

ECE/CS 8381 Introduction to IBM Quantum Computer Programming Lab 1

Name: Bingying Liang
ID: 48999397

Sep 27 2023

1.0 Overall Description of the Assignment

This homework assignment is designed to acquaint you with quantum computer programming using the IBM quantum computer simulator (`aer`) and an actual IBM quantum computer (`ibmq-provider`). First, you will run a simple quantum program that demonstrates superposition using the IBM simulator and again on an actual IBM quantum computer. Second, you will write, execute, basic quantum rotation operators, known as the Pauli matrices.

Also, note that there are questions embedded within the Jupyter notebook. You should answer these questions by entering your text in the cells. Equations can be entered using LaTeX format. You will turn in your homework assignment by populating all cells with source code (where required) and populating the markdown cells with answers to the questions. You will export your notebook to .html and email it to er-henderson@smu.edu, hendersonj@smu.edu, and mitch@smu.edu by the due date. Be sure that your saved notebook has all of the output from each code cell present for full credit. If you are unfamiliar with Jupyter notebooks and Python, there are many web-based resources that will quickly get you up to speed. You DO NOT have to be an expert Python programmer to do this homework.

You will need to install or setup the following. Detailed instructions for each of these steps is provided below. Some of you may already have Anaconda/Jupyter installed. IMPORTANT: The IBM Q simulators and backends are only compatible with Python 3.9 (or earlier) – so you MAY need to install a new conda environment and the Python 3.9 package if you are running a later version. Details are provided below.

- 1) Create an account on the IBM quantum experience website

The screenshot shows the IBM Quantum Platform dashboard. At the top, there's a purple header bar with the title "IBM Quantum Platform". Below it is a dark blue header bar with the user name "Bingying Liang" and navigation links for "Dashboard", "Compute Resources", and "Jobs". A search bar and a three-dot menu icon are also present.

The main content area has a dark background. On the left, there's a section titled "Recent jobs" with a message: "You don't have any jobs yet! Create a program to run on our quantum computers by following the instructions below." Below this is a "Get started" button.

In the center, there are three main sections:

- Your systems**: Shows 8 systems and 5 simulators.
- Documentation**: Includes "Qiskit Runtime", "Introduction to primitives", "Dynamic Circuits", and "Fundamentals of quantum algorithms".
- Learning**: Includes "IBM Quantum Composer" and "IBM Quantum Lab".

On the right side, there's a sidebar with the user's name "Bingying Liang" and a dropdown menu with options: "Manage account", "Provide feedback", "Support", and "Sign out". Below the sidebar, there's a "What's new" section with several items:

- Product update: Updates to the Composer and Sampler jobs (10 days ago)
- Product update: Updates in job estimation and forecasting queue wait times (10 days ago)
- Service alert: Upcoming updates to Open Plan access and terms (23 days ago)
- Product update: IBM Quantum has a new navigation and application updates! (About 1 month ago)
- Quantum news: IBM Quantum Awards: Open Science Prize (8 months ago)
- Product update: Updates to job executions - optimizing classical computation

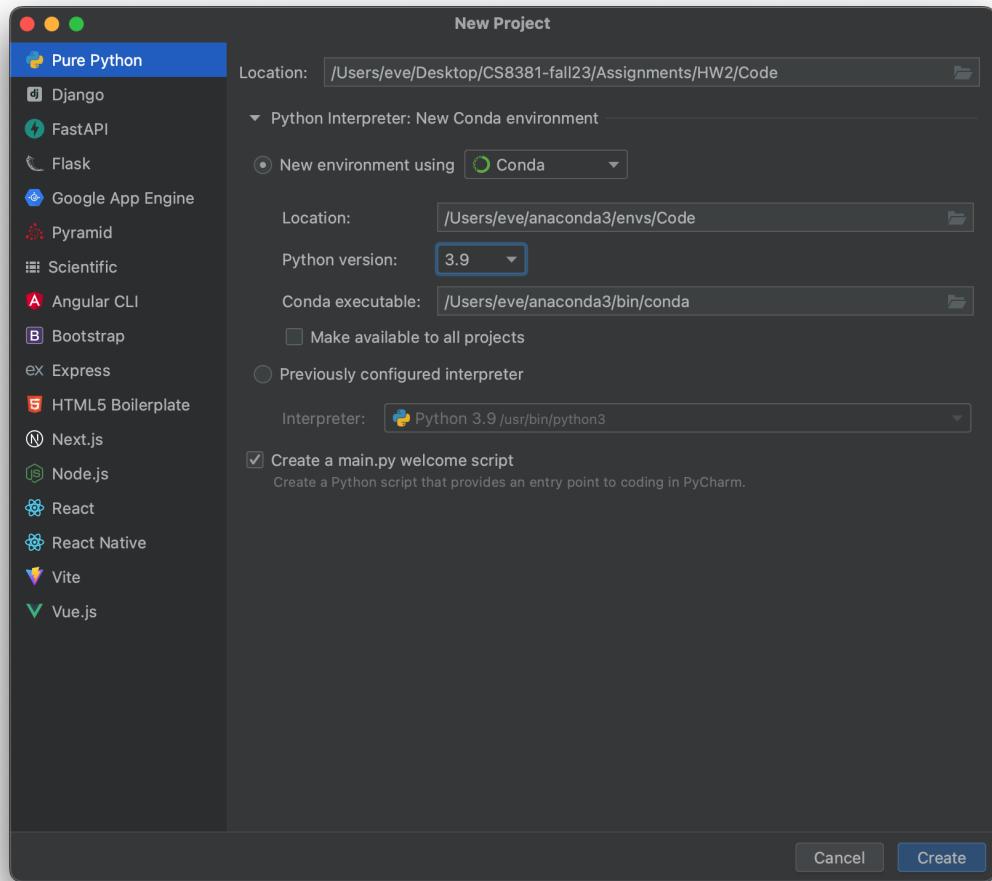
2) Install and create an environment in Anaconda

The screenshot shows the Anaconda Navigator interface. At the top, there's a header bar with the title "Anaconda NAVIGATOR", a "Upgrade Now" button, and a "Connect" button.

The main area is a grid of tool icons and descriptions:

Tool	Description	Action Buttons
DataSpell	An IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.	Install, Launch
Anaconda Notebooks	Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.	Launch
JupyterLab	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Launch
Jupyter Notebook	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
PyCharm Professional	A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.	Launch
Qt Console	PyQt GUI that supports inline figures, proper multi-line editing with syntax highlighting, graphical calltips, and more.	Launch
Spyder	Scientific Python development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.	Launch
VS Code	Streamlined code editor with support for development operations like debugging, task running and version control.	Launch
Datalore		
IBM Watson Studio Cloud		
Oracle Data Science Service		
Glueviz		

On the left side, there's a sidebar with navigation links: "Home", "Environments", "Learning", and "Community". There's also a "Anaconda Notebooks" section with a "Learn More" button and social media links for Twitter, YouTube, and LinkedIn.



```
# This is a sample Python script.

# Press ⌘R to execute it or replace it with your code.
# Press Double ⇧ to search everywhere for classes, files, tool windows, actions, and settings.

usage new"
def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}') # Press ⌘F8 to toggle the breakpoint.

# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

Terminal: Local

