/mmu-sda.git
   You now have a copy of the files (to date)

Git

1. Create yourself a github account and create a new, *empty*, **private** repository for your work on this SDA unit

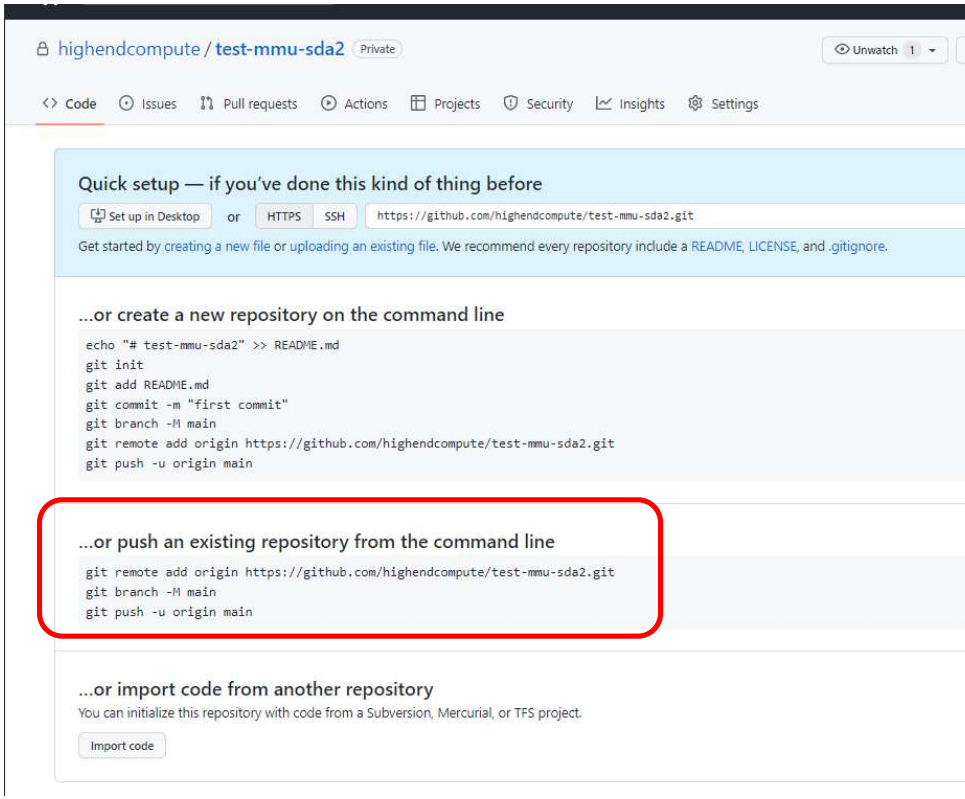
2. When you hit the green "Create repository" button, you will something like image on right
   - If you not using previously set ssh keys then **ensure you have HTTPS selected** (so you have an https URI)



Git

1. Using the URI of *your* remote repo follow the highlighted (red box) instruction *on your lab machine* (& use name of *your* github repo you just created), using your username and **Personal Access Token**

