FINAL TERM EXAME

NAME: FAIZULLAH KHAN

ID: 14840

SECTION: B

PAPER: SOFTWARE ENGINEERING

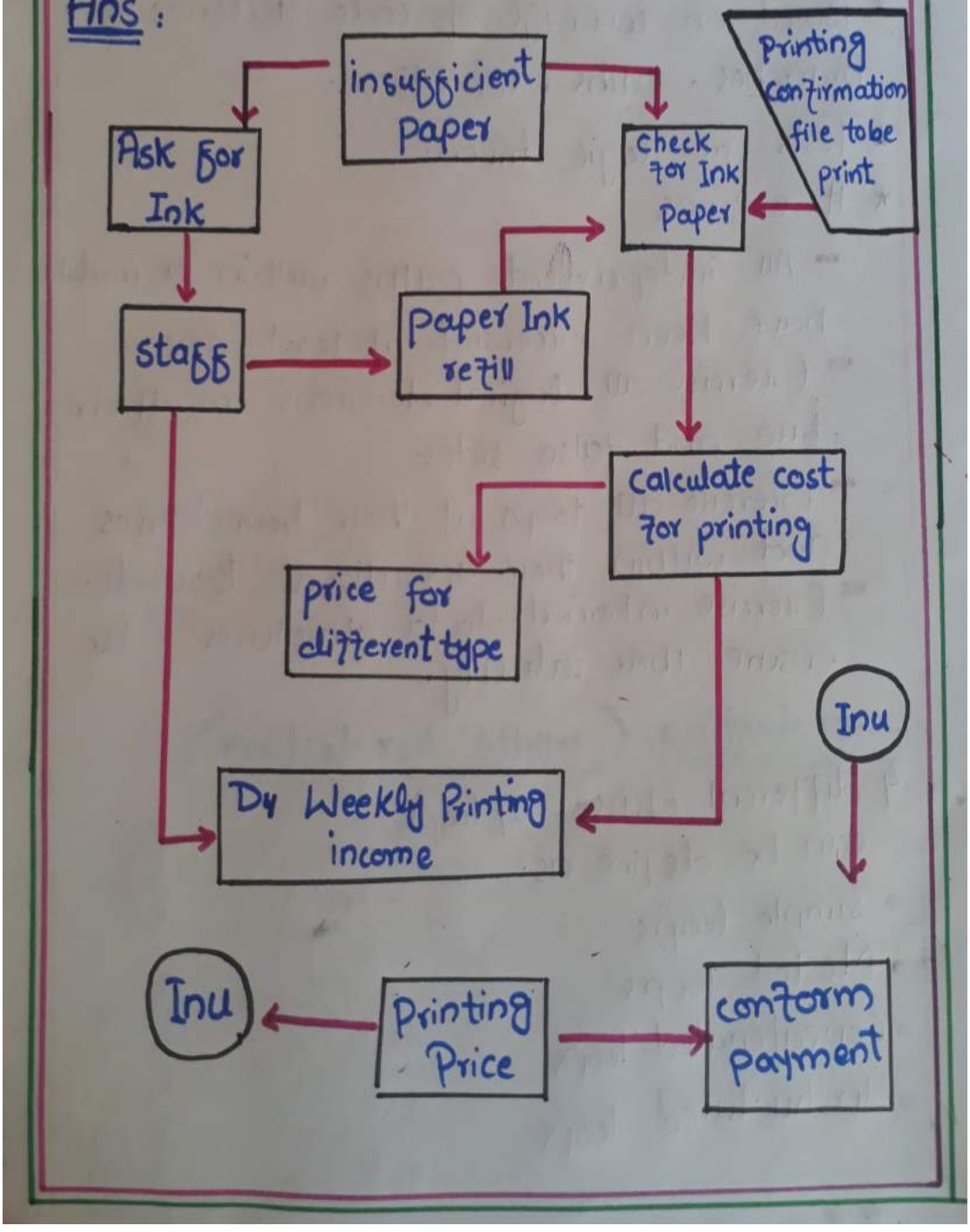
DEPRTEMENT: BS(SE)

