ADVANCED PROGRAMMING
ASSIGNMENT 4
README FILE

RUN CODE :
Code is very simple to run. There is only one “RunFile” in the main directory “2023116_Assignment4” that has main function to run the complete project. Other required project files are also included that implements the required functioning for the program system.
The interface of this CLI (+GUI) based project is Menu based and needs menu number as input command to be given on Terminal to choose the required sub-menu to implement functionalities or retrieve stored information.
It is a smooth and simple program, user just need to input the option number from the list of sub-menus shown based on operation he/she wants to perform and enter the required details related to the operation.

CONSIDERATIONS :
1 While running the program please make sure that correct logical inputs are provided (like integer input must not be given alphabets and it must be within the range provided). Please don’t give unnecessary incorrect input.
2 Registration is required for login. You may try login but it won’t happen until you register with same email and password.
3 The program allows only single Admin registration that handles all the Admin functionalites. Once registered, UserName and Password are auto generated (fixed) in case of Admin and need to enter the same to login as Admin. Admin Username=admin@ByteMe, Password=IIITD_ByteMe.
4 For GUI view first run code in CLI view create menu, make orders, all these are stored in file using IO stream. Then terminate the program and launch again but this time in GUI mode. Since complete File IO management is used you can again close the GUI/program relaunch in CLI only register admin again, it will automatically take menu, pending orders, users(customer) data on login so you can just edit, no need to register user again and launch program again in gui to view updates. No thread concept is used. Everything is managed using File IO streams.
5 Use Exit Application option to exit the program.
6 Sequence for input should be practical.

ASSUMPTIONS :
There are no specific assumptions. All the implementation done is based on Assignment guidelines and their real life implications.

Collections Used :
1) Array List : Used to store food items in menu, order history of each customer, pending orders of customers and many other places wherever required.
2) TreeSet : Used to sort food menu in ascending or descending order based on price of food item using custom comparator.
3) HashMap : Used to store food items in cart by mapping food items with their integer quantities.

OOPS concept used :
1) Classes : Different classes are made to handle different types of objects (ex Customer,Admin).
2) Interface : A functional interface(Functionalities) is used to handle operations for different users.
3) Inheritance : There is a common user class inherited by all types of user.
4) Polymorphism : Parameter Polymorphism used on Operations performed by different users.
5) Encapsulation : Complete encapsulation of data and methods for each user, food related things(like menu) and order related details is done.
6) Abstraction : User class is made abstract class.