**Day-1:**

**We created a project in Pycharm**

* Video: number\_string\_function.mp4
* **Created a package named "billing" under the project**
* Learnt the importance of creating a project in virtual environment and project structure
* Used "pip install" command to install and distribute necessary package into site-package of virtual environment
* **Created requirements.txt to maintain the installed package name and version**
* Mentioned the use of "pip install -r requirements.txt"
* Explained the difference between directory & package
* **Created a account.py module under billing package**
* Use of dir, print, help, len
* Learnt the properties of number and string data types using dir
* Introduction on function
* **Created two functions inside account.py module (calculate\_total(), add\_itmes())**
* **Executed these function**
* **Accessed global variable inside the function**
* Write doc string for module and function
* **Imported this module in python shell and verified the dir and help on this module**

**Day-2:**

* Video: tuple\_dict\_loops.mp4
* Identified the scope of billing in this project, so created cart.py and pricing.py modules
* Introduction on *for* loop
* Using debugger option, we learnt the data flow over each iteration
* Learnt the properties of *list* and string data types using dir
* Learnt while loops
* Learnt the use of *pass, continue, break* statements in loops
* Learnt *tuple* data type using dir
* Practiced on *for* loop with *range*, *enumerate*, *list*, *tuple*
* **Moved add\_items() from account.py to cart.py module**
* **Created display\_item() in cart.py**
* **Created main() in cart.py**
* **Main() is designed to call add\_item() and display\_item()**
* Iterated add\_items(*arg*) is called and so global variable items=[] stores each item passed.
* display\_items() is designed to iterate *items* variable and display the added in console
* Learnt dictionary data type properties using dir
* Learnt how to iterate a dictionary

**Day-3:**

* Video: dict\_in\_project.mp4
* Identified that keeping items and updating in billing package is irrelevant and understood the need of store package in this project
* **Created a package with name store**
* **Created store\_data.py main.py, product\_management.py cart.py and pricing.py modules**
* Learnt single & multi comment lines
* main.py is to display the menu list until the user exit
* Learnt displaying the menu option using endless loop
* Learnt how to get input from user and so respective menu choice prints message.
* Learnt if condition and call relevant function for user choice
* Learnt all operators in Python
* **Written below functions into product\_management.py**

update\_product\_details()

validate\_product\_keys()

alert\_product\_unavailability()

* **Added dict type variables: stationary, fruits, grocery data in store\_data.py**
* From python shell, learnt to get price for the product from store\_data.py