**University of Electronic Science and Technology of China**

**School of Information and Software Engineering**

**Experiment Report**

Student ID： 20232409107

Student Name： Jahid Shahidul Islam

Course Name： Python Practical Programming

Experimental Teacher： Rao Yunbo

1. Experiment title：Install Python and PyCharm

2. Experiment hours：4h Experiment location: Software Building 400

3. Objectives

At the end of this experiment, you will be able to:

* How to install Anaconda in your devices?
* How to create your first python program and packaging?
* How to install PyCharm?

4.Experiemental contents & step

(1) Installing the Anaconda for Windows

(2) Install PyCharm process

(3) Create your first python project process

(4) python program packaging

(5) other

Please show experimental results and pictures in your report.

For example:

**<1>Installing the JDK for Windows**

**Step 1:** Find a folder named JDK, and inside that folder you will find an executable file with the following name: *jdk-6uXX-windows-i586-p.exe*

The *XX* in the filename will be the software’s update number. For example, if the CD contains update 10, the *XX* in the filename will be jdk-6u10-windows-i586-p.exe. This file is an executable program that will install the Java™ 6 JDK for Windows on your computer. Run the file.

**Step 2:** A license agreement screen, such as the one shown in Figure will appear next. Read the agreement, then click the *Accept* button.

**Step 3:** A Custom Setup screen, such as shown in Fig.1, appears next. Make sure *Development Tools* is selected. Also, notice that at the bottom of the window you see the words “Install to:” followed by the path C:\Program Files\Java\jdk1.6.0\_XX. (The “XX” will be a number.) This is the location where the fi les will be installed. Click the *Next* button.