DATE-22-6-2020

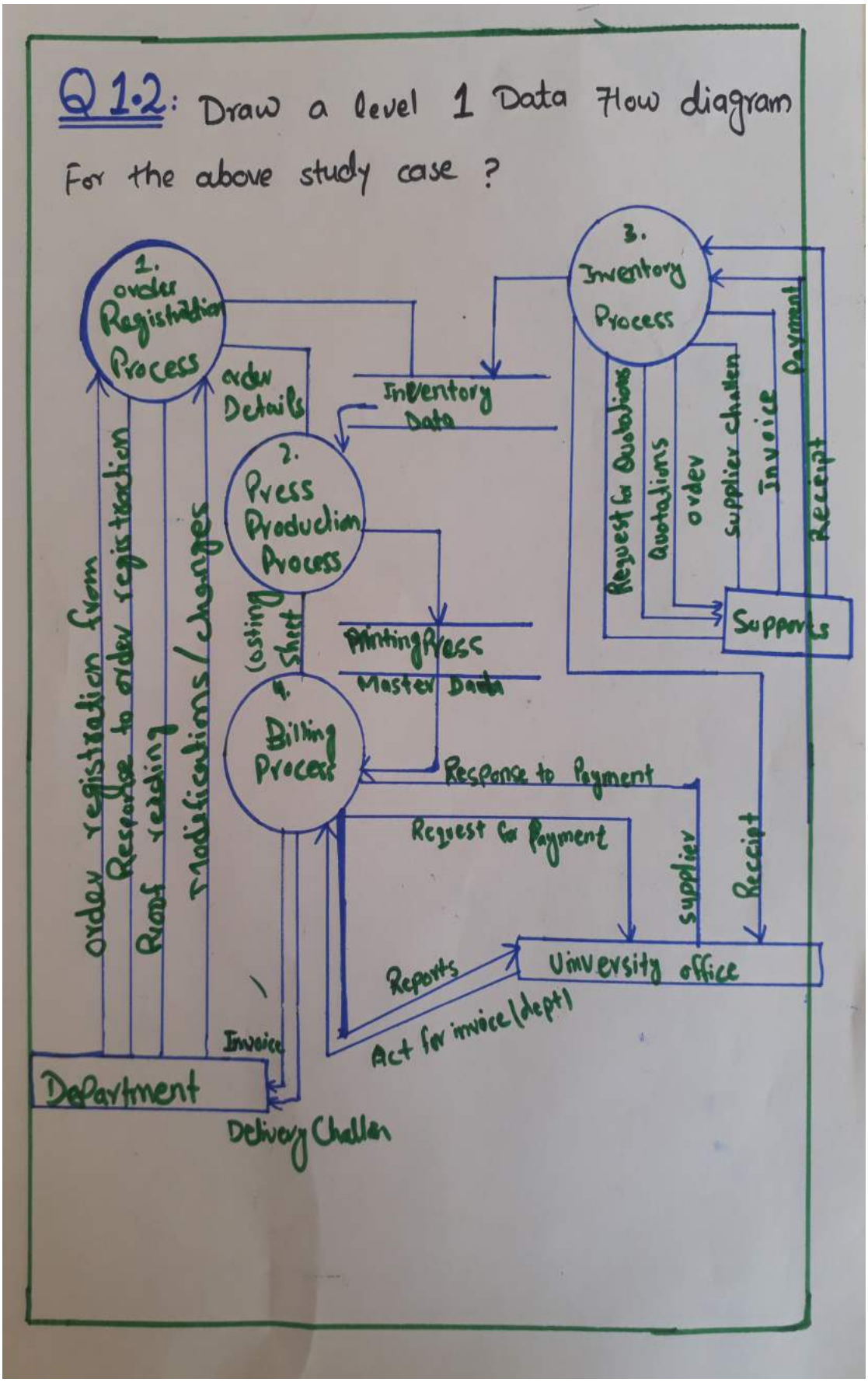
Question :01

Q:1.1 :- Draw a context diagram for INU printing press ?