```
(Code) ➜ Code git:(main) ✘ conda env list
# conda environments:
#
base          /Users/eve/anaconda3
Code          * /Users/eve/anaconda3/envs/Code

(Code) ➜ Code git:(main) ✘ import sys
Version: ImageMagick 7.1.1-13 Q16-HDR1 aarch64 21276 https://imagemagick.org
Copyright: (C) 1999 ImageMagick Studio LLC
License: https://imagemagick.org/script/license.php
Features: Cipher DPC HDRI Modules OpenMP(5.0)
Delegates (built-in): bzlib fontconfig freetype gslib heic jng jp2 jpeg jxl lcms lqr ltdl lzma openexr png ps raw tiff webp
Compiler: gcc (4.2)
```

IDE project settings can be added to Git // View Files // Always Add // Don't Ask Again

- 3) Install the Python (must be 3.9, not the latest version), qiskit, qiskit[visualization], and matplotlib libraries,

```
eve — eve@Eves-Air — ~ -zsh — 80x24
Last login: Thu Sep 21 22:04:20 on ttys000
[(base) ~ conda env list
# conda environments:
#
base          * /Users/eve/anaconda3
Code          /Users/eve/anaconda3/envs/Code

[(base) ~ pip install qiskit 'qiskit[visualization]' matplotlib
Collecting qiskit
  Obtaining dependency information for qiskit from https://files.pythonhosted.org/packages/36/43/be9bc635ff62d27a18de52f87aaaf7eb7b0bc80f1241684c4673907d5757/qiskit-0.44.1-py3-none-any.whl.metadata
    Downloading qiskit-0.44.1-py3-none-any.whl.metadata (8.2 kB)
Requirement already satisfied: matplotlib in ./anaconda3/lib/python3.11/site-packages (3.7.1)
Collecting qiskit-terra==0.25.1 (from qiskit)
  Obtaining dependency information for qiskit-terra==0.25.1 from https://files.pythonhosted.org/packages/28/38/137a26c022fb24234aafe08ab764c0611c090ce1715bd66238b629461184/qiskit_terra-0.25.1-cp38-abi3-macosx_11_0_arm64.whl.metadata
    Downloading qiskit_terra-0.25.1-cp38-abi3-macosx_11_0_arm64.whl.metadata (9.0 kB)
Collecting rustworkx>=0.13.0 (from qiskit-terra==0.25.1->qiskit)
  Obtaining dependency information for rustworkx>=0.13.0 from https://files.pythonhosted.org/packages/38/db/1800dc4a408edaf5ae1479dd822a917f78cd8e39d09f014ab2b4
```

- 4) Install Jupyter lab(or Jupyter notebook) in Anaconda

The screenshot shows a Jupyter Notebook interface with the title "Environment Check". The notebook has three code cells:

- In [11]:** `! conda env list`

```
# conda environments:
#
base                  * /Users/eve/anaconda3
Code
```
- In [17]:** `import sys
print(sys.executable)`