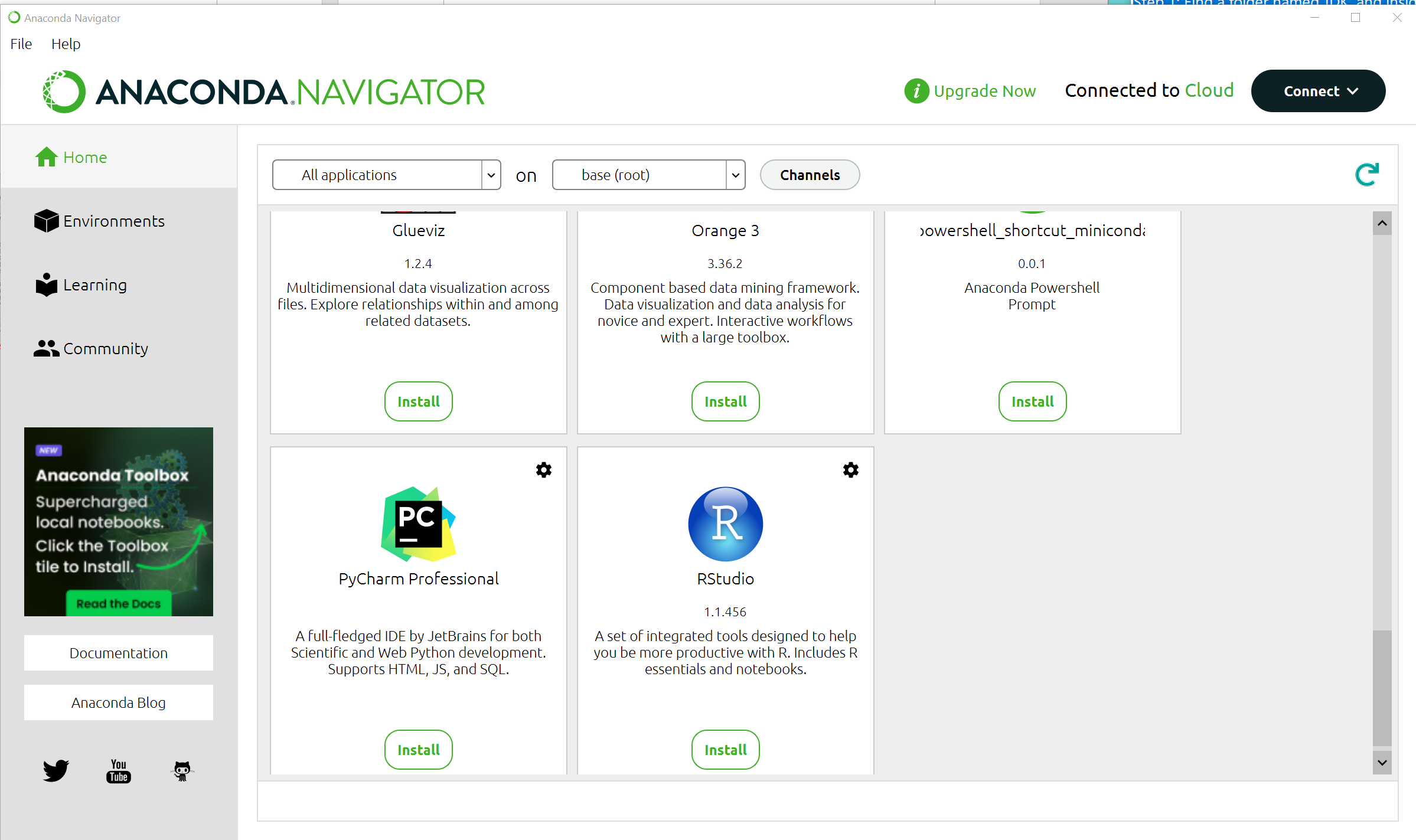
Fig.1 Custom Setup screen

…………………….

5. Experimental analysis

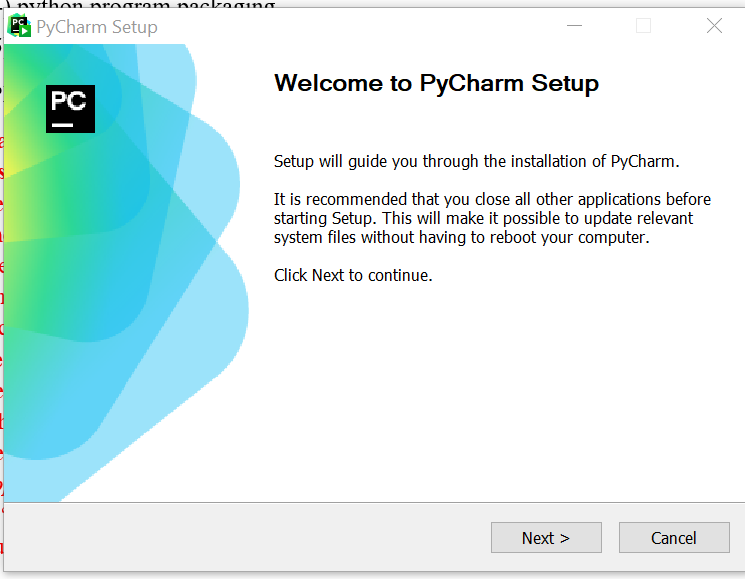
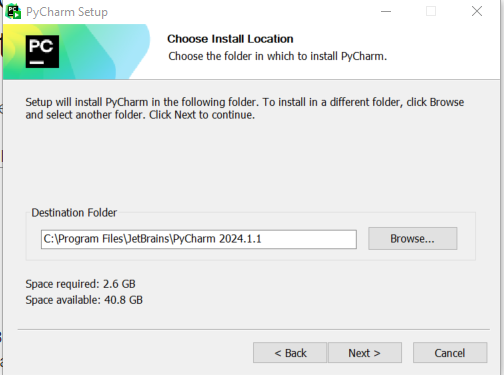
# Installing Anaconda for Windows:

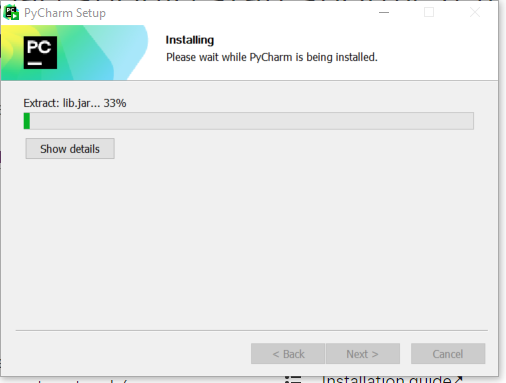
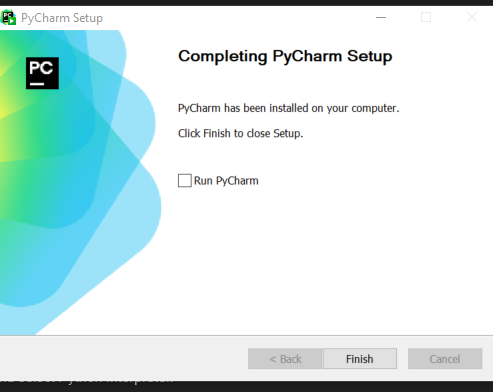
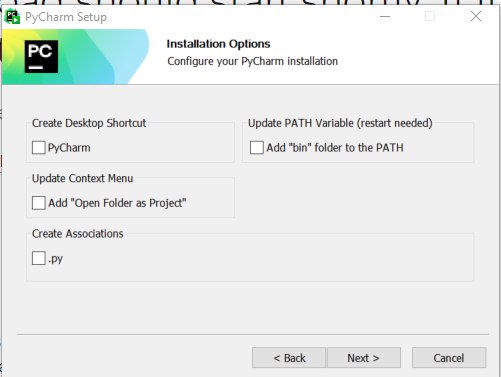
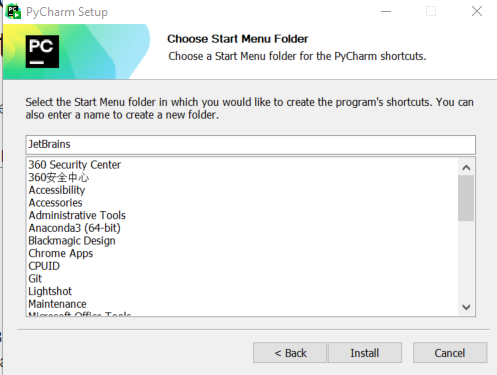
1. **Download**: Get the Anaconda installer for Windows from the official website.
2. **Launch**: Double-click the downloaded file to start the installation.
3. **Agree**: Read and accept the license agreement.
4. **Choose Install For:** Select "Just Me" unless you need it for all users.
5. **Destination**: Pick a folder in C drive without spaces or special characters for installation.
6. **Install**: Click "Install" and wait for the process to complete.
7. **Finish**: Click "Next" and "Finish" to exit the setup.



# Installing PyCharm:

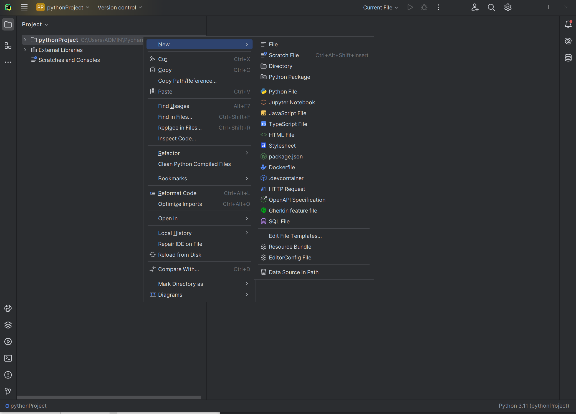
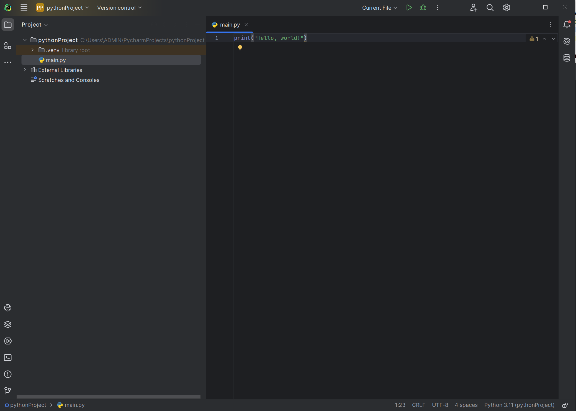
1. **Download**: Get the PyCharm installer (Community or Professional) from https://www.jetbrains.com/pycharm/download/.
2. **Run**: Open the downloaded file to start the installation.
3. **Next**: Click "Next" through the setup wizard, choosing options like installation location and shortcuts as desired.
4. **Install**: Click "Install" and wait for the process to finish.
5. **Finish**: Click "Finish" to exit the setup, optionally launching PyCharm.