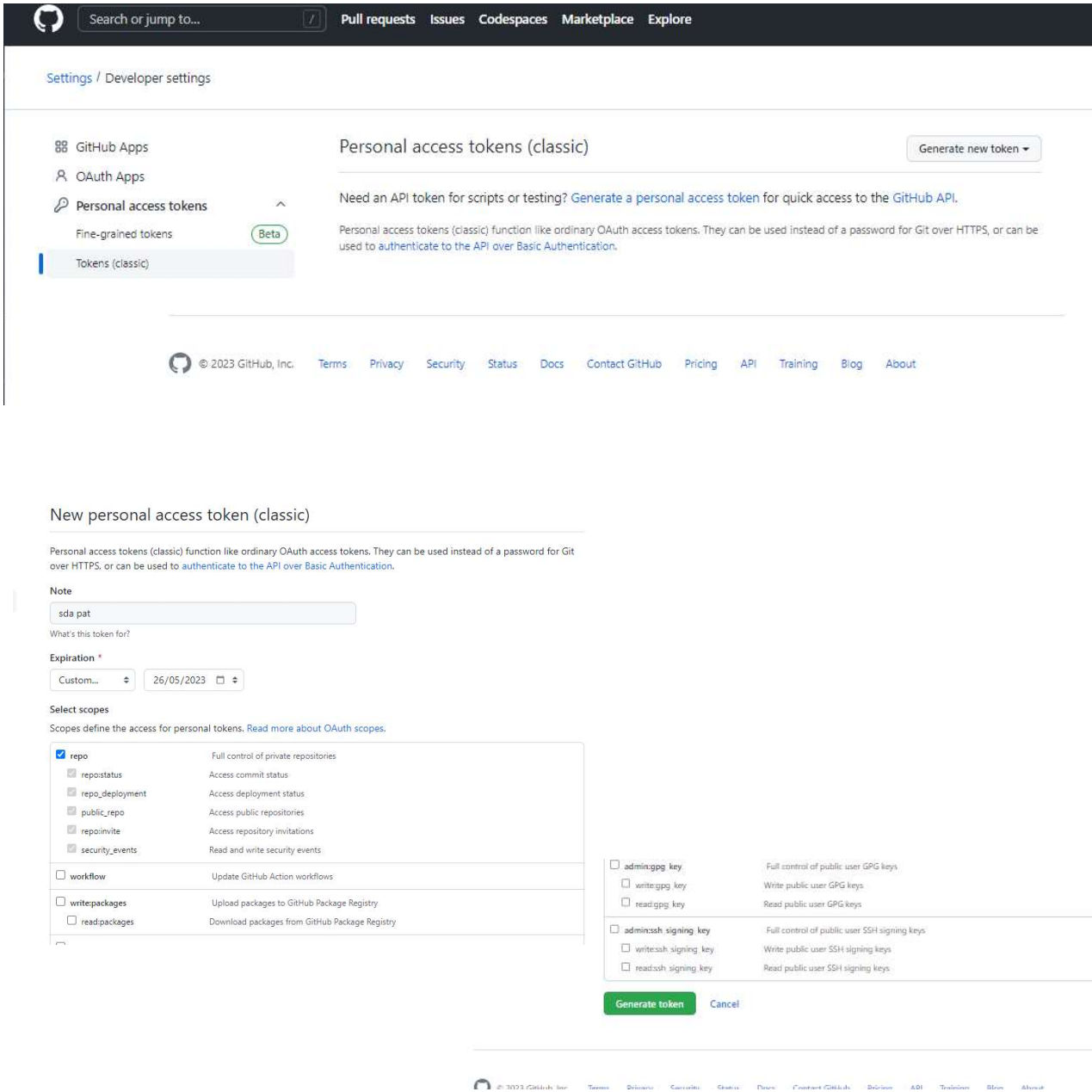
2. When successful then
   - You have now pushed a copy of the files to your own remote git repository and can work on these files and commit to your own github repository
   - Check you have the files in the remote repository!



If you do not already have a PAT see next 2 slides

Git PAT (1 of 2)

- User icon -> settings -> developer settings -> Personal access tokens -> Tokens (classic)
- Select "Generate a personal access token"
- Next screen: add Note, **select repo**, & then click green "Generate token" button at bottom

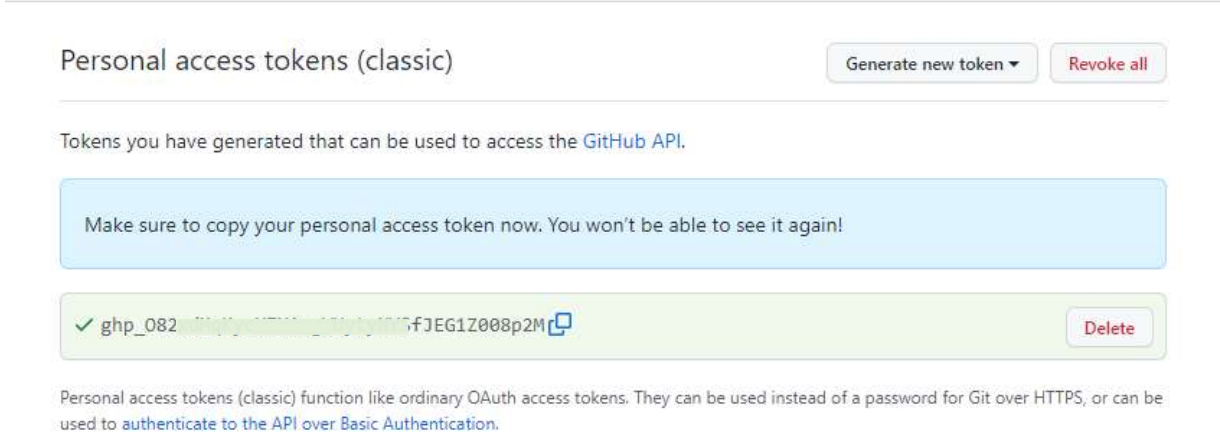


Git PAT (2 of 2)

- Next screen: **copy the PAT token somewhere safe** (but treat like a password)

- When you asked for "password" (when accessing github via CLI) then it means your PAT

- You only use actual password when logging in to web interface



Documentations

IN PAIRS:

- Examine the C code entitled "t1.c" in directory "lab01" (from the git repository)

