Faculty of Information Technology
University of Moratuwa
BSc Hons in Information Technology
BSc Hons in Information Technology Management
IN 4410 – Big Data Analytics

Level 1 - Semester 2 **Lab Sheet 06**

Spark with Python (PySpark)

After installing and testing the spark. We can check if the python version is working or not. Using this command, you can start the python in spark shell (pyspark)

```
cd C:\Bigdata\Spark\spark-3.5.1\bin
```

```
pyspark
```

```
C:\Bigdata\Spark\spark-3.5.1\bin>pyspark
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.1
      /_/

Using Python version 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024 21:26:36)
Spark context Web UI available at http://host.docker.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1710659893463).
SparkSession available as 'spark'.
>>>
```

Then type this command, to check whether the python is working or not.

```
x=sc.textFile("C:/Spark/spark-3.5.1/README.md")

x
```

In here, we declare a variable x for spark context - text file to test the spark. test.txt file is the testing file.

```
>>> x=sc.textFile("C:/Spark/spark-3.5.1/README.md")
>>> x
C:/Spark/spark-3.5.1/README.md MapPartitionsRDD[1] at textFile at NativeMethodAccessorImpl.java:0
>>>
```

To apply any operation in PySpark, we need to create a **PySpark RDD**

Then add another y value to map the transformation. Because we need to convert the data of x text file into upper case. In python normally use **lambda, if** we want to do any kind of transformations.

```
y = x.map(lambda line: line.upper())
y
```

```
>>> y = x.map(lambda line: line.upper())
>>> y
PythonRDD[2] at RDD at PythonRDD.scala:53
>>>
```

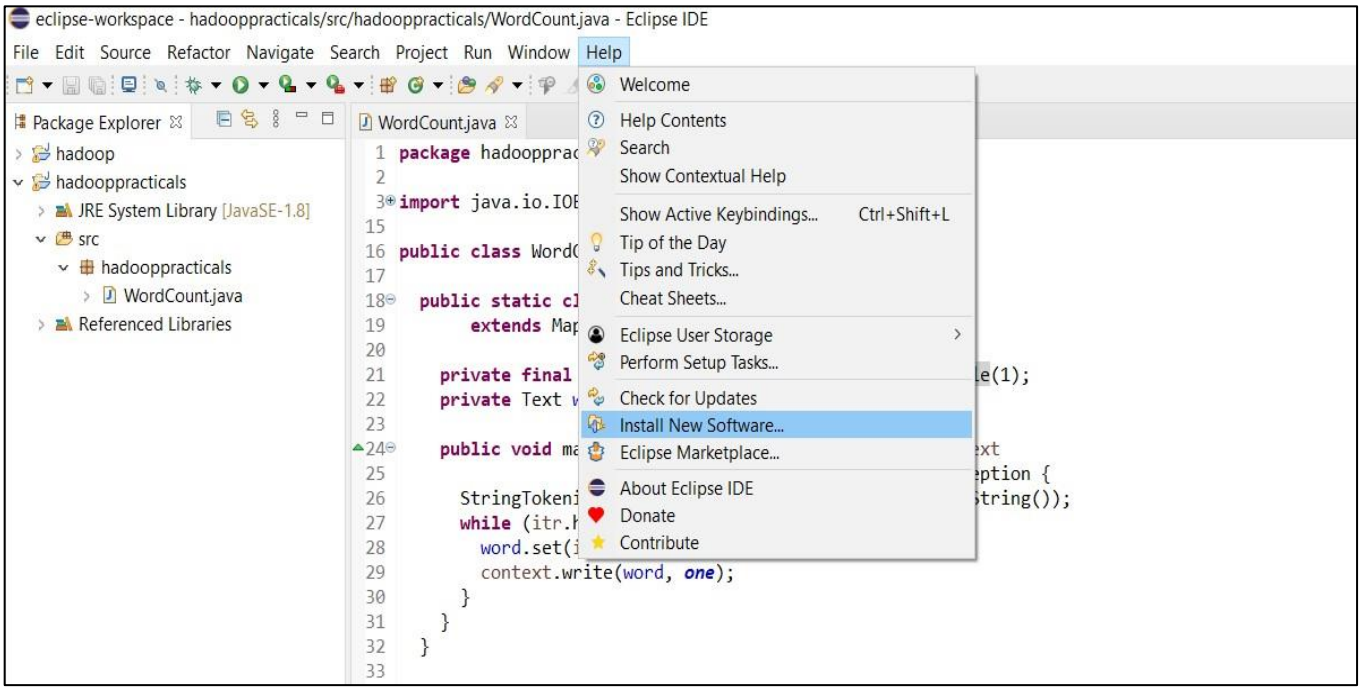
In the collect command you can see the output lines in the given text file.Using this code, we can see the content of the given text file.

```
y.collect()
```

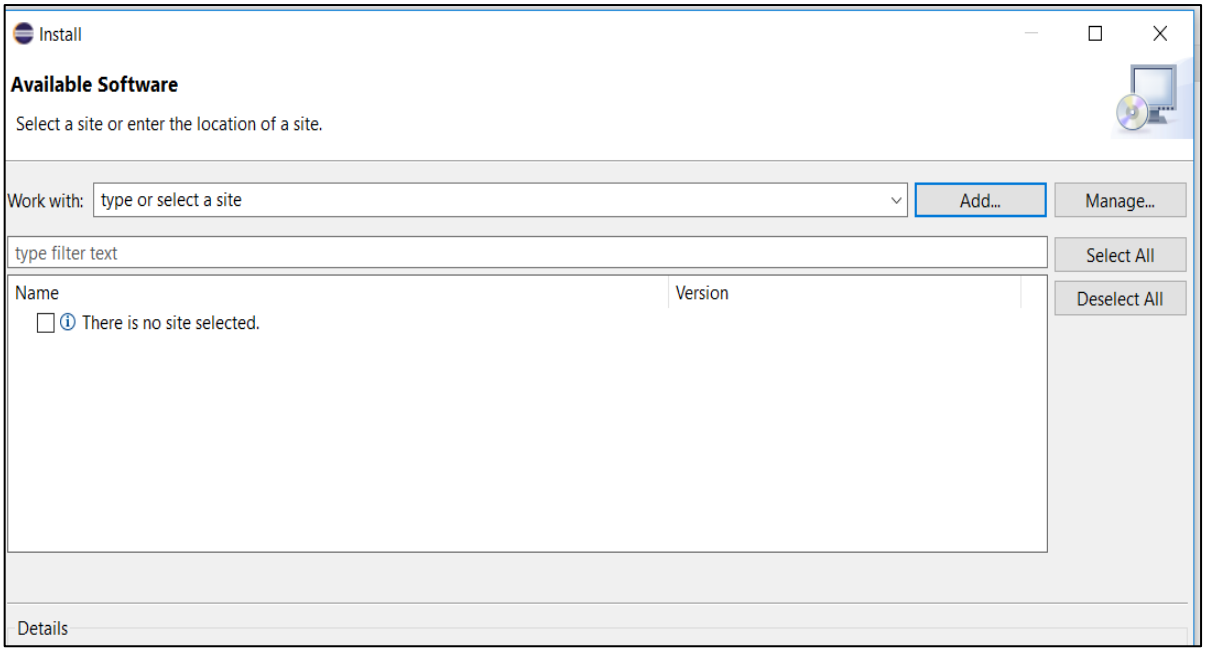
```
>>> y.collect()
['APACHE HADOOP IS A COLLECTION OF OPEN-SOURCE SOFTWARE UTILITIES THAT
ERS TO SOLVE PROBLEMS INVOLVING MASSIVE AMOUNTS OF DATA AND COMPUTATIO
TRIBUTED STORAGE AND PROCESSING OF BIG DATA USING THE MAPREDUCE PROGRA
 FOR COMPUTER CLUSTERS BUILT FROM COMMODITY HARDWARE WHICH IS STILL TH
 CLUSTERS OF (HIGHER-END) HARDWARE. ALL THE MODULES IN HADOOP ARE DESI
DWARE FAILURES ARE COMMON OCCURRENCES AND SHOULD BE AUTOMATICALLY HAND
>>>
```