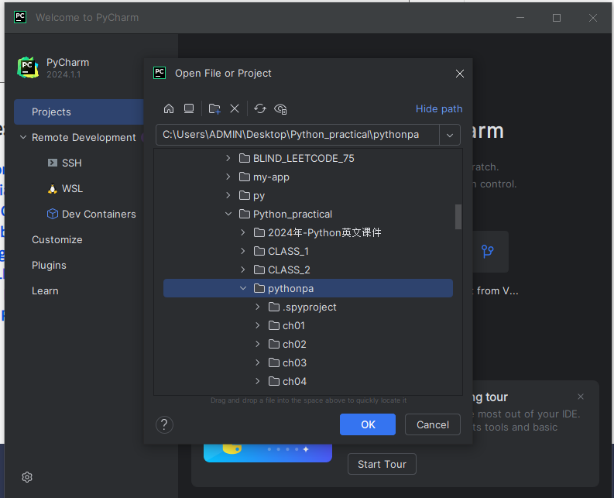


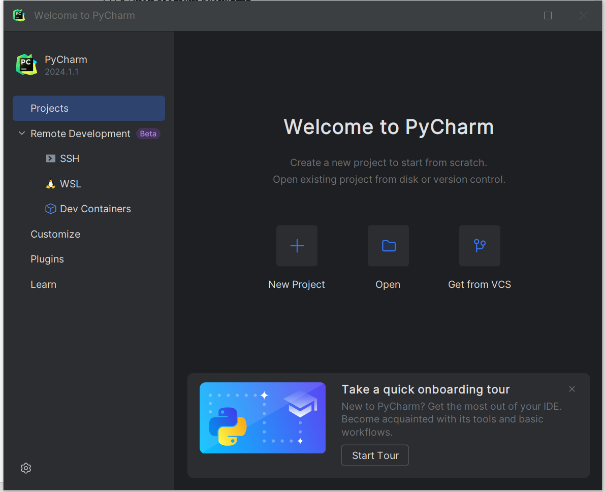
# Creating Your First Python Project:

1. **Open PyCharm:** Launch the PyCharm application.
2. **New Project:** Click "Create New Project" on the welcome screen.
3. **Set Up:** Choose location, name, and interpreter, then click "Create".
4. **Make File:** Right-click project folder -> New -> Python File (e.g., "main.py").
5. **Write Code:** Type your Python code in the file (e.g., print("Hello, world!")).
6. **Run:** Right-click in the code -> Run 'main.py'.
7. **See Results:** Check the bottom of PyCharm for your code's output.