```
/Users/eve/anaconda3/bin/python
```
- In [18]:** `! jupyter kernelspec list`

```
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
Available kernels:
python3      /Users/eve/anaconda3/share/jupyter/kernels/python3
```

Below the code cells, there is a section titled "Homework 2: Basic Quantum Programs" with a note about Qiskit's quantum computing software. At the bottom, there is a link to "Import libraries and simulator".

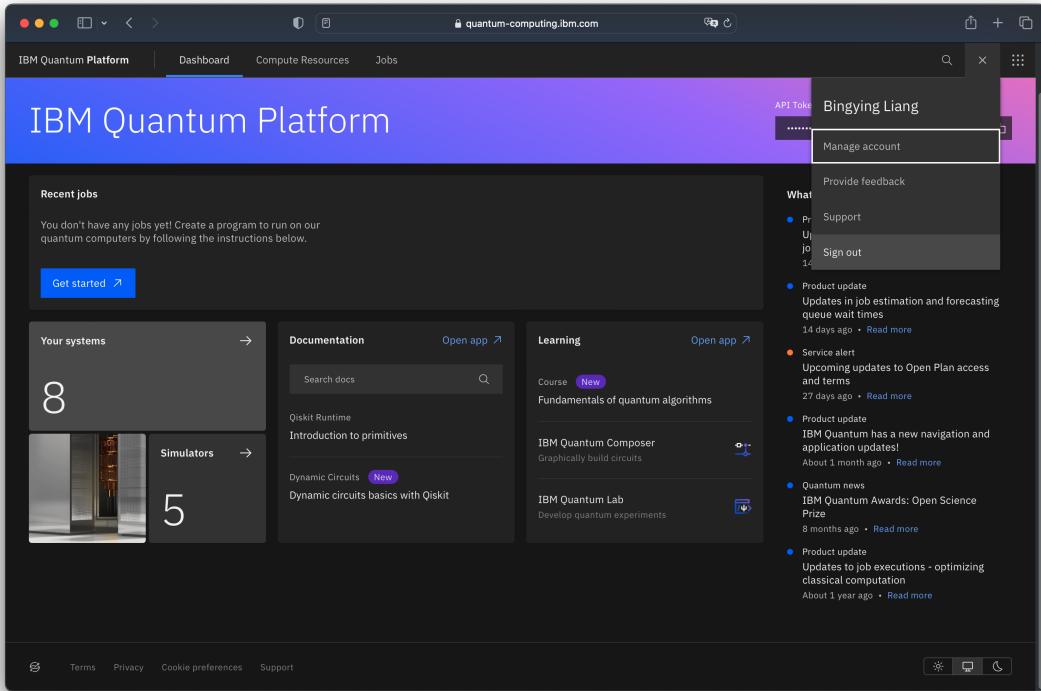
2.0 Creating an IBMQ Account

To run your program on actual quantum computing hardware, you will need to create an account that will provide you with an “API token.” The API token is required to allow you to run your program on actual hardware, otherwise the simulator (*aer*) is used.

Instructions for creating your IBM account are provided on a github page at: <https://github.com/Qiskit/qiskit-ibmq-provider>. The github page instructs you to do the following:

Create an IBM Quantum account login by visiting: <https://quantum-computing.ibm.com/login>. On the right-most side of the webpage where the login prompt appears, select “Create an IBMid account” and follow the instructions to create your account. If you already have an IBMid account, you can skip this step and just login using your IBMid credentials. After creating an account, you will be assigned an “API token.” You will need to copy the token directly into your Python code in the Jupyter notebook in section 3 entitled “Run on a real quantum computer.” A Python line is included that indicates where you should paste your IBM Q token. More details are available in the notebook.

Solution:



3.0 Installing and Configuring Anaconda with IBM Qiskit

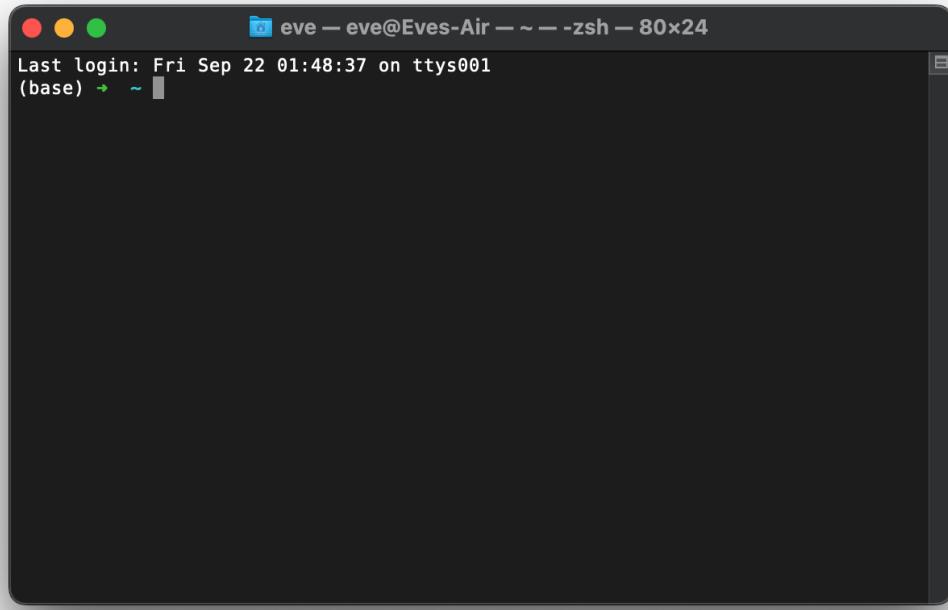
Qiskit is the name of the IBM package that we will use in this homework. It is highly recommended that you create a new environment even if you already have Anaconda installed. Furthermore, as stated above, it is **VERY IMPORTANT** that you install a previous version of Python (version 3.9 works well) as Qiskit is not compatible with the latest version of Python at the time of this writing.

- 1) Open a terminal on your machine. Enter the command `conda info` to determine if you have already installed Anaconda. If you do not have Anaconda installed, you can download it from here:

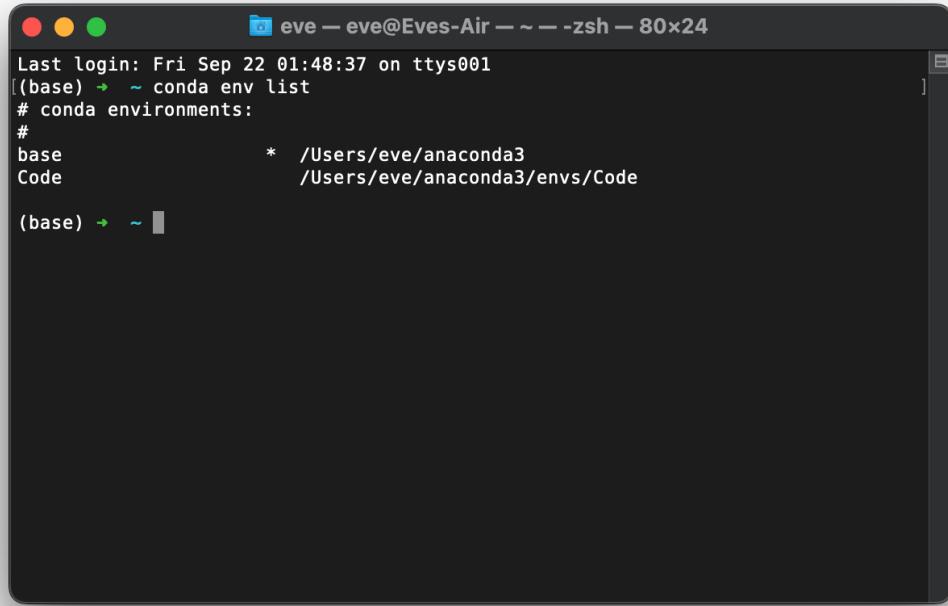
<https://conda.io/projects/conda/en/latest/user-guide/install/index.html>

- 2) Using and creating conda environments:

- i) Open a terminal on your machine.

A screenshot of a macOS terminal window titled "eve — eve@Eves-Air — ~ — zsh — 80x24". The window shows the user's last login information: "Last login: Fri Sep 22 01:48:37 on ttys001". Below this, the prompt "(base) ~" is visible. The main body of the terminal is entirely blank, showing only the dark background.

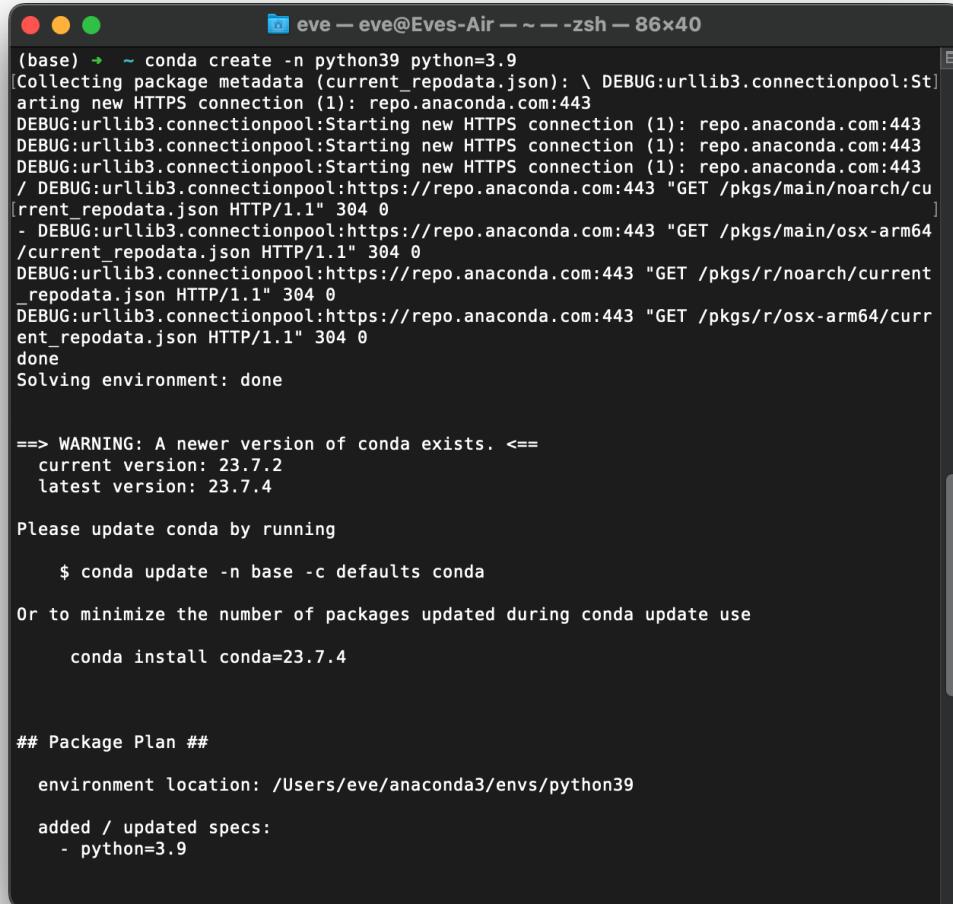
- ii) Enter the command `conda env list` to see the currently installed environments.

A screenshot of a macOS terminal window titled "eve — eve@Eves-Air — ~ — zsh — 80x24". The window shows the user's last login information: "Last login: Fri Sep 22 01:48:37 on ttys001". Below this, the user runs the command `conda env list`. The output shows two environments: "base" and "Code". The "base" environment is marked with an asterisk (*) and points to the path "/Users/eve/anaconda3". The "Code" environment points to the path "/Users/eve/anaconda3/envs/Code". The prompt "(base) ~" is visible at the bottom.

- iii) Choose a name for your new conda environment to use for this homework. Make sure it is not

a name that is already being used from the previous step. I used the name python39. Create the new environment by entering the command:

conda create {name python39 (or whichever name you prefer). This is the environment in which you will use to run your IBM quantum computer programs.

A screenshot of a macOS terminal window titled "eve — eve@Eves-Air — ~ — zsh — 86x40". The window displays the output of a "conda create" command. It shows the process of collecting package metadata, starting new HTTPS connections, and downloading packages from the Anaconda repository. A warning message indicates that a newer version of conda exists (current: 23.7.2, latest: 23.7.4), followed by instructions to update conda. The final output shows the package plan for creating a new environment named "python39", which includes the addition of the "python=3.9" spec.

