

Name : Hijrat ullah

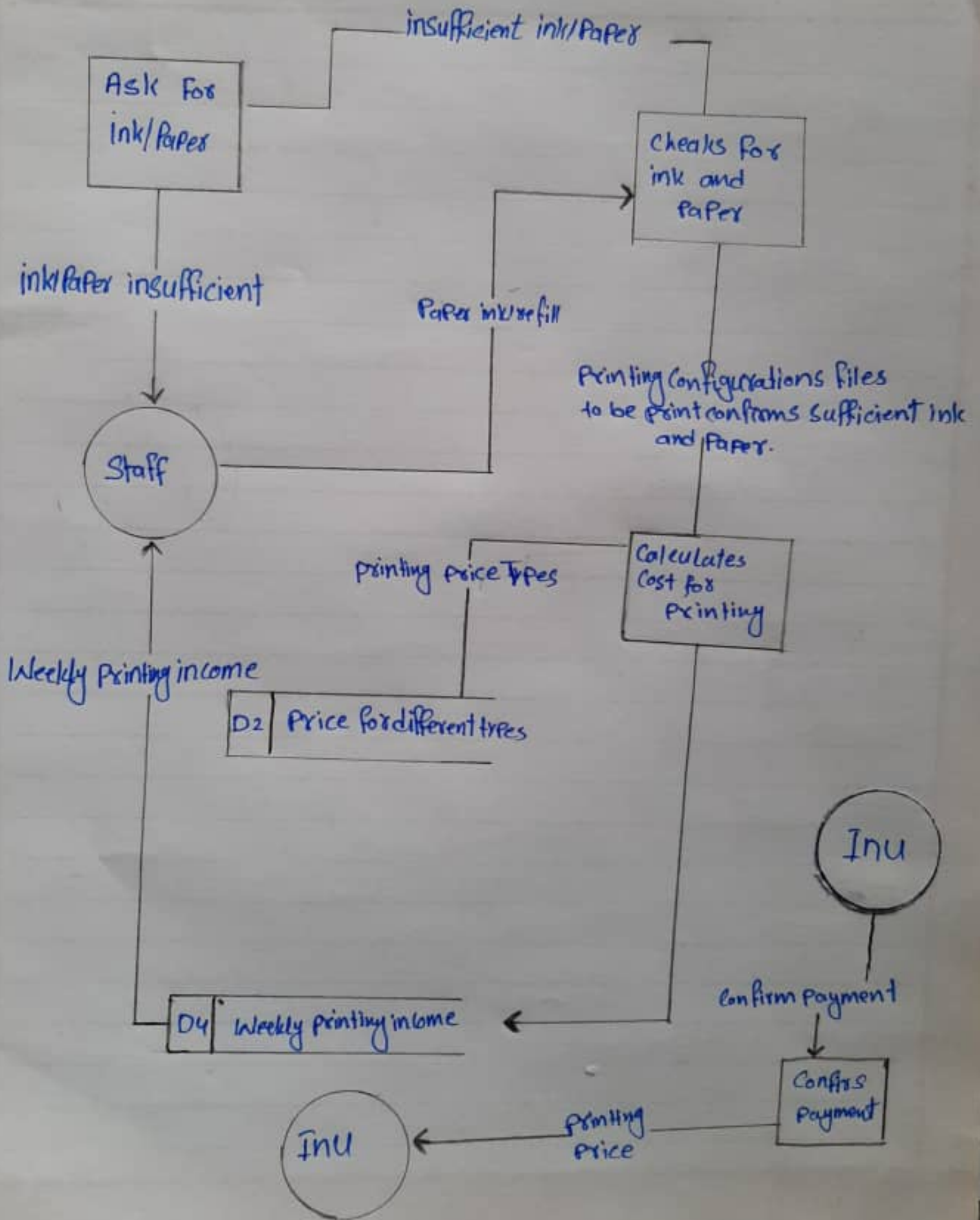
Id : 14994

Paper : Software Engineering

Section : B