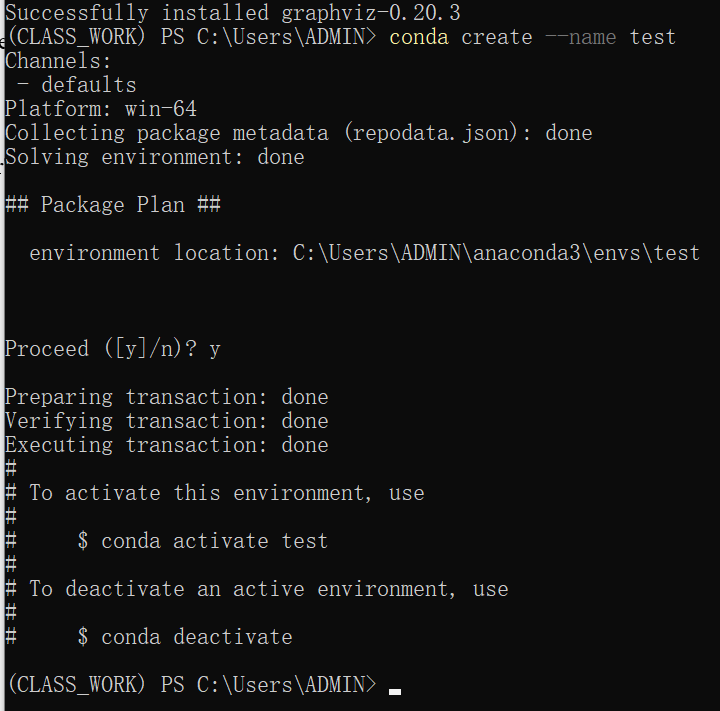
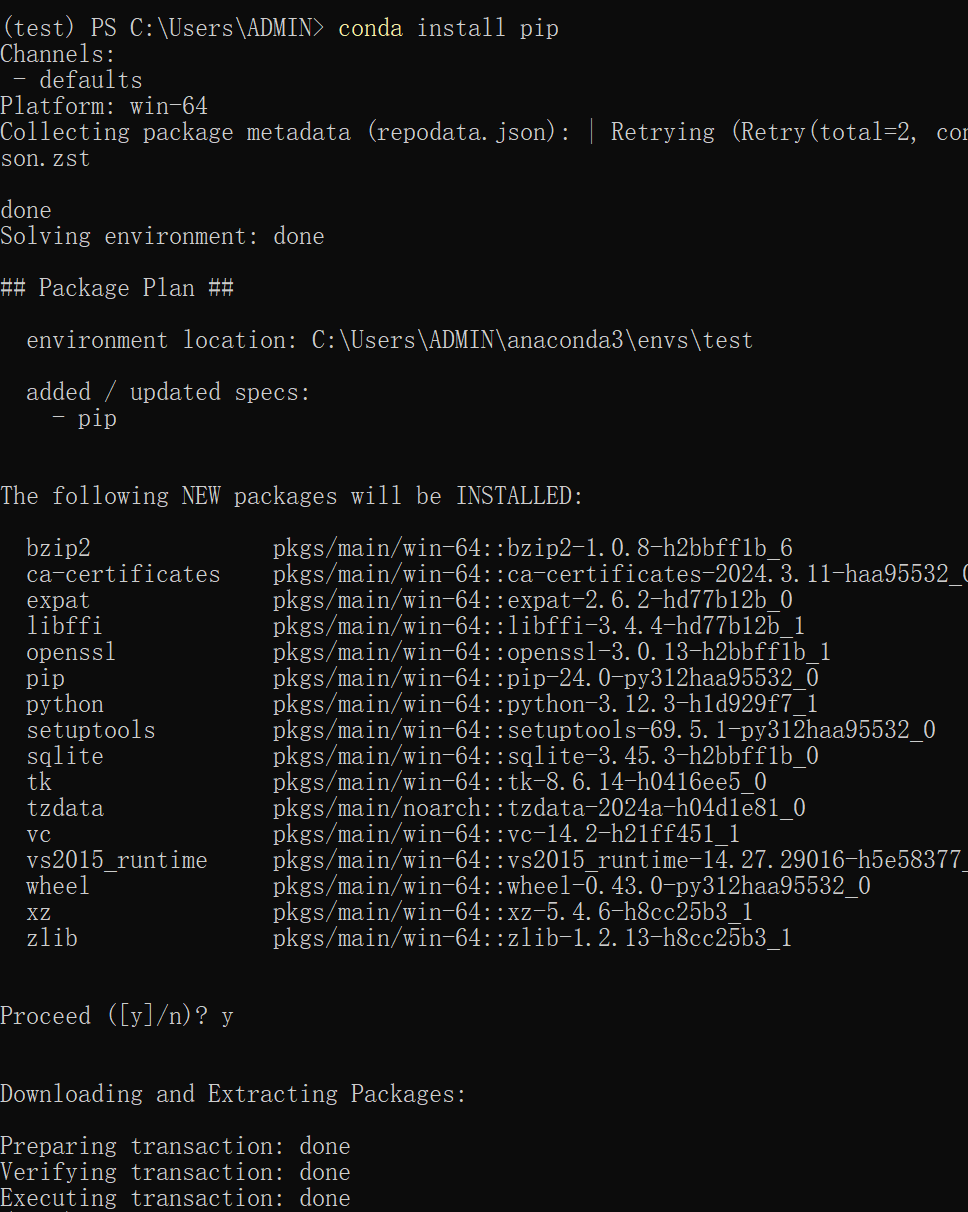