PySpark with eclipse IDE using PyDev module

- Go to the menu Help → Install new Software

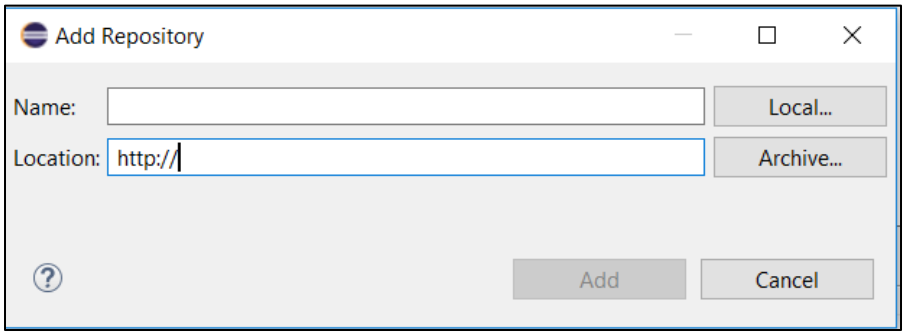


- From the “Install” window: click add button.



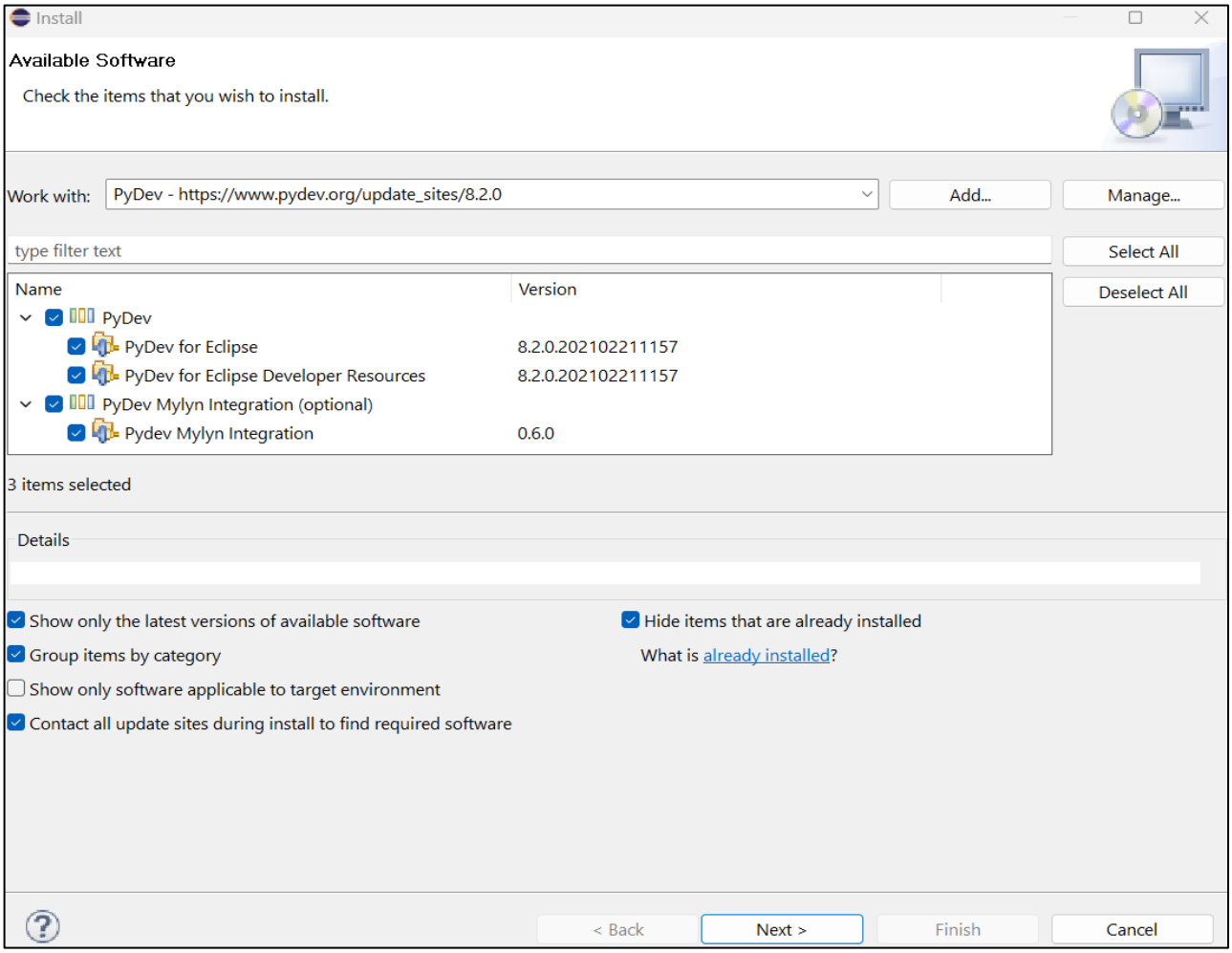
From the “Add Repository” dialog box:

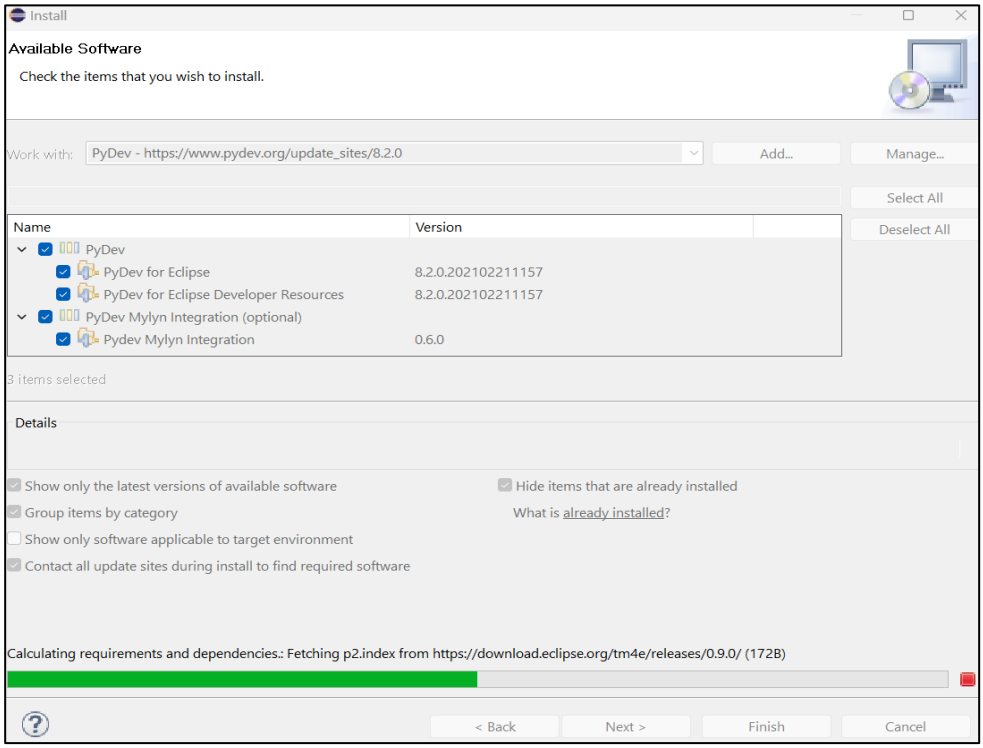
- Fill the field Name: PyDev
- Fill the field Location: https://www.pydev.org/update_sites/8.2.0



