Name of the Experiment: You are going to develop a module of Chittagong University ERP (Enterprise Resource Planning) solution for opening new departments under 7 faculties: Engineering, Science, Arts, Law, Social science, Business Administration and life science. VC office asked for the name of new departments and then propose the names in Academic Council meeting in VC office for discussion. In this meeting, the departments will be finalized. Note: the business logic of each method includes displaying message.

Introduction: We have to define a class for VC office and 7 faculty classes. And a main class that will integrate all of the classes and simulate the solution for our problem.

Objectives:

- to learn how to use multithread
- to learn how and when multithreads is used.

Analysis: Analysing our problem, we have discovered the following components of our solution:

- a class representing VC office, that will ask for new department names and finally will finalize them
- 7 super classes that extends from another helper abstract class. These classes represent the faculties and they will propose new department names.

- a Main class that will contain the main method which will simulate our solution.

From above analysis, the conceptual class diagram is given below.

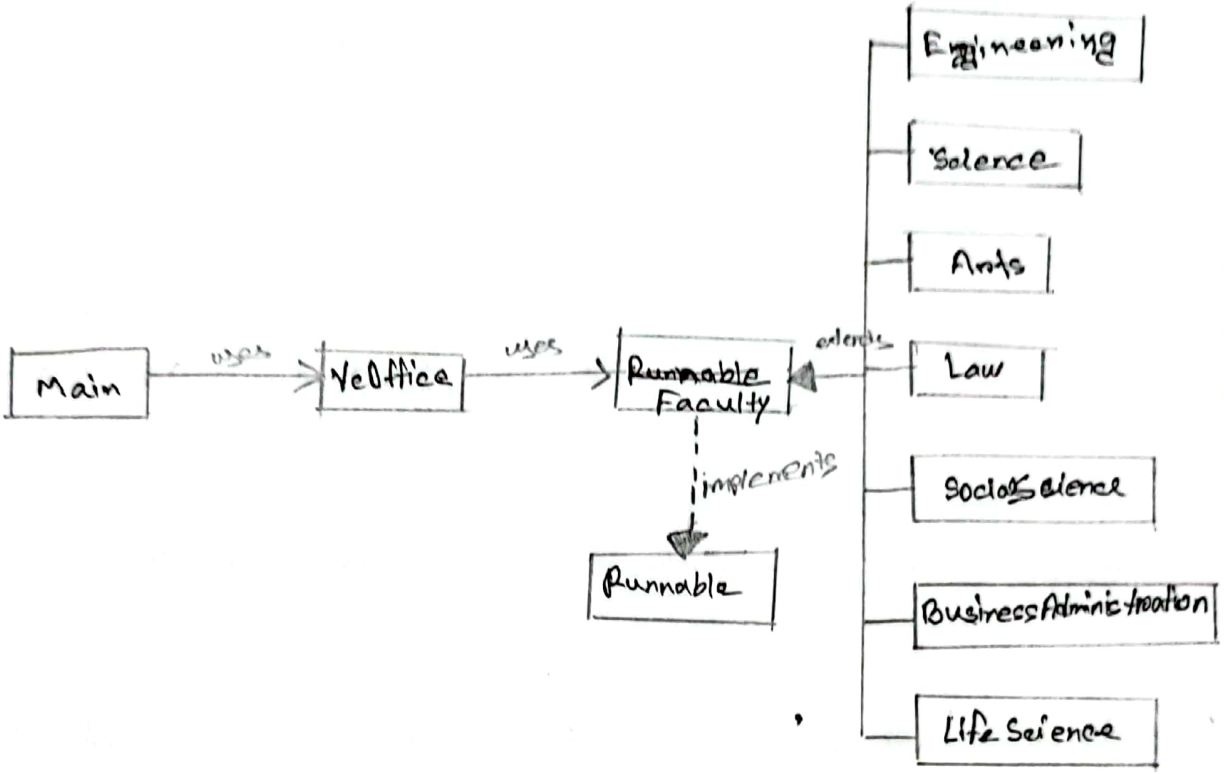

Figure1- conceptual class diagram

Design: From above analysis, the design description of the solution is given below:-

- class VcOffice: the class that represents vc office
  - ☐ Data Members:
    - faculties: an array of Faculty that will store one instance of all the faculty classes
  - ☐ Methods:
    - askForDeptNames: creates 7 threads, invokes each faculty and finalizes all the new dept names.

From above design description the pseudocode for the methods are given below,

VcOffice::askForNewDeptNames():
    creating 4 threads, starts them, join them

VcOffice::finalizeAllNewDepts():
    print the department names are finalized

Faculty::performMeeting():

Faculty::run():
    call the performMeeting() method

Engginearing::performMeeting():
    decide the new department name and propose that to vc office

Science::performMeeting():
    decide the new department name and propose that to vc office

Arts::performMeeting():
    decide the new department name and propose that to vc office

Law::performMeeting():
    decide the new department name and propose that to vc office

BusinessAdministration::performMeeting():
    decide the new department name and propose that to vc office

LifeScience::performMeeting():
    decide the new department name and propose that to vc office

SocialScience::performMeeting():
    decide the new department name and propose that to vc office.