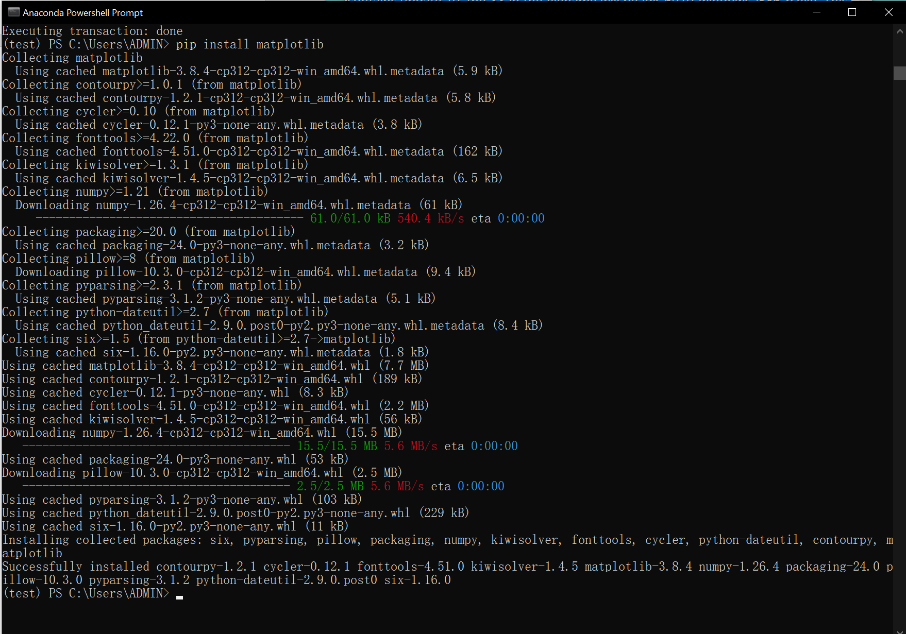
* **Open Existing Python Project:**

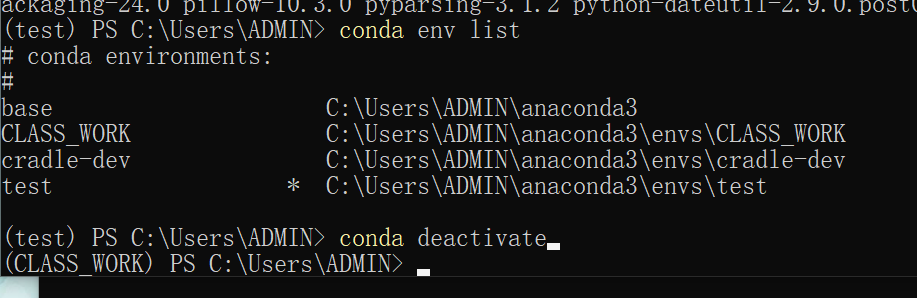
1. **Open PyCharm:** Launch the PyCharm application.
2. **Open Project:** Click "Open" on the welcome screen.
3. **Choose Folder:** Find and select your existing project folder.
4. **Open as Project:** PyCharm will recognize it as a Python project and open it.
5. **Explore & Code:** You can now view your old Python files, edit them, and start coding!