```
(base) ~ conda create -n python39 python=3.9
[Collecting package metadata (current_reodata.json): \ DEBUG:urlib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
DEBUG:urlib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
DEBUG:urlib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
/ DEBUG:urlib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/noarch/cu
[rent_reodata.json HTTP/1.1" 304 0
- DEBUG:urlib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/osx-arm64/
/current_reodata.json HTTP/1.1" 304 0
DEBUG:urlib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/noarch/current_
reodata.json HTTP/1.1" 304 0
DEBUG:urlib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/osx-arm64/curr
ent_reodata.json HTTP/1.1" 304 0
done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.2
  latest version: 23.7.4

Please update conda by running

$ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

conda install conda=23.7.4

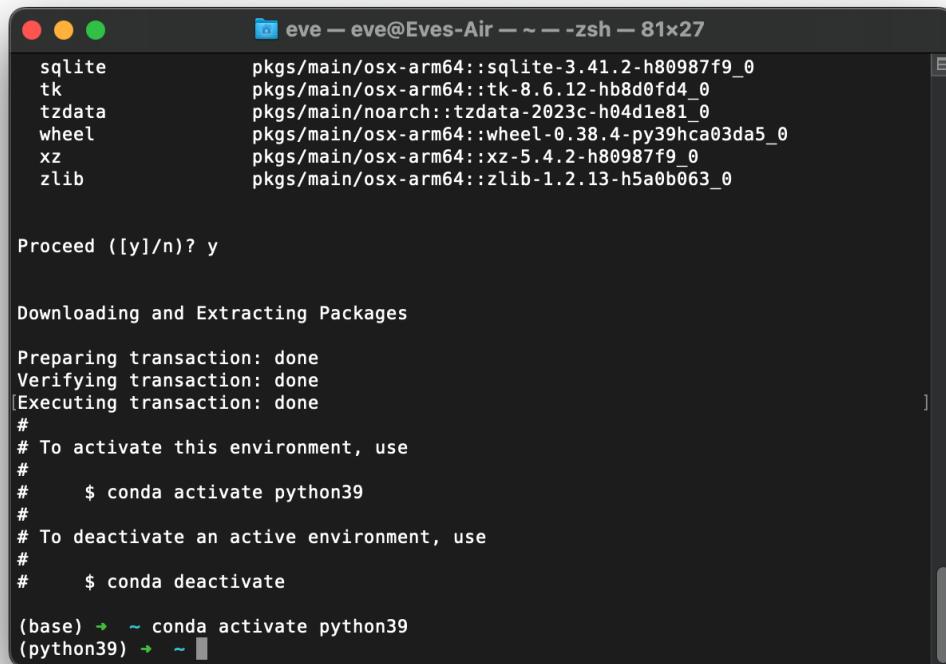
## Package Plan ##

environment location: /Users/eve/anaconda3/envs/python39

added / updated specs:
- python=3.9
```

- iv) After creating the new environment, you need to activate it. To do so, enter:

```
conda activate python39
```



```
eve — eve@Eves-Air — ~ — zsh — 81x27
sqlite          pkgs/main/osx-arm64::sqlite-3.41.2-h80987f9_0
tk              pkgs/main/osx-arm64::tk-8.6.12-hb8d0fd4_0
tzdata          pkgs/main/noarch::tzdata-2023c-h04d1e81_0
wheel           pkgs/main/osx-arm64::wheel-0.38.4-py39hca03da5_0
xz              pkgs/main/osx-arm64::xz-5.4.2-h80987f9_0
zlib           pkgs/main/osx-arm64::zlib-1.2.13-h5a0b063_0

Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate python39
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) ~ conda activate python39
(python39) ~
```

- v) When you wish to leave this environment, you can enter:

```
conda deactivate python39
```

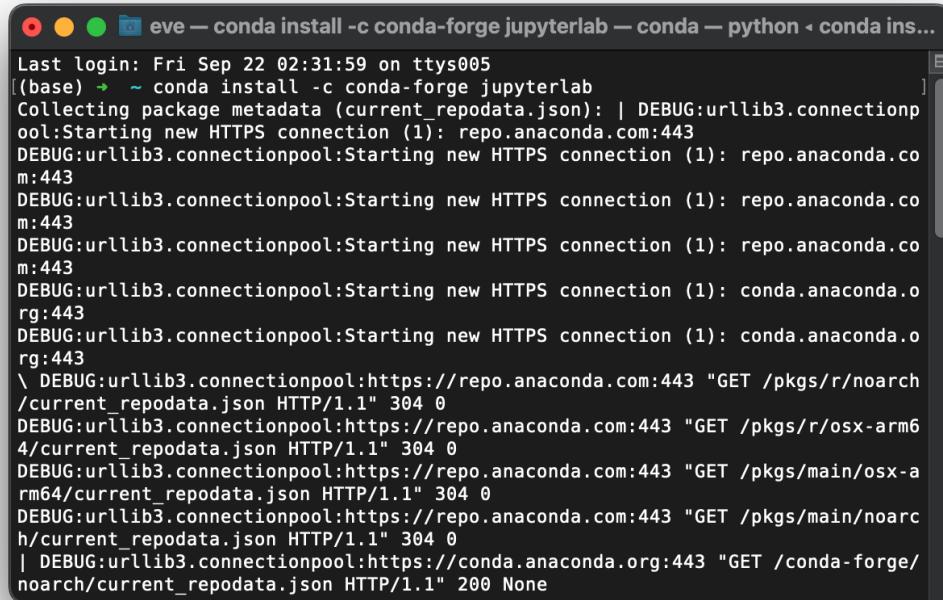


```
eve — eve@Eves-Air — ~ — zsh — 61x15
Last login: Fri Sep 22 01:59:02 on ttys002
[(base) ~ conda activate python39
[(python39) ~ conda deactivate
(base) ~
```

3) Downloading Jupyter:

- i) You need to download the `jupyter lab` package if you do not already have it installed (note that `jupyter notebook` can also be used if you prefer). `Jupyter lab` is similar to `jupyter notebook` with runnable cells, but it has additional features. `Jupyter` can be installed in the base environment since it can access kernels in other environments. Enter the following command:

```
conda -install -c conda-forge jupyterlab
```

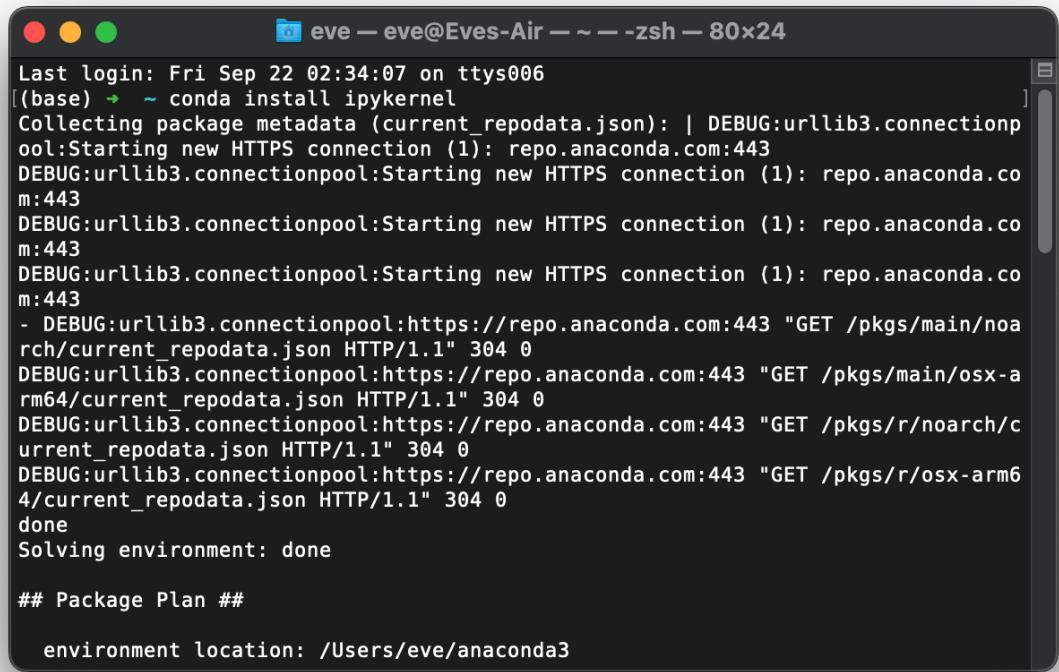


The terminal window shows the command `eve — conda install -c conda-forge jupyterlab` being run. The output includes several DEBUG messages from the urllib3 library, indicating the process of collecting package metadata and establishing HTTPS connections to the Anaconda repository at `repo.anaconda.com:443`. The final line shows the successful download of the package from `conda.anaconda.org:443`.

```
Last login: Fri Sep 22 02:31:59 on ttys005
[(base) ~] conda install -c conda-forge jupyterlab
Collecting package metadata (current_repodata.json): | DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): conda.anaconda.org:443
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): conda.anaconda.org:443
\ DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/noarch/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/osx-arm64/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/osx-arm64/current_repodata.json HTTP/1.1" 304 0
| DEBUG:urllib3.connectionpool:https://conda.anaconda.org:443 "GET /conda-forge/noarch/current_repodata.json HTTP/1.1" 200 None
```

- ii) Install the `jupyter lab` kernel by entering the command:

```
conda install ipykernel
```



```
eve — eve@Eves-Air — ~ — zsh — 80x24
Last login: Fri Sep 22 02:34:07 on ttys006
(base) ➜ ~ conda install ipykernel
Collecting package metadata (current_repodata.json): | DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
- DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/noarch/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/osx-arm64/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/noarch/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/osx-arm64/current_repodata.json HTTP/1.1" 304 0
done
Solving environment: done

## Package Plan ##

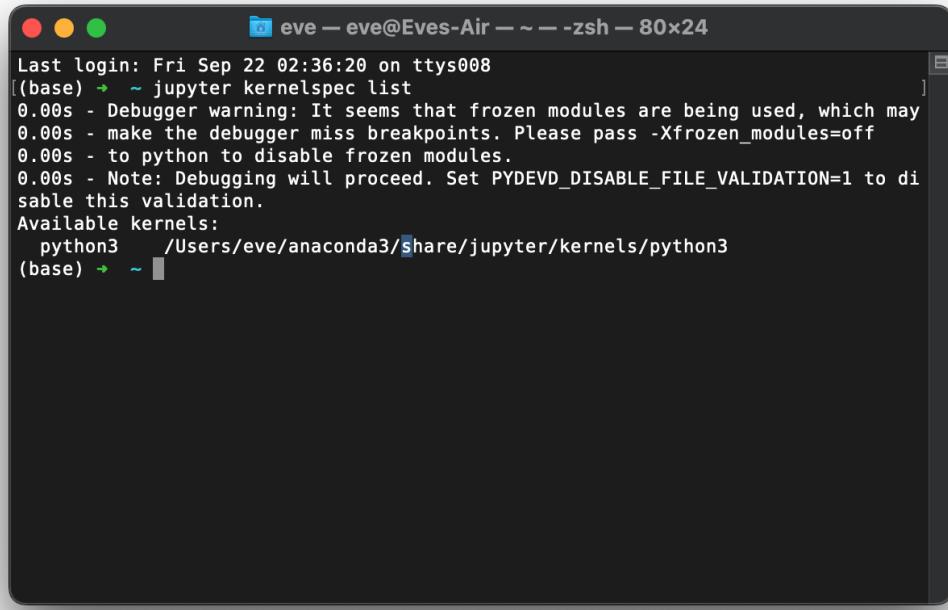
environment location: /Users/eve/anaconda3
```

- iii) Optionally, this command allows you to select the environment from jupyter lab.

```
python -m ipykernel install --user --name qc --display-name "Python (qc)"
```

We can see the available kernels for jupyter lab with command.

```
jupyter kernelspec list
```



The screenshot shows a terminal window titled 'eve — eve@Eves-Air — ~ — zsh — 80x24'. The command 'jupyter kernelspec list' was run, displaying the following output:

```
Last login: Fri Sep 22 02:36:20 on ttys008
(base) ~ jupyter kernelspec list
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to di
sable this validation.
Available kernels:
    python3    /Users/eve/anaconda3/share/jupyter/kernels/python3
(base) ~
```

4) Downloading additional (required) packages. Although `conda install` can be used and is generally faster, the `pip` utility seems to be more reliable. For this reason, I will use `pip` in the following instructions.

i) As qiskit is not compatible with the latest version of Python, it is necessary to install an earlier version. I used Python 3.9 in this homework. Enter the command:

```
conda create --name python39 python==3.9
```

(note that “python39” is the name of the environment created in step 2)

```

eve — eve@Eves-Air — ~ — zsh — 80x24
Last login: Fri Sep 22 02:42:04 on ttys008
(base) ~ conda create --name python39 python==3.9
WARNING: A conda environment already exists at '/Users/eve/anaconda3/envs/python39'
Remove existing environment (y/[n])? y