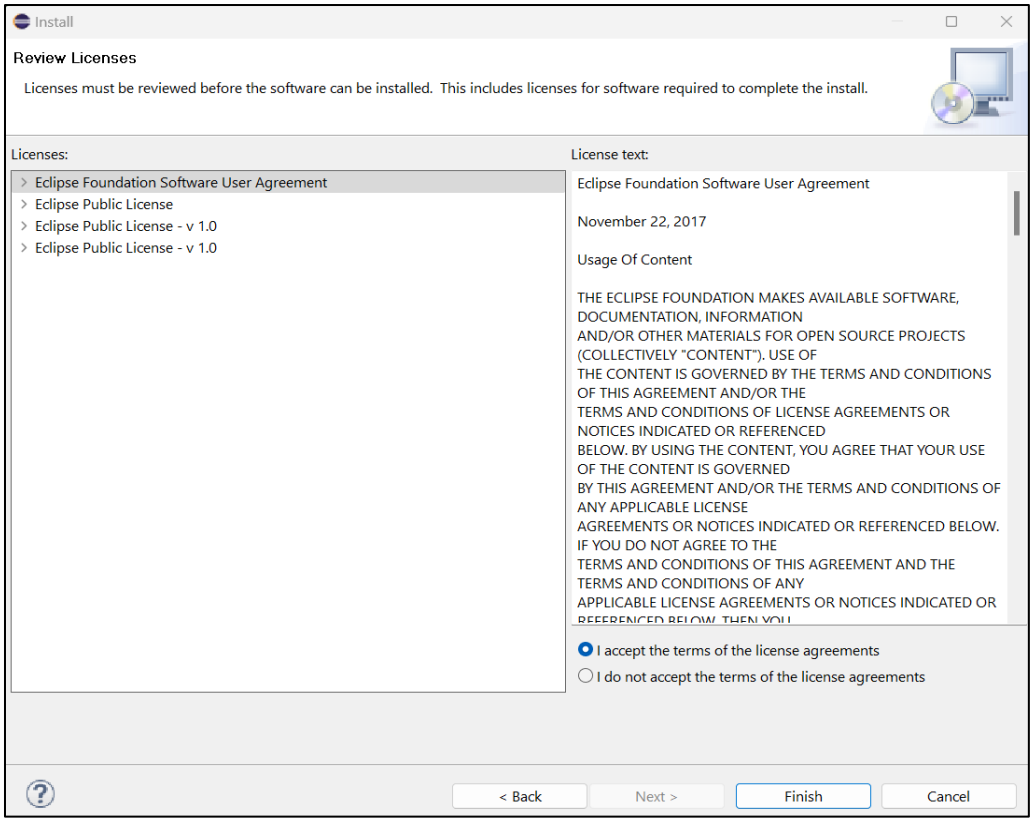
Validates with the button **OK**

Click all the ticks in here and Press Next button.

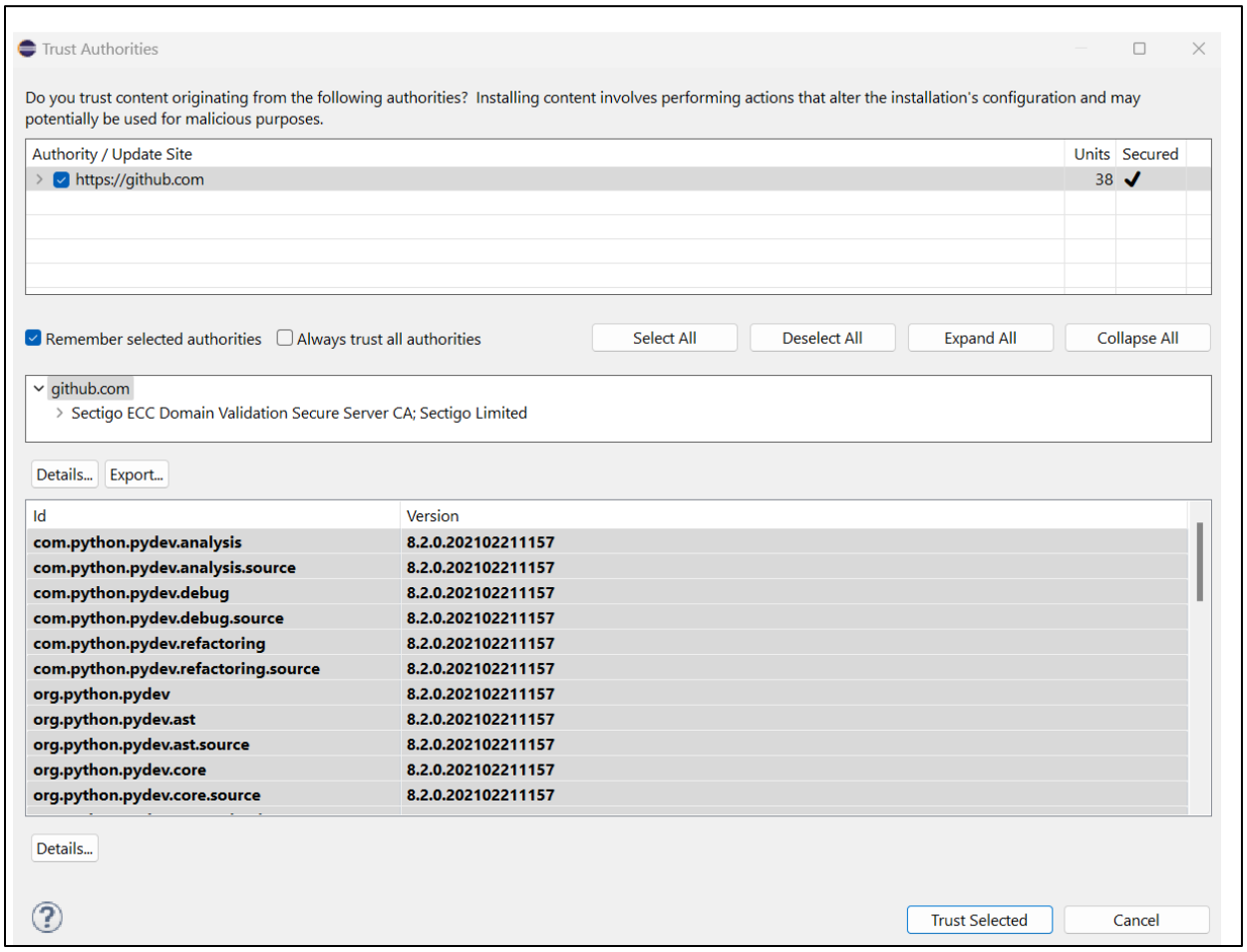




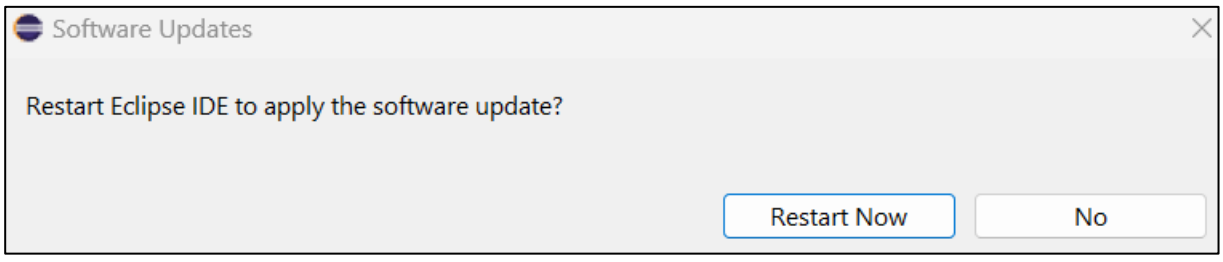
Click the next button and accept the agreement and install the PyDev feature.