- finalizeAllNewDepts : prints the message that all new dept proposals are approved.

- class
  - Faculty : abstract class that implements Runnable
    - Methods:
      - performMeeting : abstract method that will be implemented by all the 7 faculties
      - run : overridden method from Runnable interface calls to performMeeting method.

From above analysis the architectural class diagram is drawn in figure-2:-

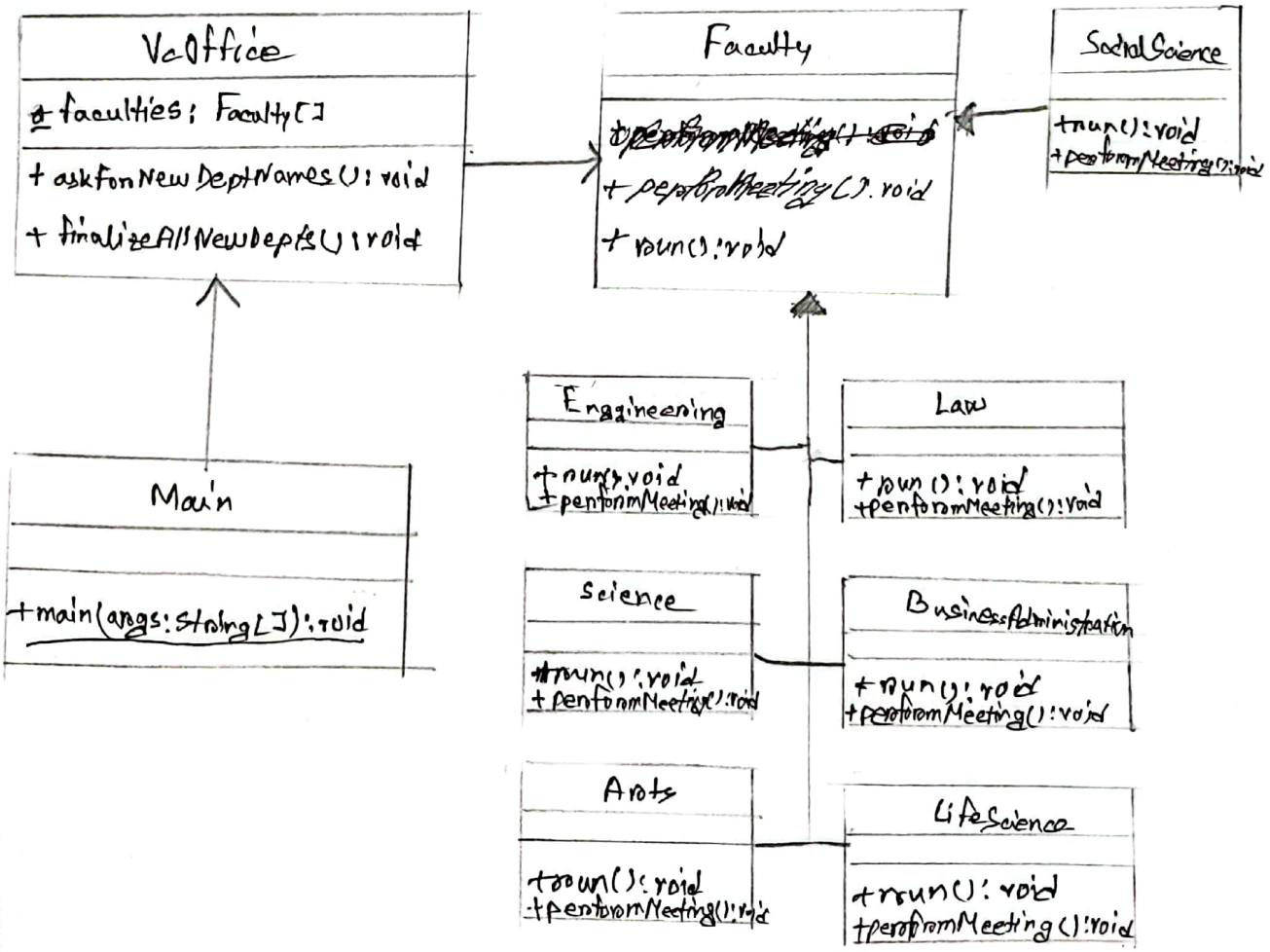

Figure-2: Architectural class diagram

Main:: main(args):

simulate the solution

Conclusion: we defined a vc office class and 7 faculty classes. Then we created Main class where main method simulates the solution.