Outlab 0: Orientation

Welcome to the Autumn 2021 offering of CS 251- Software Systems Lab! This course is intended to be your first step in becoming a handy all-round coder; an Industry 101 course, if you will. We will learn the basics of a wide range of tools - right from shell scripting and collaborating on git, to being able to build a project you can be proud of (likely a web application, but there are other possibilities!) Most importantly, this course, if done sincerely, will teach you how you can efficiently look stuff up, teach yourself, and *get the job done*.

This is a rather long orientation document, and it is worth your time to read it. Here's a summary of what you need to do this week:

- Join the class fora on MS-Teams and Piazza
- Follow the instructions in this document and set Docker up
- Find a partner to do your assignments with, and inform us through the Google form
- Read the honour code, and remind yourself that we trust you :)

We will use **MS-Teams** for live discussion sessions every Wednesday at 2 pm. These will be recorded, and we will not enforce attendance. The purpose of these meetings is for you to get your queries regarding the assignments cleared then and there. Join the team with the code **j26coo9**

We plan to use **Piazza** for asynchronous discussions: if you require any clarifications regarding the assignment, or answers for your conceptual doubts, ask here, so the entire class can benefit from the discussion. Join the forum at https://piazza.com/iit_bombay/fall2021/cs251.

Send an email/Teams message to the Prof or TAs only if you absolutely need to discuss some issue privately.

We will post assignments and expect you to turn them in on **Moodle.** We will also notify you via Piazza. The grading for a lot of our lab assignments will be automated, and although we will use fairly standard software, it's best to ensure uniformity in what versions are used and what packages or libraries you are allowed to import. This way, unexpected inconsistent behaviour that can cause the autograder to make mistakes is ruled out.

Docker

The way we propose to enforce uniformity is through **Docker**. A Docker **image** is like a snapshot of a system at some point in time. An image, once created, is immutable - *you cannot modify a Docker image*.

A Docker image can be instantiated into **Docker containers**. You can instantiate the same image with multiple containers, they'll all be independent of each other. When you run a Docker container on your machine, you are actually "simulating" the system that the image captured. So, given an image, regardless of what native machine instantiates the container, when the container is run, the results will behave *identically.*

This may sound familiar to those of you who use VMs (Virtual Machines). Docker has the same goal of virtualisation, and has the advantage of being a more lightweight solution, however a docker container is not a Virtual Machine, it behaves as one by containerizing the processes related to that system. As opposed to a virtual machine a docker container uses the host's kernel (read more if you're interested in making sense of the jargon).

The system in our Docker image is a machine that runs Ubuntu 20.04, and has a bunch of software you'll need for most of your labs installed on it. Feel free to write and test code on whatever IDE/workflow on your native machine, independent of Docker.

However, when we declare that a lab is autograded (assume autograded unless otherwise specified), **it is your duty to ensure that whatever code you are turning in compiles and runs correctly in the Docker container that is instantiated by the Docker image we provided!**

In particular, if you're using a container to test a submission, *do not install any additional packages on that, except if explicitly instructed.* This answers a lot of questions of the form, "Can we use … ?" If we haven't mentioned anything forbidding you from using it, and it works on the Docker container instantiated from our supplied Docker image, yes, you can.

LaTeX is going to be manually graded; you can do it online on Overleaf. If you don't have that kind of network bandwidth, the Docker image has you covered, you will be able to do the assignment locally. More details at the time of the assignment.

Docker: installation and usage

If, for some reason, our instructions just don't work for you, please fill this form. Consult your friends and the TAs first.

Install Docker on your machine. Instructions for Mac and Windows are linked, check the "Docker Engine" tab for Linux (the first step under "Install Docker Engine" should suffice). If you have doubts, you could refer to these instructions that the sysads compiled last year, or alternatively, post your query on Piazza, where your friends or the TAs can help. Now, download the Docker Image for CS 251, Autumn 2021, here. This is a file called `cs251aut2021image.tar.gz`, navigate to the directory where you downloaded it, and run (you may not necessarily need `sudo`)

```
$ sudo docker load < cs251aut2021image.tar.gz    # may take some time
```

The $ sign just signifies that you're on the command line prompt. If you're on Windows, just for this command, use the good old cmd.exe

See this thread if you have trouble starting your daemon. At this point, you'll see `cs251aut2021` listed if you run the command

```
$ docker images
```

You may (Linux) or may not (MacOS) have to run the next command.

```
$ xhost +
```

You will then run

```
$ sudo docker run --name myCS251container -it --net=host -v
"$HOME:/host" -e DISPLAY="$DISPLAY"
--volume="$HOME/.Xauthority:/root/.Xauthority:rw"
--cap-add=SYS_PTRACE --security-opt seccomp=unconfined
cs251aut2021
```

For Mac:

If you have trouble with the above command, try

```
$ sudo docker run --name myCS251container -it --net=host -v
"$HOME:/host" --cap-add=SYS_PTRACE --security-opt
seccomp=unconfined cs251aut2021
```

And verify that the files you create natively are accessible via docker, and vice versa.