Collecting package metadata (current_repodata.json): | DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
- DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/osx-arm64/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/noarch/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/noarch/current_repodata.json HTTP/1.1" 304 0
\ DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/osx-arm64/current_repodata.json HTTP/1.1" 304 0
done
Solving environment: unsuccessful attempt using repodata from current_repodata.j

```

- ii) Optionally, if you cannot find the appropriate packages in your default channels, you can add new channels to the package search process. For example, the channel “`conda-forge`” would be added by entering:

`conda config -add channels conda-forge`

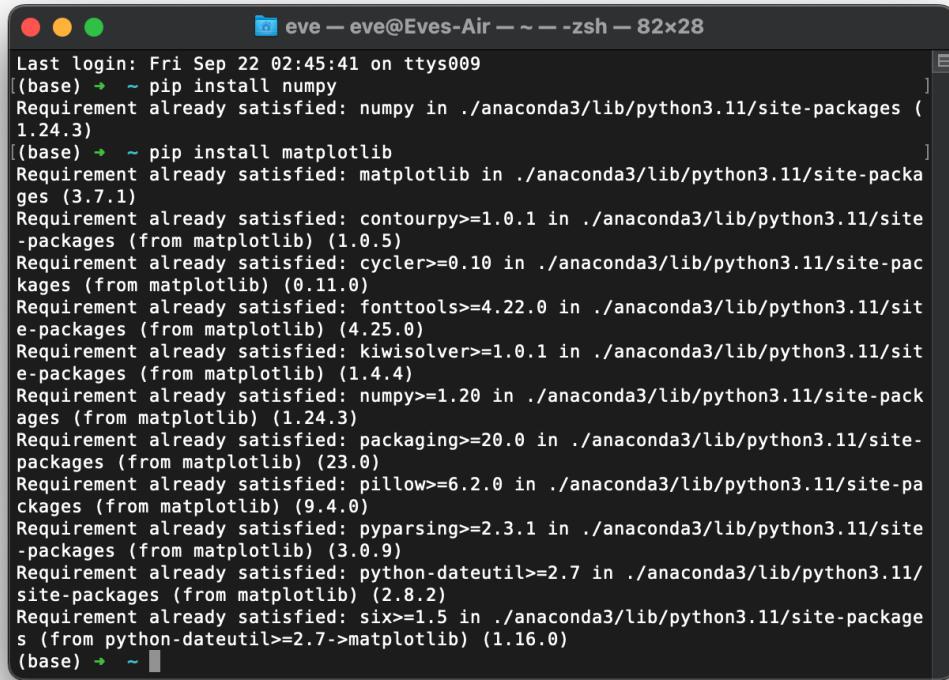
- iv) Additional libraries should be installed. You may need to explicitly install `numpy` if it was not properly installed with Python. If you need to install `numpy`, enter:

`pip install numpy`

Next, we need to install a plotting package to display qubit histograms for repeated runs/measurements.

Enter:

`pip install matplotlib`



A terminal window titled "eve — eve@Eves-Air — ~ — zsh — 82x28". The window shows the command "pip install numpy" being run, followed by a list of requirements already satisfied from the Anaconda3 site-packages. The packages listed include numpy, matplotlib, contourpy, cycler, fonttools, kiwisolver, numpy, packaging, pillow, pyparsing, python-dateutil, and six.

```
Last login: Fri Sep 22 02:45:41 on ttys009
[(base) ~] pip install numpy
Requirement already satisfied: numpy in ./anaconda3/lib/python3.11/site-packages (1.24.3)
[(base) ~] pip install matplotlib
Requirement already satisfied: matplotlib in ./anaconda3/lib/python3.11/site-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=6.2.0 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in ./anaconda3/lib/python3.11/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.11/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
(base) ~]
```

- v) Next, we need to install the required qiskit packages. Enter:

```
pip install qiskit
```

To verify a channel is present in your environment that contains a particular package, for example the qiskit package, enter:

```
conda search qiskit
```

```
Last login: Fri Sep 22 02:51:12 on ttys011
[eve] ~ - pip install qiskit
Requirement already satisfied: qiskit in ./anaconda3/lib/python3.11/site-packages (0
.44.1)
Requirement already satisfied: qiskit-terra==0.25.1 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit) (0.25.1)
Requirement already satisfied: rustworkx>=0.13.0 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (0.13.1)
Requirement already satisfied: numpy>=1.17 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (1.24.3)
Requirement already satisfied: ply>=3.10 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (3.11)
Requirement already satisfied: psutil>=5 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (5.9.0)
Requirement already satisfied: scipy>=1.5 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (1.10.1)
Requirement already satisfied: sympy>=1.3 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (1.11.1)
Requirement already satisfied: dill>=0.3 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (0.3.6)
Requirement already satisfied: python-dateutil>=2.8.0 in ./anaconda3/lib/python3.11/
site-packages (from qiskit-terra==0.25.1->qiskit) (2.8.2)
Requirement already satisfied: stevedore>=3.0.0 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (5.1.0)
Requirement already satisfied: symengine<0.10,>=0.9 in ./anaconda3/lib/python3.11/si
te-packages (from qiskit-terra==0.25.1->qiskit) (0.9.2)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.11/si
te-packages (from python-dateutil>=2.8.0->qiskit-terra==0.25.1->qiskit) (1.16.0)
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in ./anaconda3/lib/python3.11/si
te-packages (from stevedore>=3.0.0->qiskit-terra==0.25.1->qiskit) (5.11.1)
Requirement already satisfied: mpmpath>=0.19 in ./anaconda3/lib/python3.11/si
te-packages (from sympy>=1.3->qiskit-terra==0.25.1->qiskit) (1.3.0)
(base) ~
```

To support certain graphics operations, it is necessary to install an additional qiskit package. Note that users of later versions of Mac OSX will be using zsh as the default linux shell in the terminal. The command to install the qiskit visualization package for zsh uses single quotes whereas that for Windows users or other linux shells does not. For this reason, I give both versions of the installation command:

`pip install qiskit[visualization]` (for Windows or older versions of Mac OSX that do not use zsh)

`pip install 'qiskit[visualization]'` (for Mac OSX users running zsh)

Previous releases of the visualization package used the following commands, but these should not be necessary for this homework and are included here only if the above pip install command is causing issues.

`pip install qiskit-terra[visualization]` (for Windows or older versions of Mac OSX that do not use zsh)

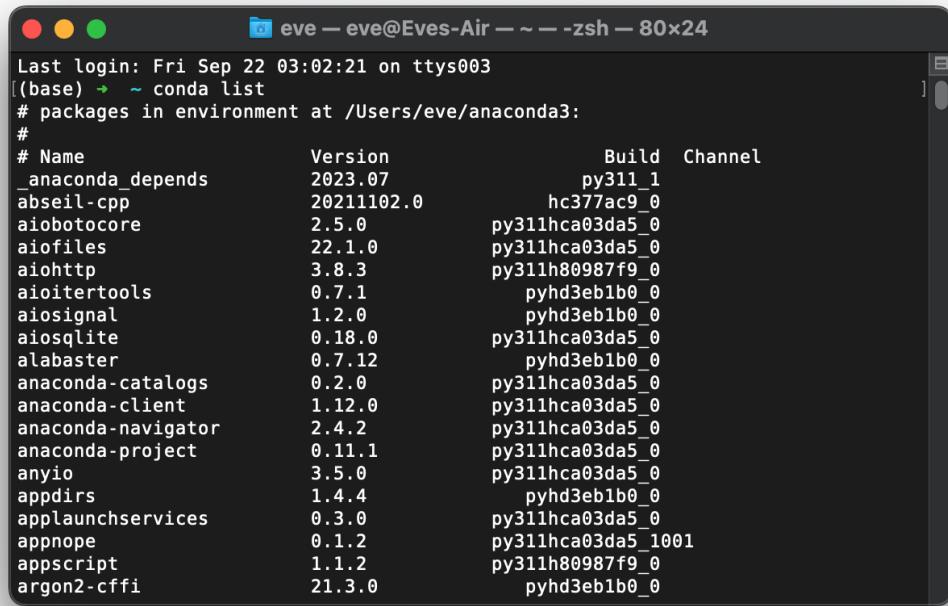
```
eve — eve@Eves-Air — ~ — zsh — 80x24
Last login: Fri Sep 22 02:55:57 on ttys001
(base) ~ pip install 'qiskit[visualization]'
Requirement already satisfied: qiskit[visualization] in ./anaconda3/lib/python3.11/site-packages (0.44.1)
Requirement already satisfied: qiskit-terra==0.25.1 in ./anaconda3/lib/python3.11/site-packages (from qiskit[visualization]) (0.25.1)
Requirement already satisfied: rustworkx>=0.13.0 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (0.13.1)
Requirement already satisfied: numpy>=1.17 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (1.24.3)
Requirement already satisfied: ply>=3.10 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (3.11)
Requirement already satisfied: psutil>=5 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (5.9.0)
Requirement already satisfied: scipy>=1.5 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (1.10.1)
Requirement already satisfied: sympy>=1.3 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (1.11.1)
Requirement already satisfied: dill>=0.3 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (0.3.6)
Requirement already satisfied: python-dateutil>=2.8.0 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (2.8.2)
Requirement already satisfied: stevedore>=3.0.0 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra==0.25.1->qiskit[visualization]) (5.1.0)
```

