Q. 2.2 : Define the following terms:

unit testing:

Objectives: • To test the function of unit of Code such as a program or module
• To test internal logic
• To verify internal design
• To test path and Condition Coverage
• To test exception Condition and error handling

When : after mod are Coded.

Input : • internal Application Design
• Master Test plan
• unit Test plan

output : • unit test Report

Who : Developer

Methods : • White box testing techniques

Tools : Debug
• Re - structure
• Code Analyzers
• path / Statement Coverge tools

Education: · Testing methodology
· Effective use of tools.

System Testing :-

Objective: · To verify that the system components perform control function
· To perform inter-system test
· To demostrate that the system perform both functionally and operationaly as specified
· To perform appropriate types of test relating to Transation flow, installation Rolibility, Regression etc.
When: after Integration Testing
Input: · Detailed Requirment and external application Design
· Master Test plan
· System Test plan
output: System Test Report.
Who: Development team and usere
Methods: Problem/ Configiration managment
Tools: · Depends
Education: Testing methodology

Black Box Testing :-

- No knowledge of internal design or code required.
- Tests are based on requirment and functionality
- Not based on any knowledge of internal design or code
- Covers all Combined Part of a System
- Test are data driven (Test are based on putting some data to eneak the System)
- It uncovers:
  - incorrect or missing function
  - interface errors
  - Errors in data structure or external data based access.
  - Performance errors
  - initialization and termination error.

White Box Testing :-

- Based on Knowledge of internal logic an application's Code
Base on Coverage of Code Statement branches Paths, Conditions
Tests are logic driven
it ensures
- All independent Paths within a module have been exercised at list once
- Exercise all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and withen their operational bounds
- Exercise internal data structure to ensure their validity.

Question No 3:-

Part 1:.

Ans:.

In practice. There is not a clear-cut distinction between these types of maintenance, when the System adapt to new environment then add functionlity to take advantage of new environmental features. Software faults are aften exposed ~~becom~~ beause users use the System in unanticipated way. These types of maintenance are are recognized but a different person Sometime give them different names.

Corrective maintenance:-

is universally used to refer to maintenance for fault repair

Adaptive maintenance:-

Sometimes means adapting to new environment and Sometimes means adapting the Software to new requirements.

Perfective maintenance:-

Sometimes means perfecting the Software by implementing new requirements: in other cases it means maintaining the functionality of the System but improving its structure and performance.

Question 3 :-

Part 2 :-

Ans:

System re-engineering

* Re-structuring or re-writing part or all of a legacy system without changing its functionality.
* Applicable where some but not all sub-systems of a larger system require frequent maintenance.
* Re-engineering involves adding effort to make them easier to maintain. The system may be re-structured and redocumented.
* Reduced risk