For Windows:
Note: You need to run this command in your windows powershell and NOT INSIDE WSL

Only mounting c Drive --

```
$ docker run --name myCS251container -it --net=host -v
"C:/:/host" --cap-add=SYS_PTRACE --security-opt
seccomp=unconfined cs251aut2021
```

Mount multiple volumes --

```
$ docker run --name myCS251container -it --net=host -v
"C:/:/host/c" -v "D:/:/host/d" --cap-add=SYS_PTRACE
--security-opt seccomp=unconfined cs251aut2021
```

This will add your c and d drives to /host/c and /host/d inside your newly created container

This creates and starts a container called `myCS251container` (you can use any name that you find convenient instead). Remember, the container is an instance of a machine running Ubuntu 20.04, with some pre-installed software. While starting it up, part of the command mounts our own native filesystem on that of the virtualized machine, and gives permission to read and write stuff. The other part mainly tweaks some security options so that debugging tools like gdb work without hassle. After running this, you will be in the shell of the *Docker container*, and you'll be root. The command

```
# cd /host
```

should put you in your regular home directory (i.e. that of your native machine), and you can then issue the usual shell commands to compile programs and run scripts and executables within the Docker container. When you're done:

```
# exit
```

```
$ sudo docker stop myCS251container
```

Now that your container is created, issue the following commands when you want to use it again.

```
$ sudo docker start myCS251container
```

```
$ sudo docker exec -it myCS251container /bin/bash
```

Other useful commands:

```
$ sudo docker container ps -a  # will list all stopped/running containers
```

```
$ sudo docker container rm myCS251container
```
 # will delete the container named myCS251container in case you mess up and want to set up the container again

Pairs for Outlabs

Evaluation logistics? Woh ham kar lete hai. Aap jaake Dream 11 pe Team banao*. This course has a bunch of take-home assignments we call “Outlabs”, and a Course Project. The Outlabs are to be done in pairs; we may allow bigger teams for the project. We will let you know in due course. For now, fill the form in the link to let us know who you’re pairing up with to do Outlabs.

We will finalise the credit distribution for Outlabs and Project and let you know soon.

Honour Code

*The Dream 11 reference was for ice-breaking only. We neither endorse gambling, nor attempting outlabs in groups of eleven.

We understand that this course can get overwhelming at times, and it could be difficult to cope with the academic load, moreso in the midst of a pandemic. However, we believe in your strength, we believe that you will not compromise on academic integrity. We believe that you value learning with a clear conscience.

You can discuss the broad outline of your approach with your classmates. **However, for the exact details of the code, you can only consult your partner**. You can, of course , refer to any online tutorial or StackOverflow thread for help. However, you must acknowledge any such reference with the **precise link** in a file called `references.txt` for every submission.

We will be running standard code similarity detection software against your submissions. In case there’s an abnormally high score, we will first look at your `references.txt`. If the similarity can be attributed to the links pointed to, and it is reasonably clear that the entire class could potentially access your common source, and this source is not a rogue trying to announce assignment solutions, then you have nothing to worry about.

However, if that isn’t the case, we will investigate, and it could possibly get escalated to DADAC as a plagiarism case. We will consider you innocent until proven guilty, and you will get a fair chance to explain your actions.

Sharing assignment code unsolicited is as morally wrong as asking for assignment code. Coercing someone for code is not cool, real friends don’t do that.

Addendum: Docker Instructions for anyone using Arch Linux

```
$ systemctl start docker

$ sudo docker load < cs251aut2021image.tar.gz

$ sudo docker images

$ sudo docker run --name myCS251container -it --net=host -v
"$HOME:/host" --mount
type=bind,source="$HOME/.Xauthority",target="/root\.Xauthority
" --cap-add=SYS_PTRACE --security-opt seccomp=unconfined
cs251aut2021

# cd /host
```