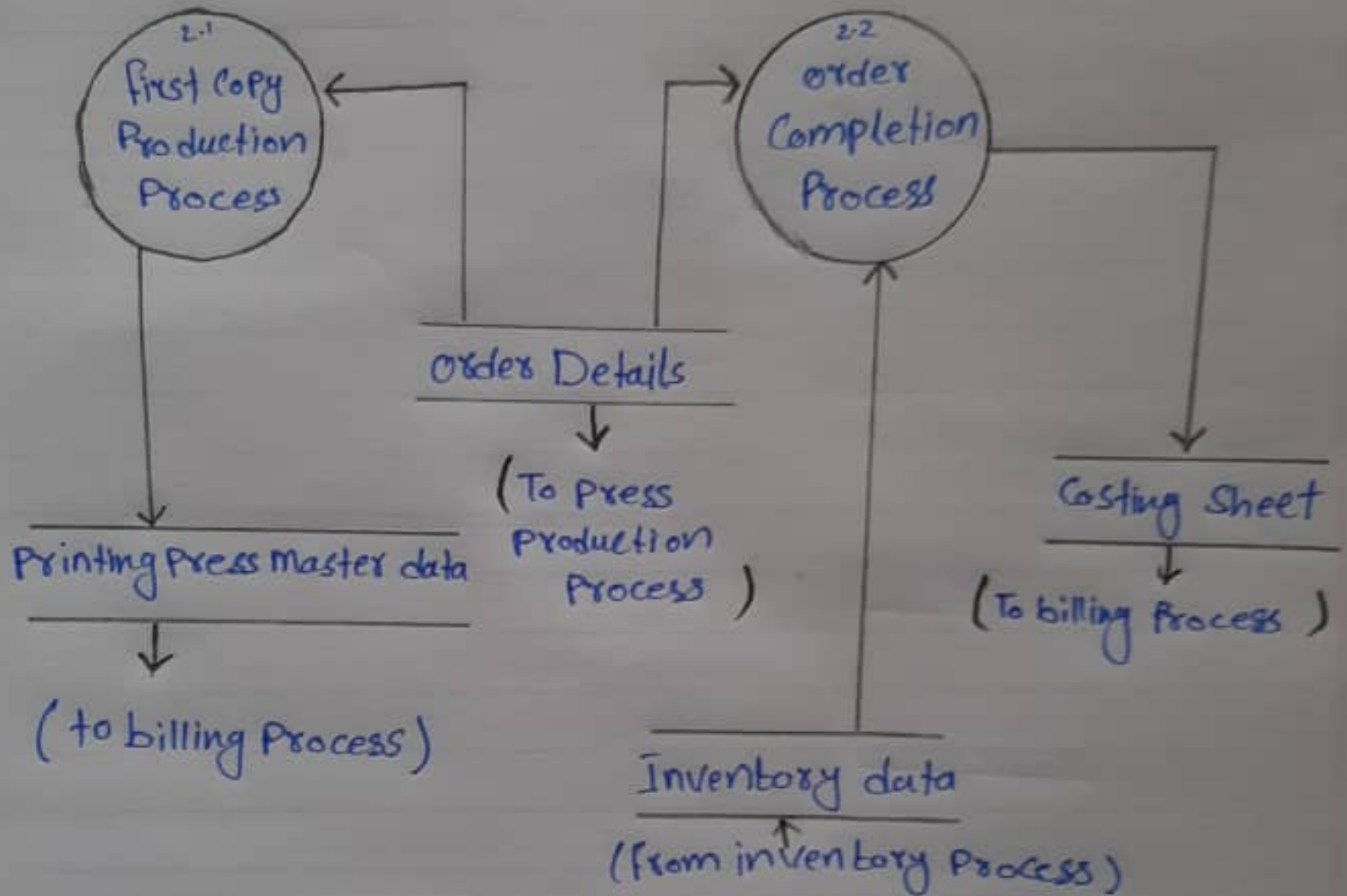
Program : BS SE

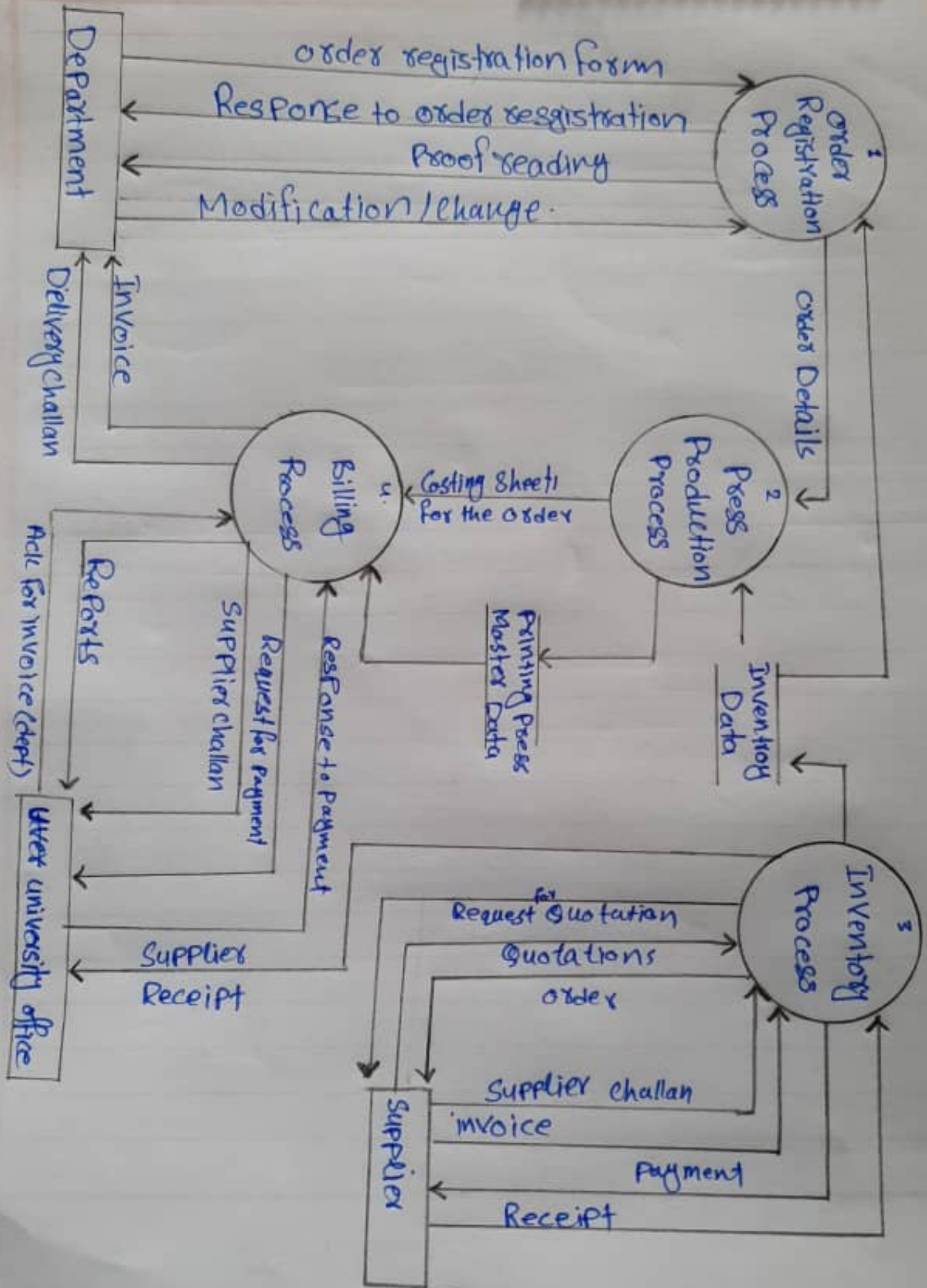
Instructor : Sir Ghassan Husnain.