```
eve — eve@Eves-Air — ~ — zsh — 80x24
Last login: Fri Sep 22 03:01:00 on ttys002
(base) ~ pip install 'qiskit-terra[visualization]'
Requirement already satisfied: qiskit-terra[visualization] in ./anaconda3/lib/python3.11/site-packages (0.25.1)
Requirement already satisfied: rustworkx>=0.13.0 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (0.13.1)
Requirement already satisfied: numpy>=1.17 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (1.24.3)
Requirement already satisfied: ply>=3.10 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (3.11)
Requirement already satisfied: psutil>=5 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (5.9.0)
Requirement already satisfied: scipy>=1.5 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (1.10.1)
Requirement already satisfied: sympy>=1.3 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (1.11.1)
Requirement already satisfied: dill>=0.3 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (0.3.6)
Requirement already satisfied: python-dateutil>=2.8.0 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (2.8.2)
Requirement already satisfied: stevedore>=3.0.0 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (5.1.0)
Requirement already satisfied: symengine<0.10,>=0.9 in ./anaconda3/lib/python3.11/site-packages (from qiskit-terra[visualization]) (0.9.2)
```

- 5) Verify that all required packages are installed by entering:

```
conda list
conda list (package_name)
```

If any packages are missing repeat the appropriate installation steps above.

A screenshot of a macOS terminal window titled "eve — eve@Eves-Air — ~ — zsh — 80x24". The window shows the output of the command "conda list".

```
Last login: Fri Sep 22 03:02:21 on ttys003
[(base) ~] conda list
# packages in environment at /Users/eve/anaconda3:
#
# Name           Version      Build  Channel
_anaconda_depends    2023.07      py311_1
abseil-cpp          20211102.0   hc377ac9_0
aiohttpocore         2.5.0       py311hca03da5_0
aiofiles            22.1.0      py311hca03da5_0
aiohttp              3.8.3       py311h80987f9_0
aioiterools          0.7.1       pyhd3eb1b0_0
aiosignal            1.2.0       pyhd3eb1b0_0
aiosqlite             0.18.0      py311hca03da5_0
alabaster            0.7.12      pyhd3eb1b0_0
anaconda-catalogs    0.2.0       py311hca03da5_0
anaconda-client        1.12.0      py311hca03da5_0
anaconda-navigator     2.4.2       py311hca03da5_0
anaconda-project       0.11.1      py311hca03da5_0
anyio                 3.5.0       py311hca03da5_0
appdirs                1.4.4       pyhd3eb1b0_0
applaunchservices     0.3.0       py311hca03da5_0
appnope                0.1.2       py311hca03da5_1001
appscript              1.1.2       py311h80987f9_0
argon2-cffi           21.3.0      pyhd3eb1b0_0
```

4.0 Performing the Homework Exercise

Using the `jupyter lab` environment. In the terminal window, make sure that you are in the correct `conda` environment (`python39` in the example here). If you are not in the correct environment, enter:

```
conda activate python39 (or the environment name you chose, if different)
```

```

File Edit View Run Kernel Tabs Settings Help
localhost:8890/lab/tree/Desktop/CS8381-fall23/Assignments/HW2
jupyter lab
(jupyter) ~ - conda activate python39
(jupyter) ~ - jupyter lab ServerApp] Package jupyterlab took 0.0000s to import
[1 2023-09-22 03:20:17.535 ServerApp] Package jupyter_lsp took 0.0092s to import
[1 2023-09-22 03:20:17.535 ServerApp] A '_jupyter_server_extension_points' function was not found in jupyter_lsp. Instead, a '_jupyter_server_extensions' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[1 2023-09-22 03:20:17.535 ServerApp] A '_jupyter_server_extension_points' function was not found in jupyter_lsp. Instead, a '_jupyter_server_extensions' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[1 2023-09-22 03:20:17.535 ServerApp] Package jupyter_lsp took 0.0041s to import
[1 2023-09-22 03:20:17.540 ServerApp] Package notebook took 0.0000s to import
[1 2023-09-22 03:20:17.541 ServerApp] Package notebook_shim took 0.0000s to import
[1 2023-09-22 03:20:17.541 ServerApp] Package notebook_shim was not found in notebook_shim. Instead, a '_jupyter_server_extensions' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[1 2023-09-22 03:20:17.541 ServerApp] jupyter_lsp | extension was successfully linked.
[1 2023-09-22 03:20:17.545 ServerApp] jupyter_server_terminals | extension was successfully linked.
[1 2023-09-22 03:20:17.545 ServerApp] jupyter_lab | extension was successfully linked.
[1 2023-09-22 03:20:17.705 ServerApp] notebook_shim | extension was successfully linked.
[1 2023-09-22 03:20:17.720 ServerApp] notebook_shim | extension was successfully loaded.
[1 2023-09-22 03:20:17.721 ServerApp] jupyter_lsp | extension was successfully loaded.
[1 2023-09-22 03:20:17.722 ServerApp] jupyter_lab extension loaded from /Users/eve/Library/python/3.9/lib/python/site-packages/jupyterlab
[1 2023-09-22 03:20:17.723 LabApp] JupyterLab application directory is /Users/eve/Library/Python/3.9/share/jupyter/lab
[1 2023-09-22 03:20:17.723 LabApp] Extension Manager is 'ppyl'.
[1 2023-09-22 03:20:17.726 ServerApp] notebook | extension was successfully loaded.
[1 2023-09-22 03:20:17.726 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[1 2023-09-22 03:20:17.726 ServerApp] The port 8888 is already in use, trying another port.
[1 2023-09-22 03:20:17.726 ServerApp] The port 8889 is already in use, trying another port.
[1 2023-09-22 03:20:17.727 ServerApp] JupyterLab extension loaded from /Users/eve/Library/python/3.9/lib/python/site-packages/jupyterlab
[1 2023-09-22 03:20:17.727 ServerApp] Jupyter Server 2.7.3 is running at:
[1 2023-09-22 03:20:17.727 ServerApp] http://localhost:8890/lab?token=0lad5aa7026dcc311829d244b6a91f6c8c4b68cb17e95
[1 2023-09-22 03:20:17.727 ServerApp] http://127.0.0.1:8890/lab?token=0lad5aa7026dcc311829d244b6a91f6c8c4b68cb17e95
[1 2023-09-22 03:20:17.727 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[1 2023-09-22 03:20:17.732 ServerApp]