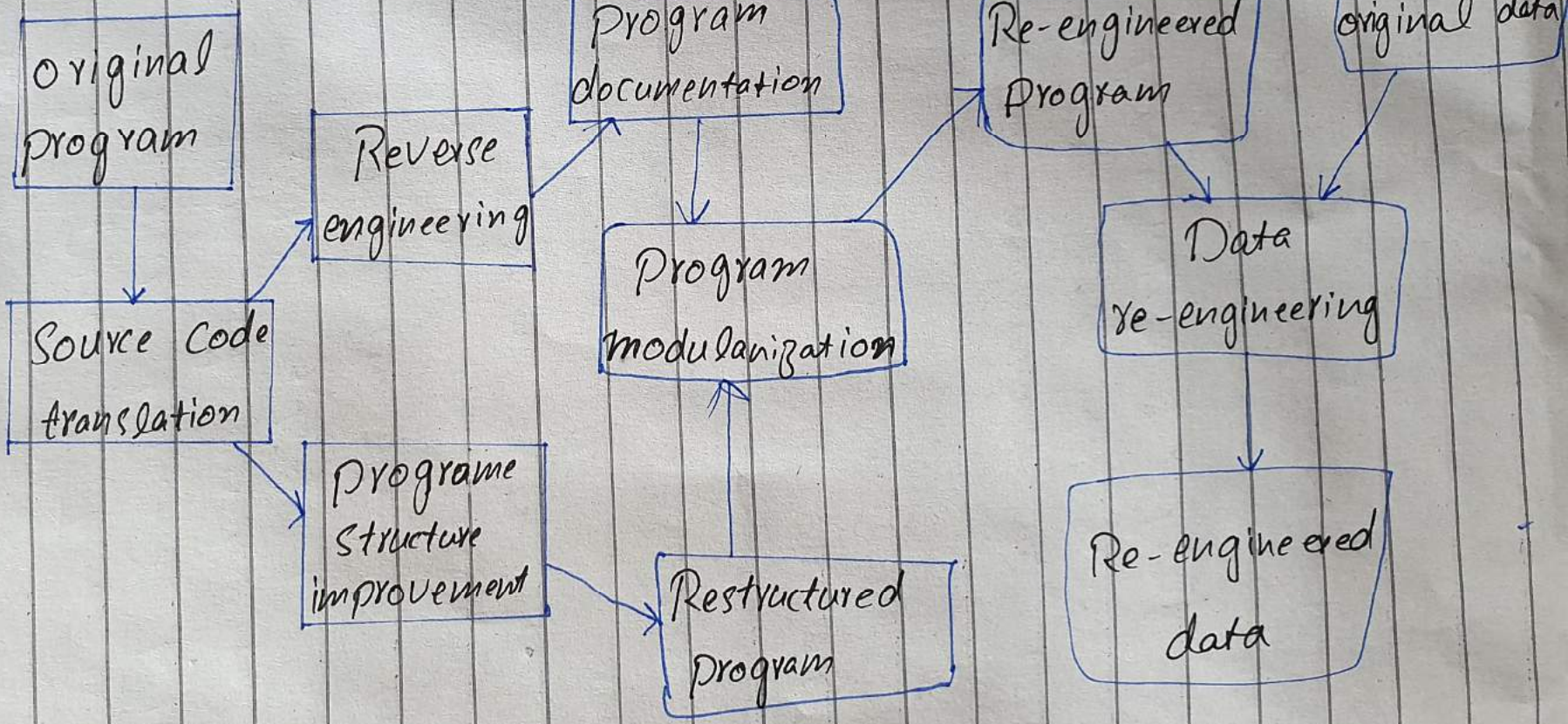
There is a high risk in new software development. There may be development problems, staffing problems and specification problems.

* Reduced cost

The cost of re-engineering is often significantly less than the costs of developing new software.

P# 13

Diagram


original program
Source Code translation
Reverse engineering
Program documentation
Program modularization
Programe Structure improvement
Restructured Program
Re-engineered Program
original data
Data re-engineering
Re-engineered data