Question No 1-2
Answer

P-2





Date: _____

Question No 2

Part No 2.1

Answer:

Testing can detect only the presence of errors, not their absence because the main goal of the testing is "To observe the behavior of the particular software and to check whether it meet its requirement expectation or not."

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 anomaly to be later fix. Testing cannot demonstrate the fault other than specified in every circumstance. It is always possible that a test have overlooked could discover further problem with the system.

Date: _____

Question No 2:

Part No 2.2:

Answer:

Unit Testing: =>

- The most 'micro' scale of testing
- Tests done on Particular function or code modules.
- Requires knowledge of the internal program design and code.
- Done by Programmers (not by testers)

Date: _____

unit testing:

- Objectives:**
- To test the function of a program or a 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 modules 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 Coverage tools

- Education:**
- Testing methodology
 - Effective use of tools.

Date: _____

System Testing :->

- Objectives:**
- To verify that the system components perform control functions.
 - To perform inter-system test
 - To demonstrate that the system perform both functionally and operationally as specified
 - To perform appropriate types of test relating to Transaction flow, installation Reliability, Regression etc.

- When:**
- After Integration Testing.

- Input:**
- Detailed Requirement and external Application Design
 - Master Test Plan
 - System Test Plan

- Out Put:**
- System test Report

- Who :**
- Development team and users

- Methods:**
- Problem / Configuration management

- Tools :**
- Depends

- Education:**
- Testing methodology.

Date: _____

Black Box Testing: =>

- No knowledge of internal design or code required.
- Tests are based on requirement and functionality
- Not based on any knowledge of internal design or code
- Covers all combined parts of a system
- Test are data driven (Test are based on putting some data to check the system)
- It uncovers:
 - Incorrect or missing functions
 - Interface errors
 - Errors in data structure or external database access
 - Performance errors
 - Initialization and termination errors.

Date: _____

P-9

White Box Testing :->

- Based on knowledge of internal logic of an application's code.
- Based on coverage of code statement, branches, paths, conditions.
- Tests are logic driven
- It ensures.
 - All independent paths within a module have been exercised at least once
 - Exercise all logical decisions on their true and false sides.
 - Execute all loops at their boundaries and within their operational bounds
 - Exercise internal data structure to ensure their validity.

Date: _____

Question No 3:

Part No 3.1:

Answer:

In Practice, There is not a clear-cut distinction between these types of maintenance, when the system adapt to new environment, then add functionality to take advantage of new environmental feature. Software faults are often exposed because users use the system in unanticipated ways. These types of maintenance 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' sometime means adapting to new environment and sometimes means adapting the software to new requirements.

'Perfective maintenance' sometimes means perfecting the software by implementing perfecting new requirements; in other cases it means maintaining the functionality of the system but improving its structure and performance.

Date: _____

Question No 3

Part No 3.2

Answer:

System re-engineering \Rightarrow

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

Advantages of reengineering:

• Reduced risk:

- There is 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.

Date: _____

Reengineering Cost Factors:

- ★ The quality of the Software to be reengineered
- ★ The tool support available for reengineering.
- ★ The extent of the data conversion which is required.
- ★ The availability of expert staff for reengineering.
 - This can be problem with old systems based on technology that is no longer widely used.

The reengineering Process:

P-13