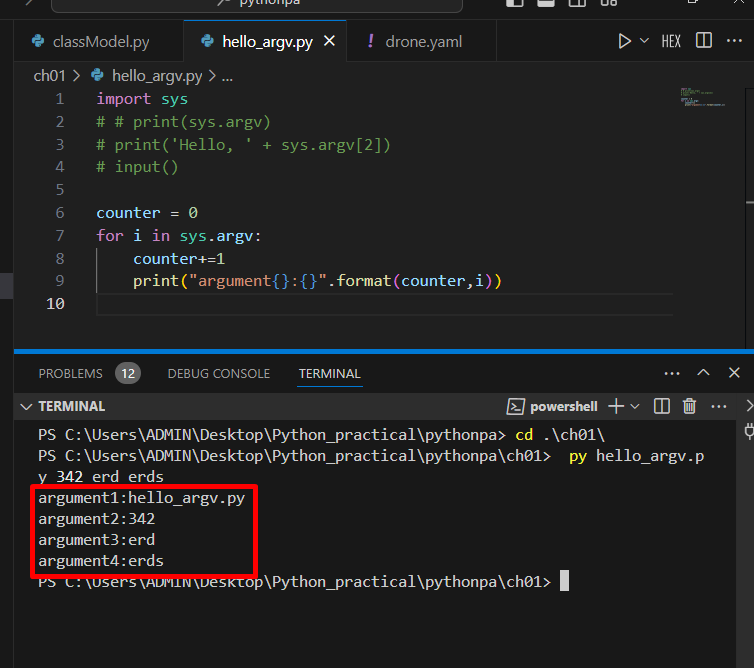


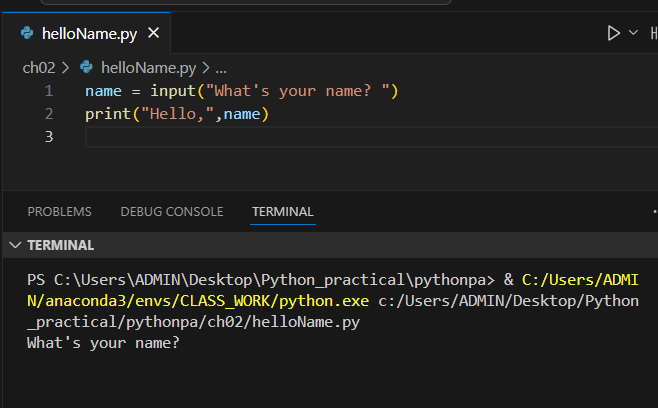
# Packaging Python Programs with Conda:

1. **Create Environment**: conda create -n test
2. **Activate:** conda activate test
3. **Install Pip (if needed):** conda install pip
4. **Install Packages:** pip install matplotlib graphviz
5. **List Packages (optional):** conda list
6. **Deactivate (optional):** conda deactivate



1. Python Problems:





# Experimental Analysis:

This initial experiment was all about laying a strong foundation for our exploration of practical Python programming. Think of it as carefully setting up our workspace before embarking on a complex project. We started by installing Anaconda, a powerful toolkit that bundled together everything we need to start coding, from core Python libraries to specialized tools for data science and visualization like NumPy, SciPy, and matplotlib.

Anaconda's secret weapon, the "conda" package manager, became our trusted assistant in organizing this toolkit. With conda, we learned to create isolated project spaces, preventing conflicts and ensuring that each project had the right tools at hand. This is akin to organizing our workspace with clearly labeled compartments, ensuring everything has its place.

We also welcomed PyCharm into our workflow, a sophisticated IDE that acts like an intelligent assistant, streamlining our coding with features like code completion and error detection. Imagine having a helpful guide whispering suggestions and catching mistakes before they cause problems – that's PyCharm!

By creating a dedicated "test" environment using conda, we learned to manage project dependencies, ensuring smooth operation and reproducibility. This is crucial, like ensuring all the parts of a machine work together harmoniously. We then expanded our toolbox by installing packages like matplotlib, which will empower us to create insightful data visualizations, and graphviz, a powerful tool for representing complex relationships.

Finally, we got our hands dirty with introductory Python problems, practicing our skills with basic exercises. Think of this as testing the sharpness of our tools and getting a feel for the materials before starting a masterpiece.

This experiment has successfully equipped us with the essential tools and techniques for navigating the world of Python. We've established a solid workspace, organized our tools, and practiced our skills. Now, with confidence and excitement, we're ready to dive deeper into more advanced concepts and tackle real-world challenges with Python. Our journey has just begun, and we're eager to see what we can build!

Report score：

Instructor's signature：