Name : Mushtaq Khan

ID : 14927

Section : B

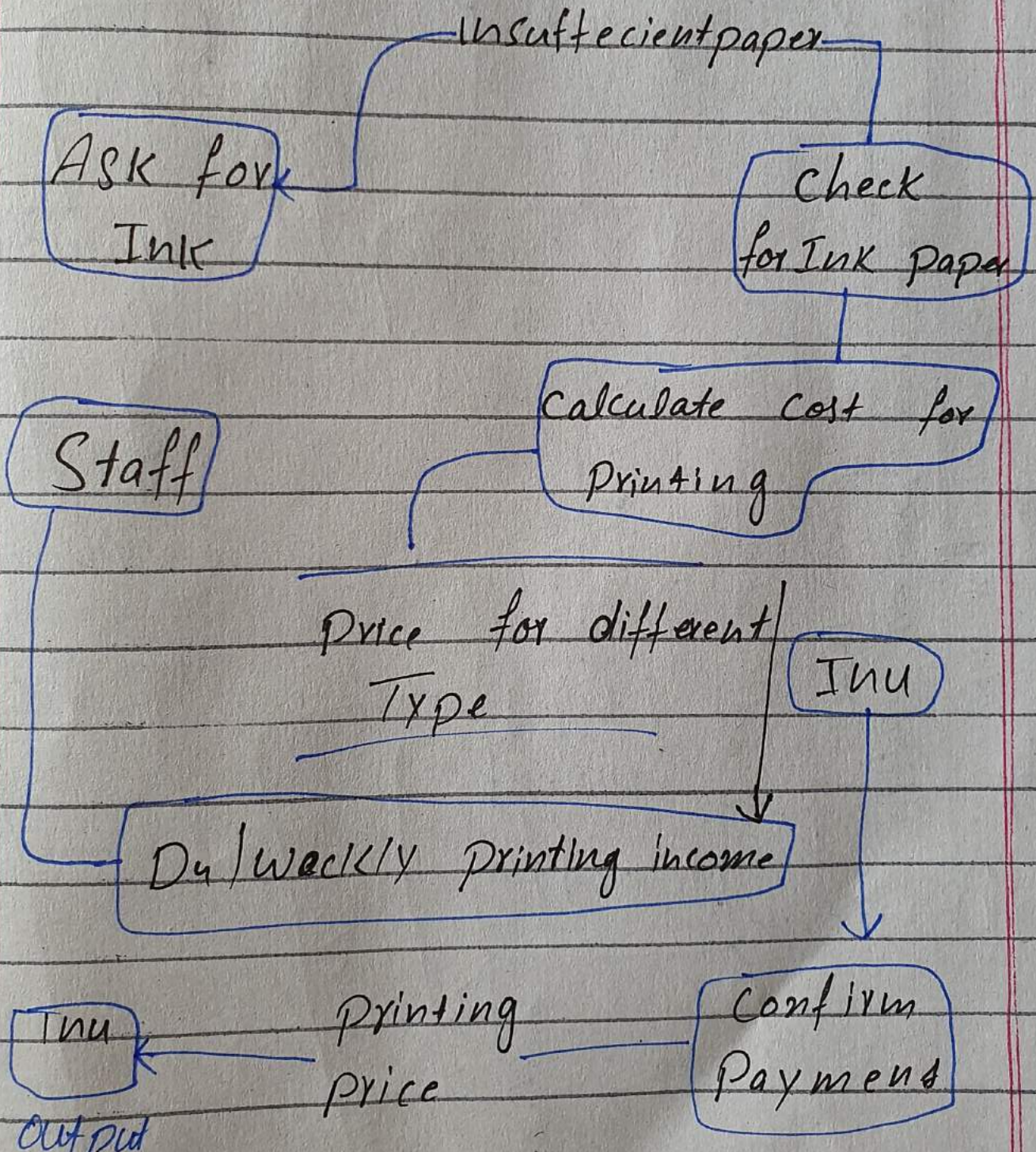
Programe : BS (SE)