1. Discuss what you think it is doing?
2. What could be improved?

Documentations (15 mins)

IN PAIRS:

- Examine the C code entitled "t1.c"

1. Discuss what you think it is doing?
2. What could be improved?

3. Create a stylesheet and apply to the code
4. Compare your stylesheet to another group's
    1. What differs? Discuss
    2. What is similar

Optimising Compilers (30 mins)

1. Also in directory "lab01"…
2. You should see aerosol.c and get_wtime.c files
   - Compare aerosol.c to your rewritten t1.c
3. There is a gcc compiler available on lab machines
4. Compile with default optimisation, naming executable default.exe:
   gcc aerosol.c get_wtime.c -o default.exe
5. We can then run this simulation. Let's say 25K aerosol particles for 5 timesteps, check the output and see how long it takes

See next slide on how to tell simulation how many particles etc

Optimising Compilers

- Example output

```
C:\Users\55142816\OneDrive - MMU\@teaching\L5 sw design & arch\labs\lab01>gcc aerosol.c get_wtime.c -o default.exe

C:\Users\55142816\OneDrive - MMU\@teaching\L5 sw design & arch\labs\lab01>default.exe 25000 5
Initializing for 25000 particles in x,y,z space...  (malloc-ed)    INIT COMPLETE
Time 0. System energy=899944
Now to integrate for 5 timesteps
At end of timestep 1 with temp 300.000000 the system energy=893429 and total aerosol mass=18851.7
At end of timestep 2 with temp 299.997000 the system energy=889388 and total aerosol mass=19564.7
At end of timestep 3 with temp 299.994000 the system energy=886955 and total aerosol mass=20257.4
At end of timestep 4 with temp 299.991000 the system energy=885608 and total aerosol mass=20930.2
At end of timestep 5 with temp 299.988000 the system energy=885019 and total aerosol mass=21584
Time to init+solve 25000 molecules for 5 timesteps is 86.7572 seconds
Centre of mass = (0.072349,-0.294246,50.0574)

C:\Users\55142816\OneDrive - MMU\@teaching\L5 sw design & arch\labs\lab01>
```

- NB your time may vary if there any CPU/GPU consuming processes also running on the machine

Optimising Compilers

- Referring back to lecture notes if needed, now compile various levels of optimisation.
- Which do you expect to run the fastest?
- What do you also need to check?

UML
(Unified Modelling Language)

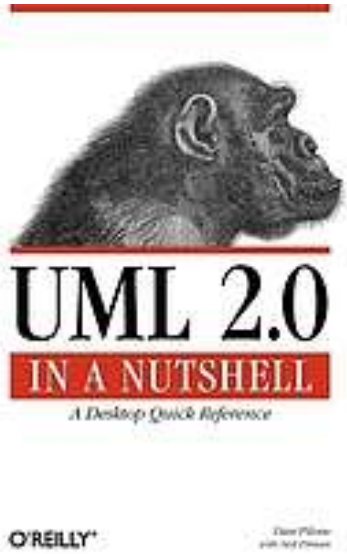
30 MINUTES

UML (30 – 90 mins)

- Unified Modelling Language
- We do not cover in this unit how to do UML, but will be used a lot during rest of unit
- NOW is good time to revise your knowledge

UML training

- If you unsure how to do a Class Diagram or the difference between, say, Sequence Diagram and a Use Case diagram then spend 30 minutes reading:
  - https://w3cschoool.com/tutorial/uml-tutorial
  - UML 2.0 in a Nutshell (Pitman)
    - Available as eBook from MMU library
    - See course unit reading list
  - UML 2 for Dummies (Schardt)
    - (requested that MMU library stocks this)



Aerosol Code as OO (30 mins)

- Consider the aerosol code from the "Optimising Compiler" exercise above
  - Currently written in C (without OO)
  - Fairly easy to understand the flow

- Draft (i.e. design!) how you would implement this in an OO language (Java) by constructing the relevant UML diagram/s.
  - What are the objects?

- Pair up with a neighbour and compare your designs
  - What differs? Discuss
  - What is in common?

Linux (30 mins – 1 hour)

- We will use the Linux OS later in the course
- UNIX & Linux are good skills to have for research & for industry

- If you have not used Linux before, reboot your lab machine and at the BIOS prompt select "Debian" option. Your username is ad*NNNNNNNN* where *NNNNNNNN* is your MMU ID. Password is your MMU password.

- Read and try out the commands in these references
  - http://www.ee.surrey.ac.uk/Teaching/Unix/
  - https://ubuntu.com/tutorials/command-line-for-beginners

Linux (advanced)

- Open a Terminal window and
  1. follow the 'git' exercise from above
  2. Follow the 'optimising compiler' exercise from above (15 mins)