And then select the github authorities and press Trust Selected button

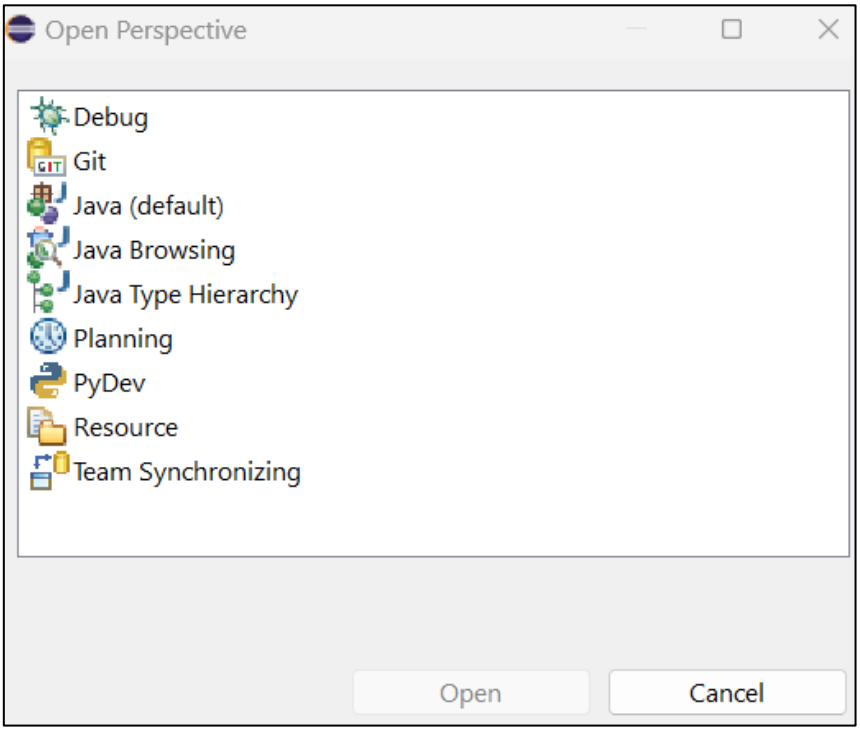


Then you can restart again the Eclipse IDE

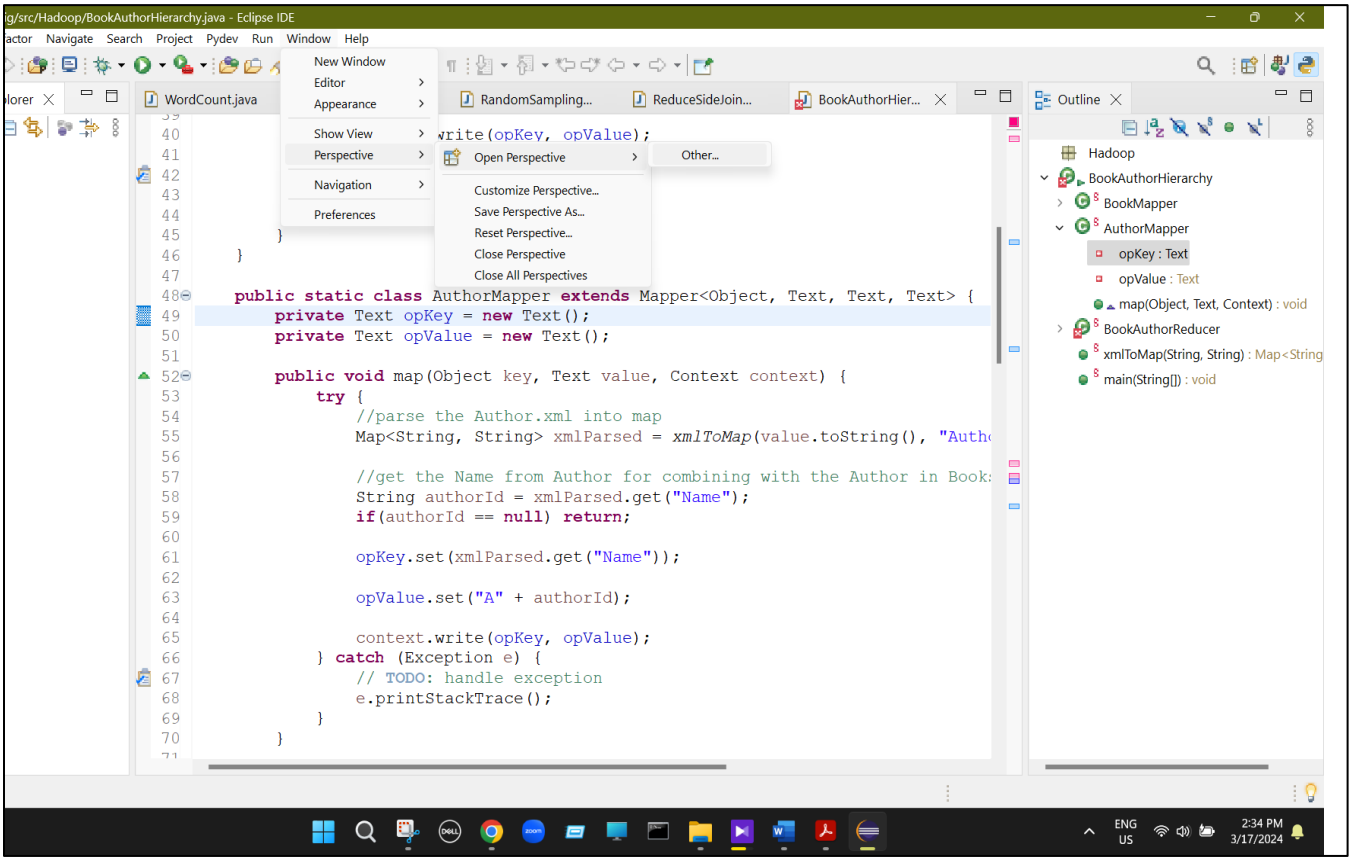


And then finish and restart the eclipse again as run as administrator.