Simple 1 0 0
jupyter lab 1

```

- i) To activate the jupyter notebook within your default browser, enter:

jupyter lab This command should open a web application in your default web browser (note, I used Mac OSX Safari, but Firefox is usually the preferred browser). Within the web application, browse to the jupyter notebook file, QuantumLab5.ipynb, and click on it to open it. You can navigate to the notebook by using the leftmost side panel that functions as the file directory tool. Alternatively, if you set up the kernel tab using the “python -m ipykernel install ...” command discussed previously, you can simply select that widget to restart and shutdown the kernel.

The screenshot shows a Jupyter Notebook interface with two tabs open: 'HW2 - JupyterLab' and 's2.smu.edu_~mitch_class_8381_week5_QC_HW2-assign.ipynb'. The notebook content includes:

Environment Check

Check that environment containing quantum libraries is being used. qc jupyter kernel is running using libraries in the conda environment

```
[1]: ! conda env list
# conda environments:
# base                  /Users/eve/anaconda3
Code                  /Users/eve/anaconda3/envs/Code
python39              * /Users/eve/anaconda3/envs/python39
```

```
[2]: import sys
print(sys.executable)
/Library/Developer/CommandLineTools/usr/bin/python3
```

```
[3]: ! jupyter kernelspec list
Available kernels:
python3    /Users/eve/Library/Python/3.9/share/jupyter/kernels/python3
```

Homework 2: Basic Quantum Programs

First, we will begin by importing Qiskit, IBM's quantum computing software that is written in Python! To create an experiment and run it, we will need to use Qiskit's Circuits, Registers, and Compilers. Programming using qiskit allows us to programmatically extend QASM code to use for loops and if statements to design circuits faster. QASM is an assembler-like instruction set that allows you to specify quantum circuits.

Import libraries and simulator

Aes is a backend simulator for the IBM quantum computers (other backends are run on real QC). Numpy is good with arrays and matrices. pyplot is for visualization of probability distributions.

- ii) Start at the top of the notebook and follow the directions written in text, and by running the cells with Python/QASM source code, inserting your own code where requested, and answering the questions by typing your answer in the markdown cells. You can select code cells and run them by clicking on the “run/play” button in the top ribbon menu (the small triangle), or, you can select the cell and use <SHIFT> <ENTER> <ENTER> to run the cell.
- iii) The first few cells are fully populated with source code, so you will just execute those as is. Later cells require you to enter your own code to run. Note that the notebook also has homework questions embedded within it. You can answer those questions by adding text to the notebook in a markdown cell that follows each question. Be sure to save your notebook before exiting `jupyter lab` and please export your notebook to .html before turning it in by email.
- iv) To exit `jupyter lab`, enter “<CTRL> C” in the terminal window. For Mac OSX users, note that it is actually the <CTRL> key and NOT the <CMD> key that is used.

```

eve@Eves-Air: ~
[1 2023-09-22 03:20:17.727 ServerApp] http://localhost:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
[1 2023-09-22 03:20:17.727 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2023-09-22 03:20:17.727 ServerApp] 
To access the server, open this file in a browser:
file:///Users/eve/Library/Jupyter/runtime/jpserver-7444-open.html
Or copy and paste one of these URLs:
http://localhost:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
http://127.0.0.1:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
[1 2023-09-22 03:20:17.727 ServerApp] Note: non-SSL connections are not supported.
[1 2023-09-22 03:21:15.185 ServerApp] Building notebook at /Users/eve/Desktop/CS8381-fall23/Assignments/HW2/s2.smu.edu.ipynb
[1 2023-09-22 03:21:15.185 ServerApp] Kernel started: 6a9a0793-828b-499a-9bb2-ed0e49da6c48
[1 2023-09-22 03:21:15.533 ServerApp] Connecting to kernel 6a9a0793-828b-499a-9bb2-ed0e49da6c48
[1 2023-09-22 03:21:15.533 ServerApp] Connected to kernel 6a9a0793-828b-499a-9bb2-ed0e49da6c48.
[1 2023-09-22 03:21:15.545 ServerApp] Connecting to kernel 6a9a0793-828b-499a-9bb2-ed0e49da6c48.
[1 2023-09-22 03:21:15.545 ServerApp] Connected to kernel 6a9a0793-828b-499a-9bb2-ed0e49da6c48.
[I 2023-09-22 03:21:15.565 ServerApp] Got events for closed stream object at 0x106e91640>
[1 2023-09-22 03:23:15.442 ServerApp] Saving file at /Desktop/CS8381-fall23/Assignments/HW2/s2.smu.edu_mitc...class_8381_week5_OC_HW2-assign.ipynb
[1 2023-09-22 03:23:15.442 ServerApp] Saving file at /Desktop/CS8381-fall23/Assignments/HW2/s2.smu.edu_mitc...class_8381_week5_OC_HW2-assign.ipynb
[1 2023-09-22 03:53:10.958 ServerApp] interrupted
[1 2023-09-22 03:53:10.958 ServerApp] Serving notebook from local directory: /Users/eve
1 active kernel
Jupyter Server 2.7.3 is running at:
http://localhost:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
http://127.0.0.1:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
Shutdown this Jupyter server [y/n]? y[1 2023-09-22 03:53:15.963 ServerApp] No answer for 5s:
[I 2023-09-22 03:53:15.964 ServerApp] resuming operation...
[1 2023-09-22 03:53:15.964 ServerApp] interrupted
[1 2023-09-22 03:53:15.964 ServerApp] Serving notebooks from local directory: /Users/eve
1 active kernel
Jupyter Server 2.7.3 is running at:
http://localhost:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
http://127.0.0.1:8890/lab?token=0lad5aa7026dc311829d244b64a91f6c8c4cb68cb17e95
Shutdown this Jupyter server [y/n]? y[1 2023-09-22 03:53:15.881 ServerApp] Shutdown confirmed
[1 2023-09-22 03:53:15.881 ServerApp] Shutting down all extensions
[1 2023-09-22 03:53:15.889 ServerApp] Shutting down 1 kernel
[1 2023-09-22 03:53:15.909 ServerApp] Kernel shutdown: 6a9a0793-828b-499a-9bb2-ed0e49da6c48
(python39) ~ 

```

5.0 Submitting the Completed homework

Follow all directions in the notebook (read carefully). When your homework is complete and your notebook has been properly updated, save it to your disk using the “file” drop down menu within the web browser and export it to .html. Make sure that all output cells are properly populated with the output of each code cell. I strongly suggest that you first email the completed notebook to yourself, then open it and examine the content to make sure that you are turning in the updated and completed notebook. Notebooks turned in without the updated source code, output cells, and question answers will be counted as late submissions. When you are sure that you have a properly updated notebook file, please email it to the instructor and the TAs. Send the email to:

mitch@smu.edu

and

erhenderson@smu.edu

and

hendersonj@smu.edu

Have fun programming an actual quantum computer!