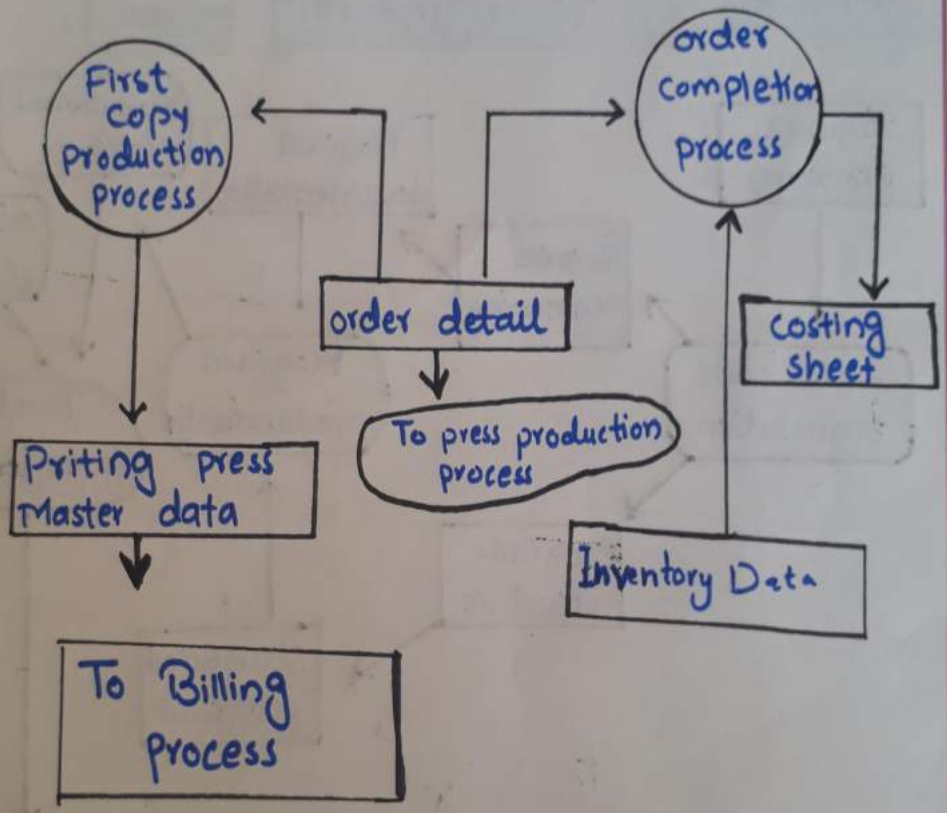
Ans :



Q 1.2: Draw a level 1 Data flow diagram For the above study case ?



Q 1.3: Draw a level 2 (DFD) for the order registration process, press production process, inventory process, and Billing process ?



Question 2:

Q2.1: Explain why testing can only detect the presence of error, not their absence ?

Answer: The goal of software is to note the software behaviour to meet it's obligation exceptation. Testing is a set of actions where the analyzer attempt to make the software to behave on irregular so as to recognize a fault or anomaly to be later fix.

★★ ★★

Q 2.2: Define the following term.

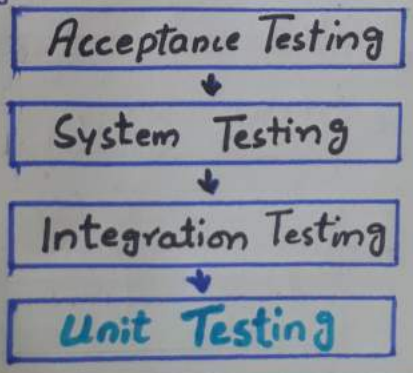
(1) Unit Testing.
(2) System Testing.
(3) Black Box Testing.
(4) White Box Testing.

Ans:

Unit Testing:

* A unit is the smallest testable part of any software.
* It usually has one or few inputs and usually a single output.
* Test done on particular function or code modules.
* Requires knowledge of the internal program design and code.
* Done by programmers (not by testers)

Diagram:



2 System Testing:

System testing is a level of software testing where a complete and integrated software is tested.
The purpose of thise test is to evaluate the system's complainse with the specified requirements.