Course : Software eng

instructor name: Ghassan hassnain

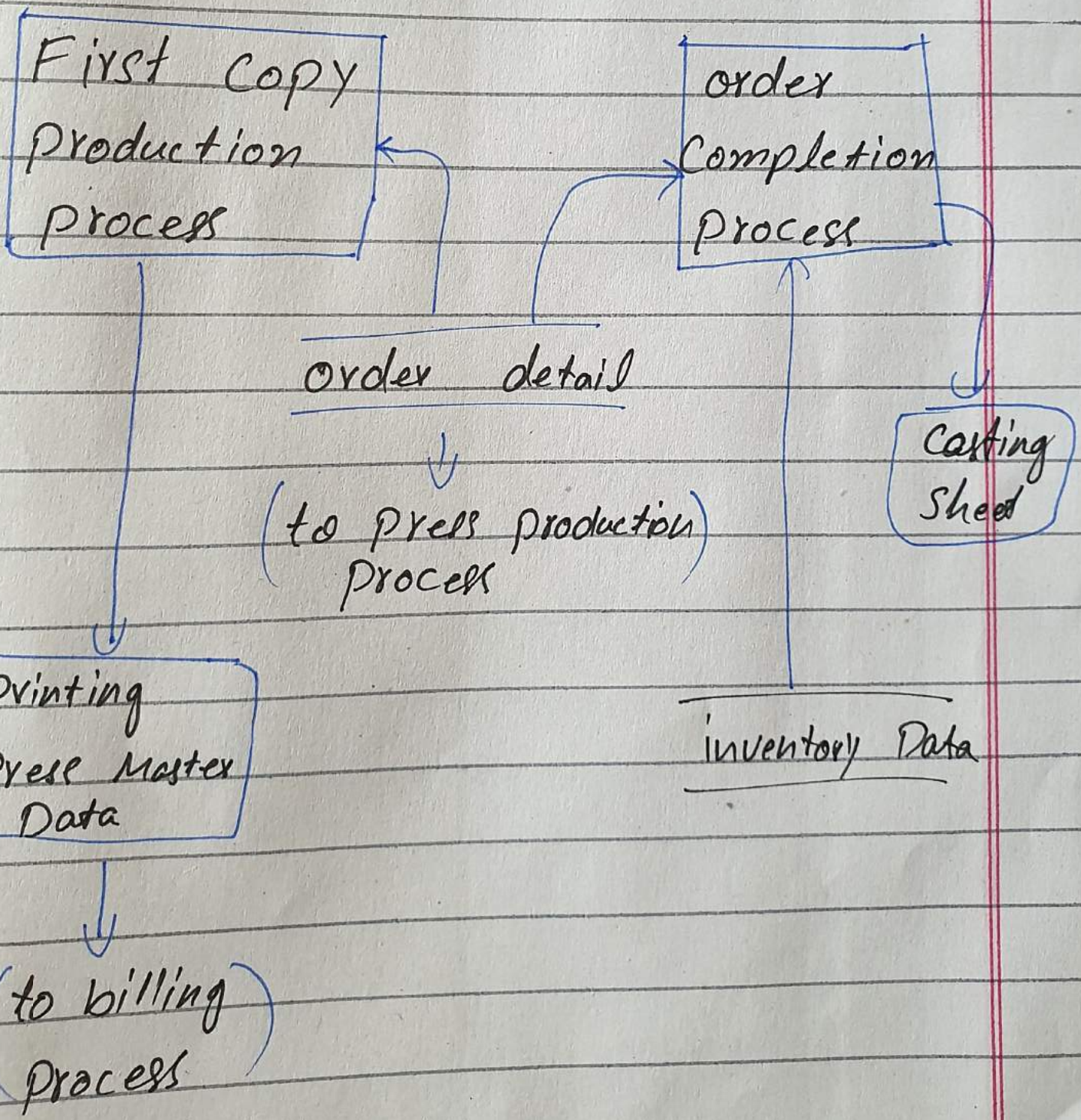
Question No 1 :-

Par 1 :-



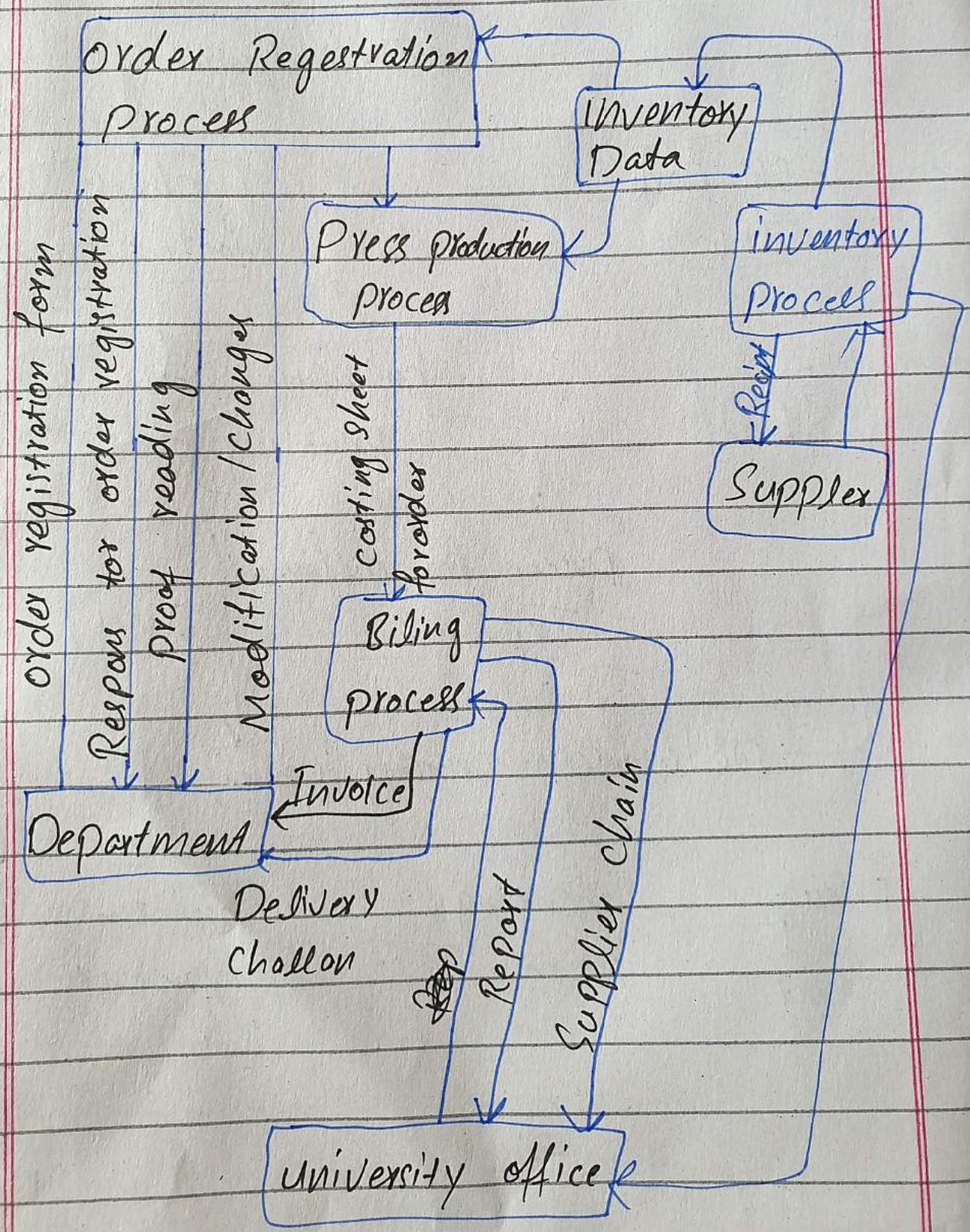
Question No #1:-

Part 2 :-



Question No 1 :-

Part 3 :-



Question No 2

Par 1:. Explain Why testing Can only detect the presence of errors, not their absence?

Ans:. Testing Can detect only the presence of error, not their absence because the main goal of the testing is a part of broader process of software verification and validation. It consists of a set of activities, where the testers try to make the Software behave anomalous in order to detect or ~~other~~ anomaly to be later fix. Testing Cannot demonstrate the faults other than specified in every circumstance. it is always possible that a test have overlook Could discover further problem with the System.

Question No 2

Part 2.2

Unit Testing:-

- The most 'micro' scale of testing
- Testing done on particular function or Code modules
- Requires Knowledge of the internal program design and code
- Done by Programmers (not by testing)