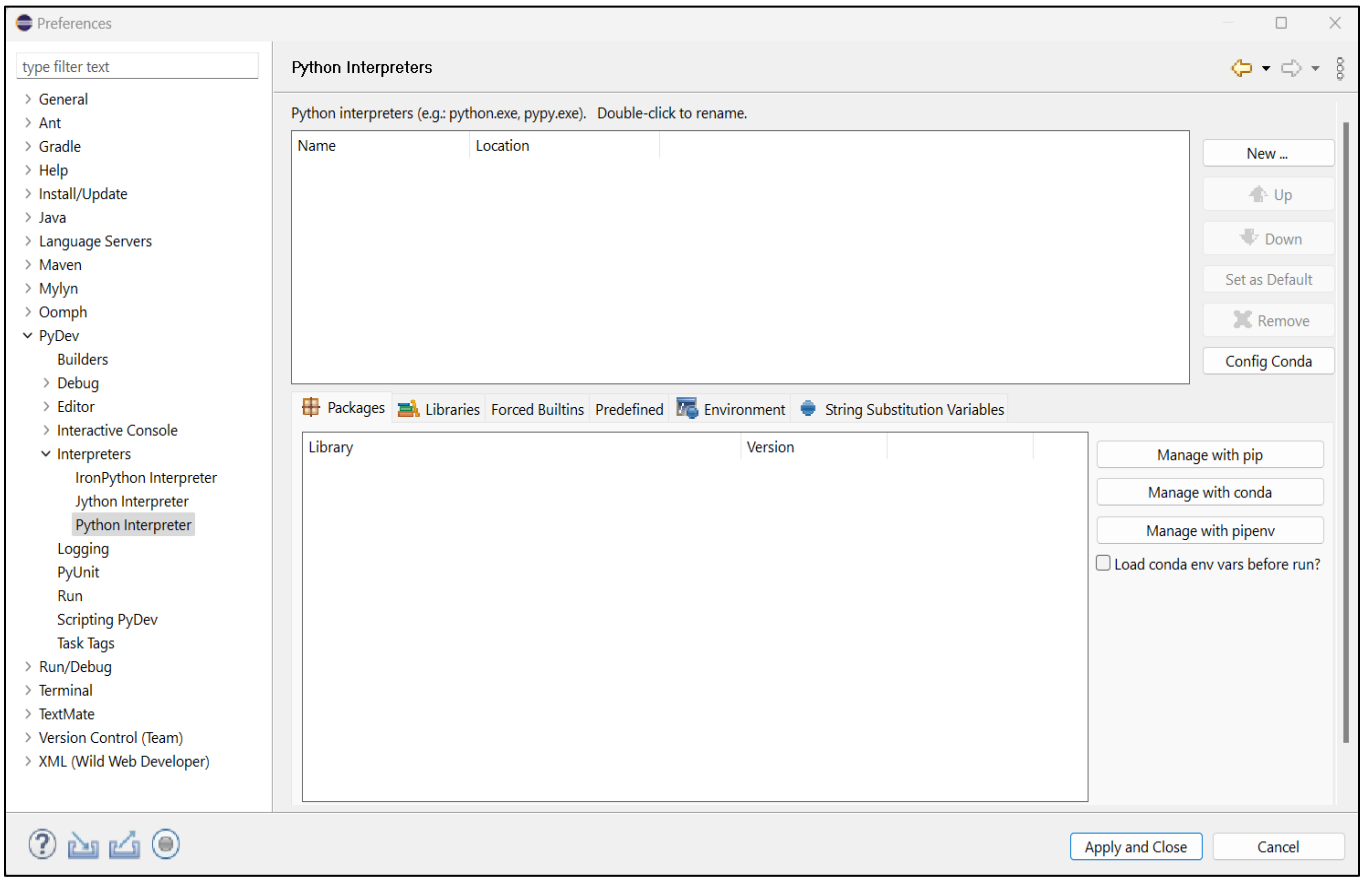
Then click the open perspective and double click the PyDev icon.



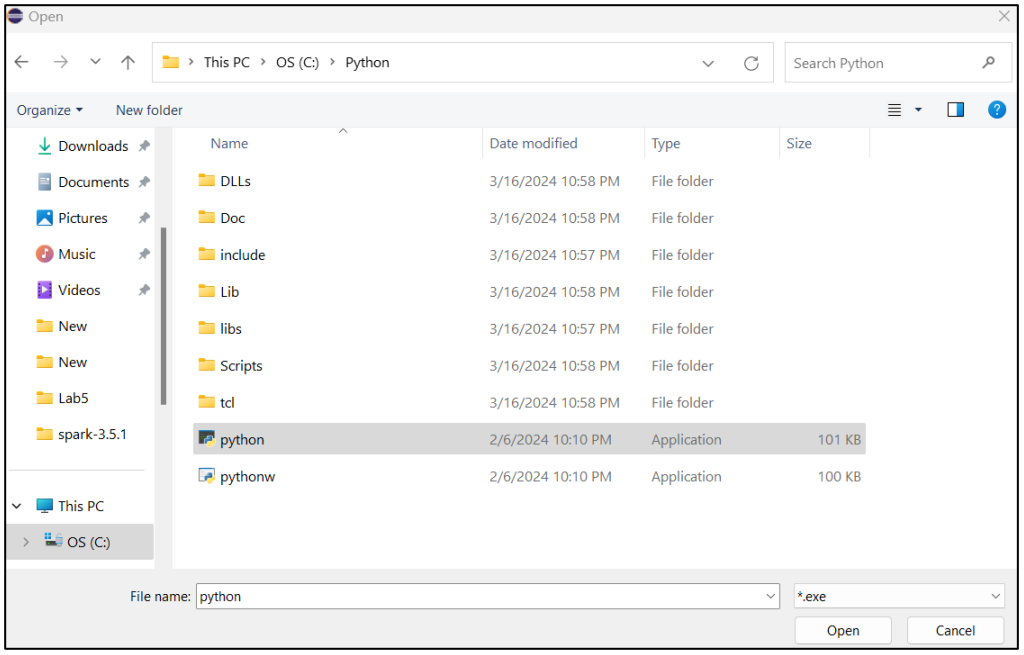
Then go to Window ==> Perspective ==> Open Perspective ==> Other



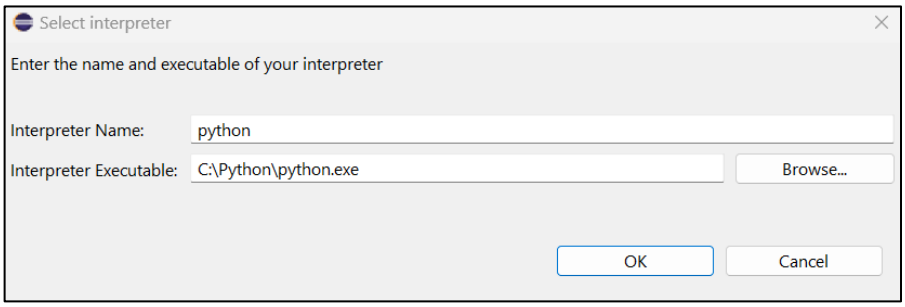
Then go to Window ==> Preferences ==> PyDev ==> Interpreters ==> Python Interpreters



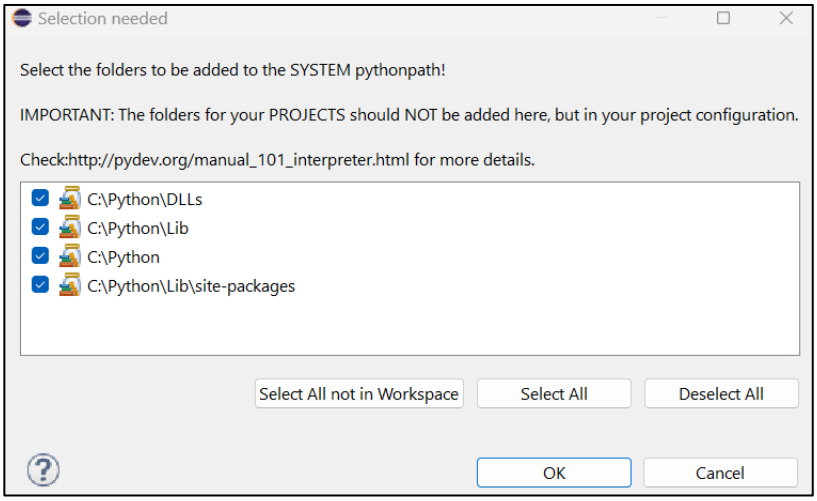
Click New button and Browse the **python.exe** file in Python folder in C: drive



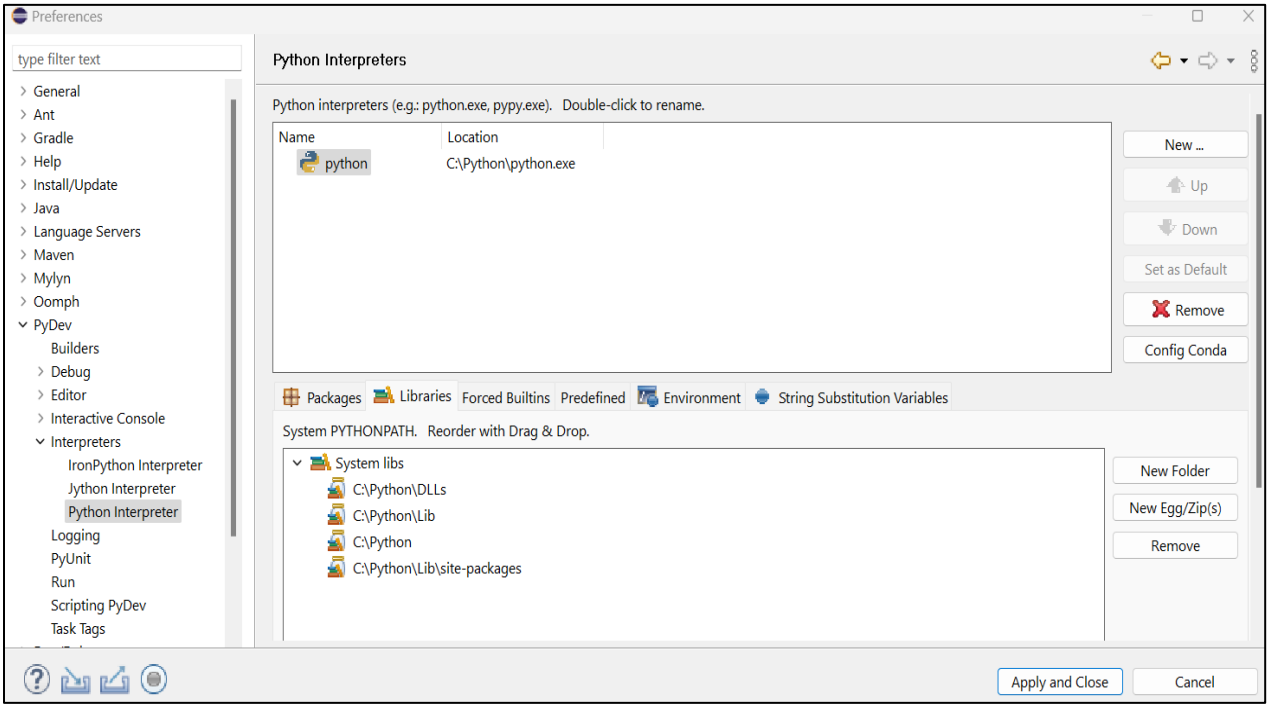
Also type interpreter name as **python**



Then it will appear in this box, you can click Ok for that.



Then it will appear in this box within the libraries

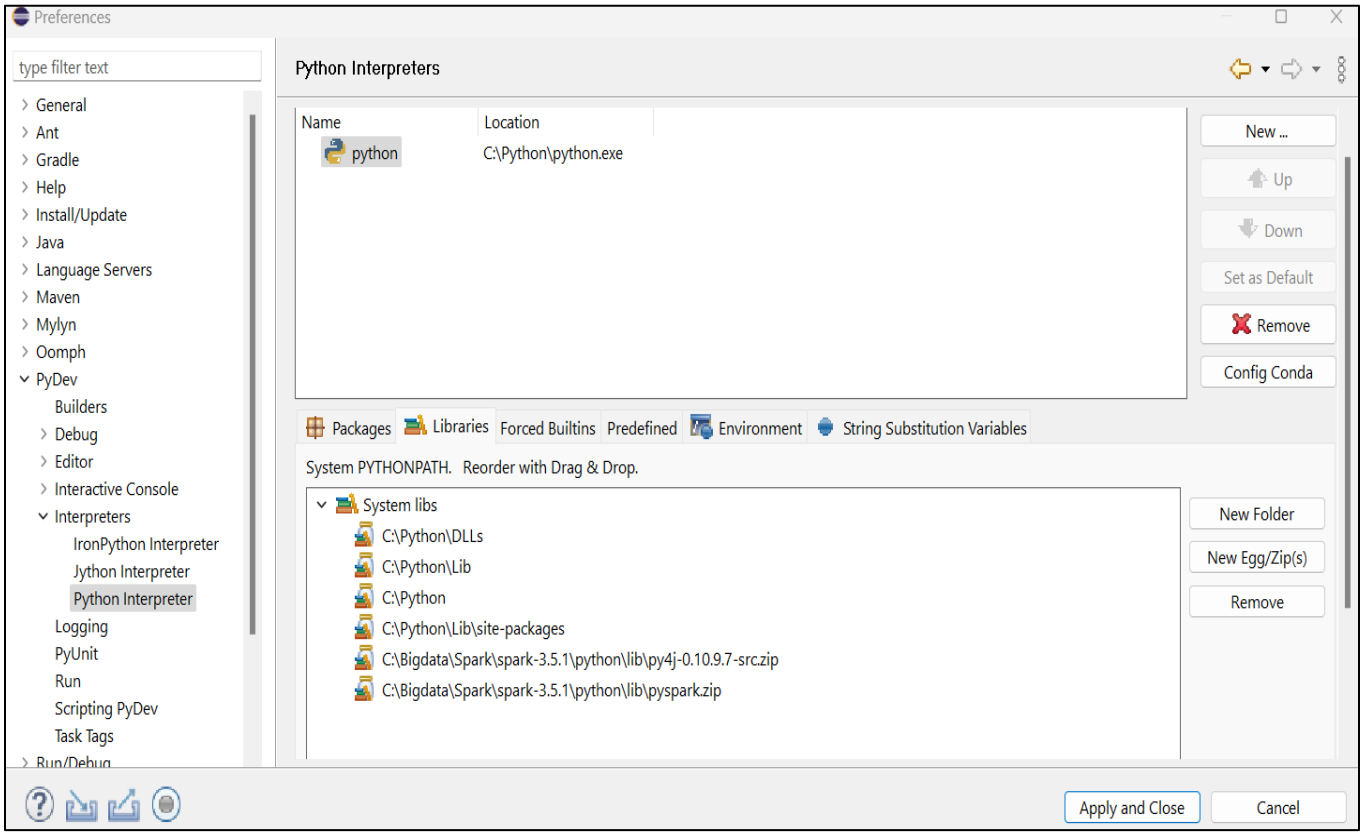


Then go to the libraries and add the pyspark files using **New/Egg/Zips button**

After that select, the **zip** folder type and go to this path (Sometimes path may be different according to user's machine)

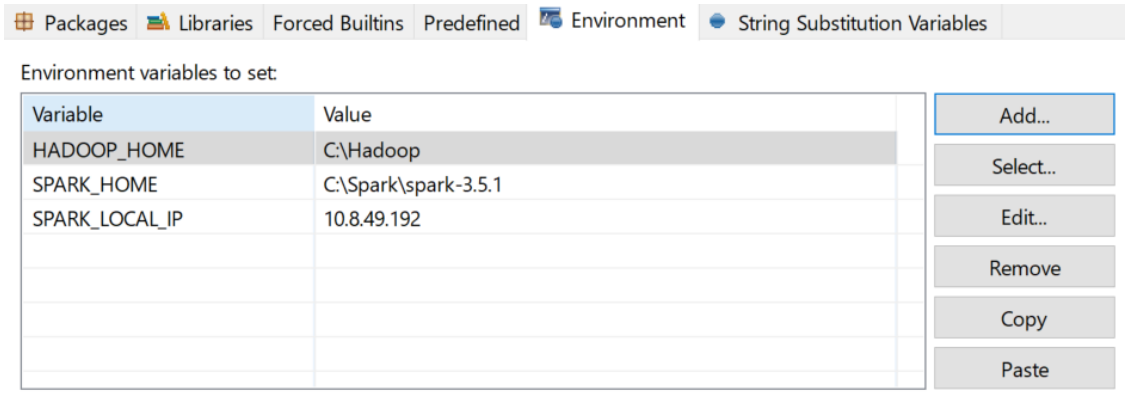
```
C:\Spark\spark-3.5.1\python\lib\py4j-0.10.9-src.zip
```

```
C:\Spark\spark-3.5.1\python\lib\ pyspark
```

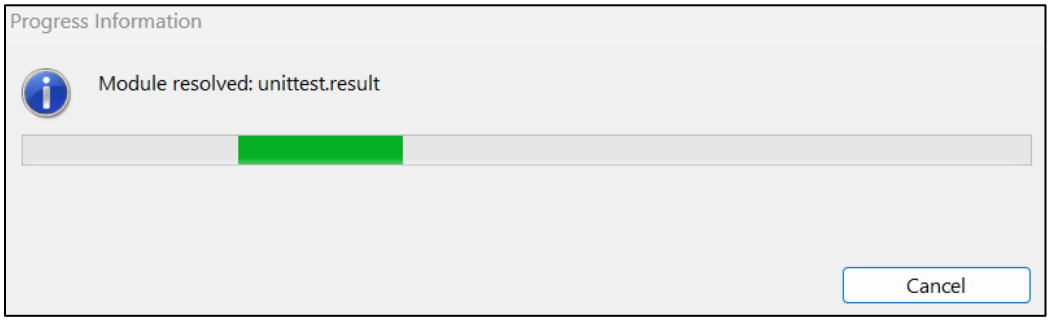


And then go to the Environment and **Add** Hadoop Home, Spark Home and Spark localIP Address

To get the local ip address, go to cmd, and type **ipconfig** command and then get the IPv4 address from there.



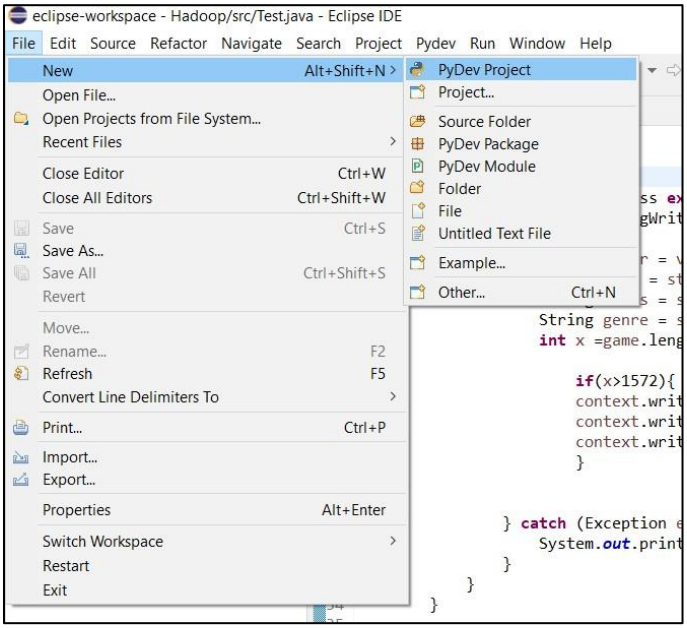
Then click Apply and Close.



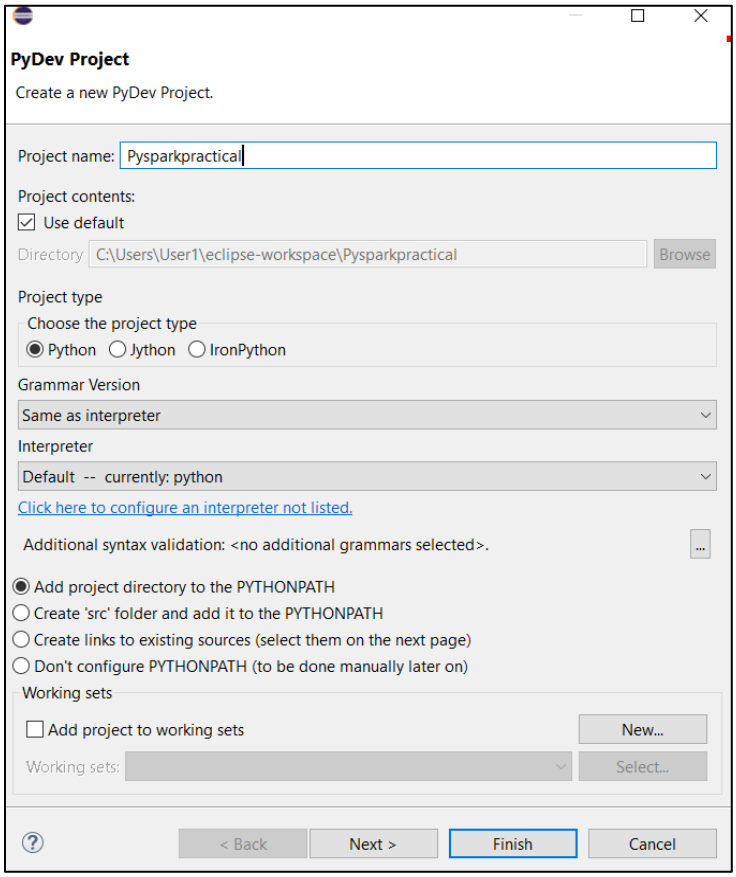
Now configuration steps are done.

Coding and Compilation

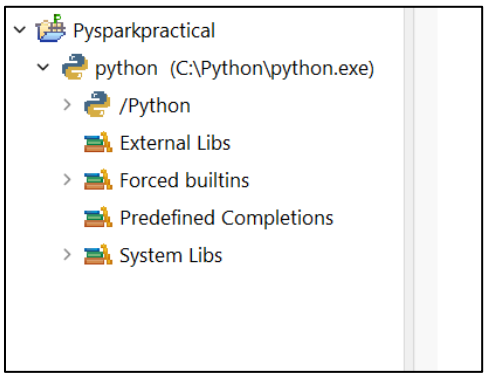
Open a New PyDev Project



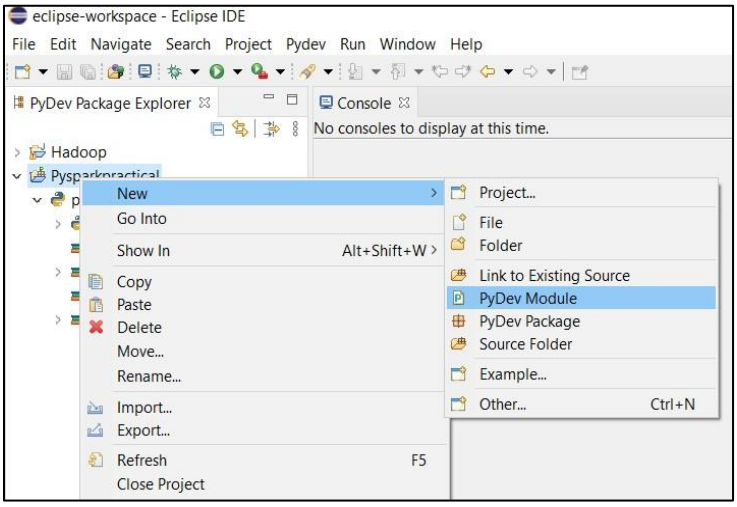
Give it a name and select the grammar version.



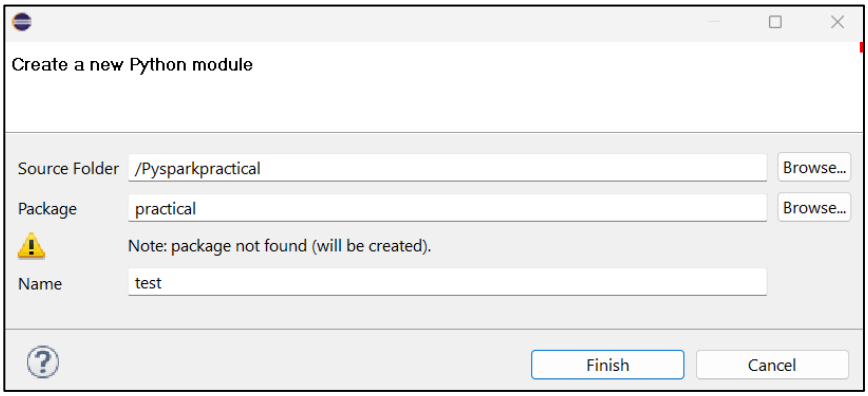
Then Next and Finish. So the project will be created.



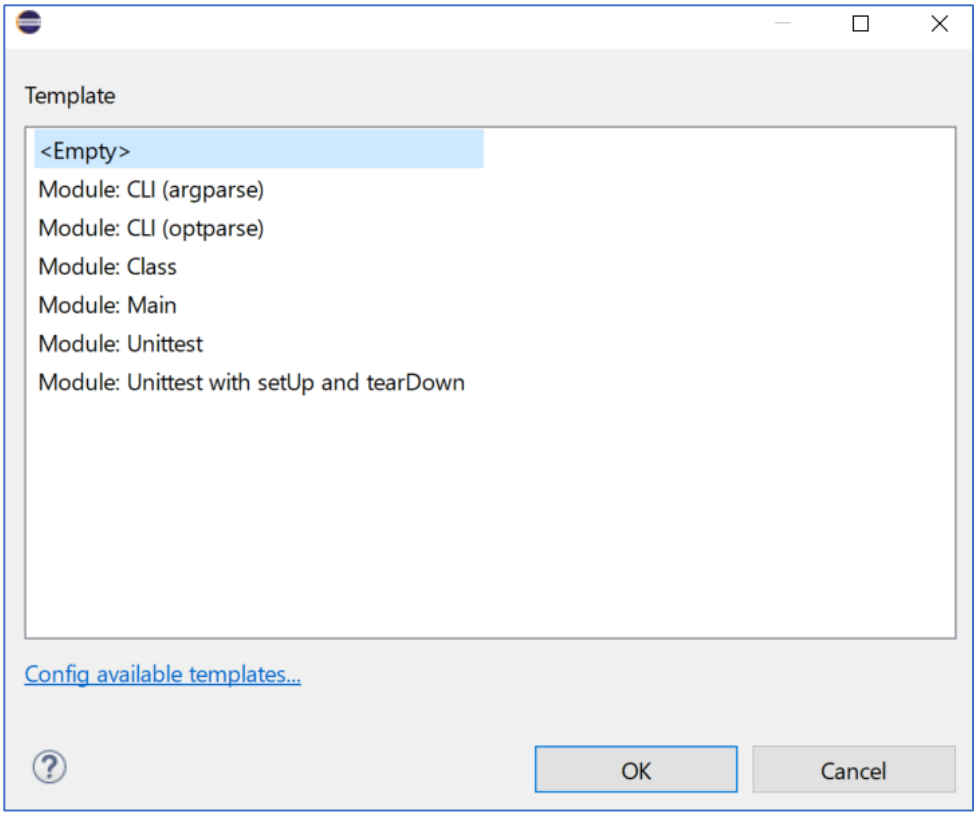
After that open, a PyDev module



Then give package name and a name for the module.



Put this as empty

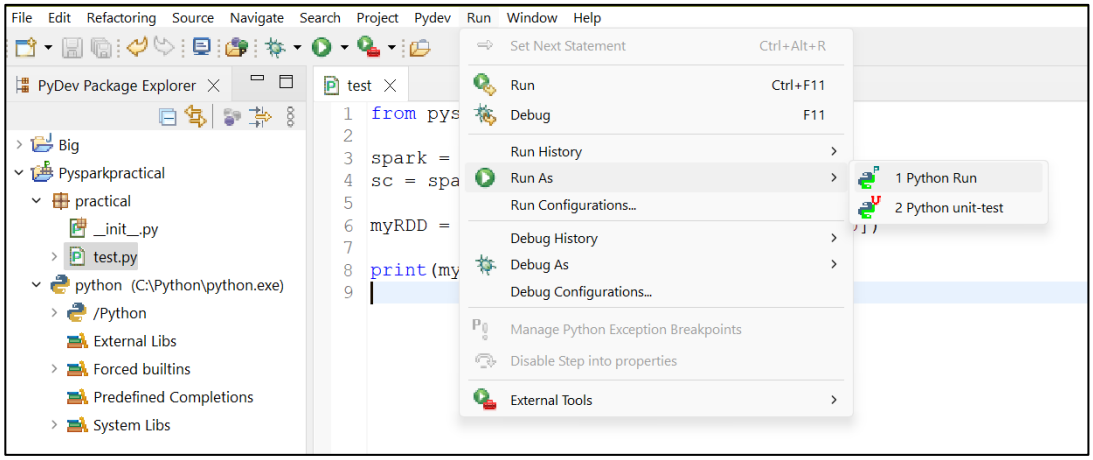


Then type this code

Example 1 – Pop out the numbers from an array

```
from pyspark.sql import SparkSession

import os
import sys

os.environ['PYSPARK_PYTHON'] = sys.executable
os.environ['PYSPARK_DRIVER_PYTHON'] = sys.executable

spark = SparkSession.builder.getOrCreate()
sc = spark.sparkContext

myRDD = sc.parallelize([100,200,300,400,500])

print(myRDD.take(3))
```

To run this file



Without importing os and sys variables, you can't get a output for this code

Output

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

[Stage 0:>                                                          (0 + 1) / 1]



[Stage 1:>                                                          (0 + 4) / 4]

                                                                                [100, 200, 300]
SUCCESS: The process with PID 8096 (child process of PID 12524) has been terminated.
SUCCESS: The process with PID 12524 (child process of PID 2248) has been terminated.
SUCCESS: The process with PID 2248 (child process of PID 9804) has been terminated.
```

Example 2 – Word Count

```
import pyspark

sc = pyspark.SparkContext('local[*]')

txt= sc.textFile('C:/Bigdata/word.txt')
print(txt.count())

python_lines = txt.filter(lambda line: 'apache' in
line.lower())
print(python_lines.count())
```

Output

```
[Stage 0:>                                                  (0 + 2) / 2]

                                                                      108


[Stage 1:>                                                  (0 + 2) / 2]

                                                                      11
```