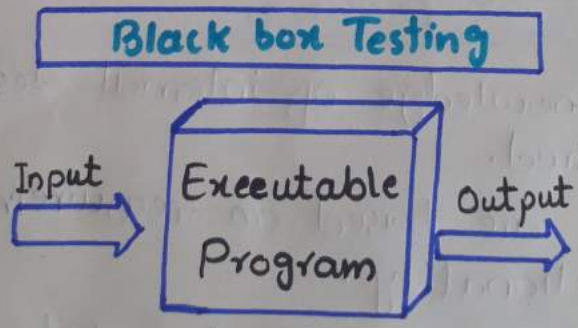
Diagram:

3 Black Box Testing:

* No knowledge of internal design or code required.
* Tests are based on requirements and functionality.
* Not based on any knowledge of internal design or code.
* Covers all ~~data~~ combined parts of a system.

* Tests are data driven (Tests are based on putting some data to check the system).
* This method is named so because the software program, in the eyes of the tester, is like a black box, inside which one cannot see. This method attempts to find errors in the following categories.
  - incorrect or missing functions.
  - interface error.
  - Errors in data structure or external database acces.
  - Behavior or performance errors.
  - Initialization and termination errors.

Diagram.



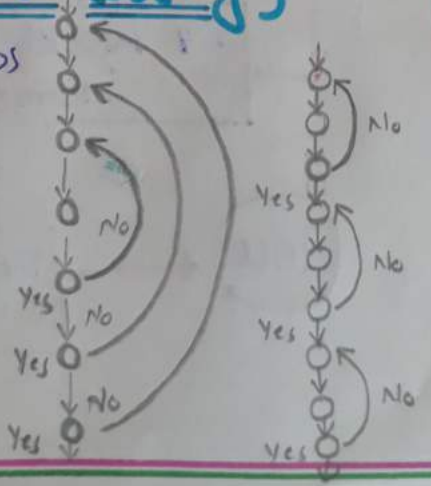
4 White Box Testing:

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

Loop testing (White box testing)

⇒ 4 different classes of loops can be define as.

- Simple loops
- Nested loops
- Concatenated loops
- Unstructured loops



Question : 3

Q 3.1: Briefly describe the three main types of software maintenance. why is it sometime difficult to distinguish between them?

Answer: The main types of software maintenance are as follow.

(1) Bug Fixing: This is fixing issue found in the software after it has been launched. The bugs are there conceivably because testing was not through as it should have been or clients have exposed bugs by using the software in unexpected manner. Coding blunders, design mistakes, and requirements error are the least, center, and most expensive to address, respectively.

(2) Environment adaptation:

When The hardware or platform that the system was worked to run on changes at that point the software must change as

well in order to be compatible and avoid being absolete.

(3) Functionality addition:

Software must be refreshed or changed so it confirm with any new requirements.

These types of maintenance are recognized but a different person sometimes gives them different names.

'Corrective maintenance' is universally used to refer to maintenance for fault repair'

"Adaptive maintenance' sometime means adapting to new enviroment and sometimes means adapting the software to new requirements.

'Perfective maintenance' sometime means perfecting the software by implementing new requirements; in other cases it means maintaning the functionality of the system but improving it's structure and performance

★★ ★★

Q 3.2 :- What are the principal factors that affect the costs of system reengineering? Also briefly explain the reengineering process with the help of diagram ?

Answer:

Reengineering Cost Factors

* The quality of the software to be reengineered.
* The tool support avaiable for reengineering
* The extent of the data conversion which is required.
* The avaiability of expert staff for reengineering.
* This can be problem with old system based on techonology that is no longer widely used.

System re-engineering

- Re-structuring or re-writing part or all of a legacy system plus changing it's functionality according to new requirements.
- Applicable: where some but not all

sub-systems of a larger system requires frequent maintenance.

- Re-engineering involves adding effort to make them easier to maintain. The system may be re-structured and re-documented.

Advantages of re-engineering

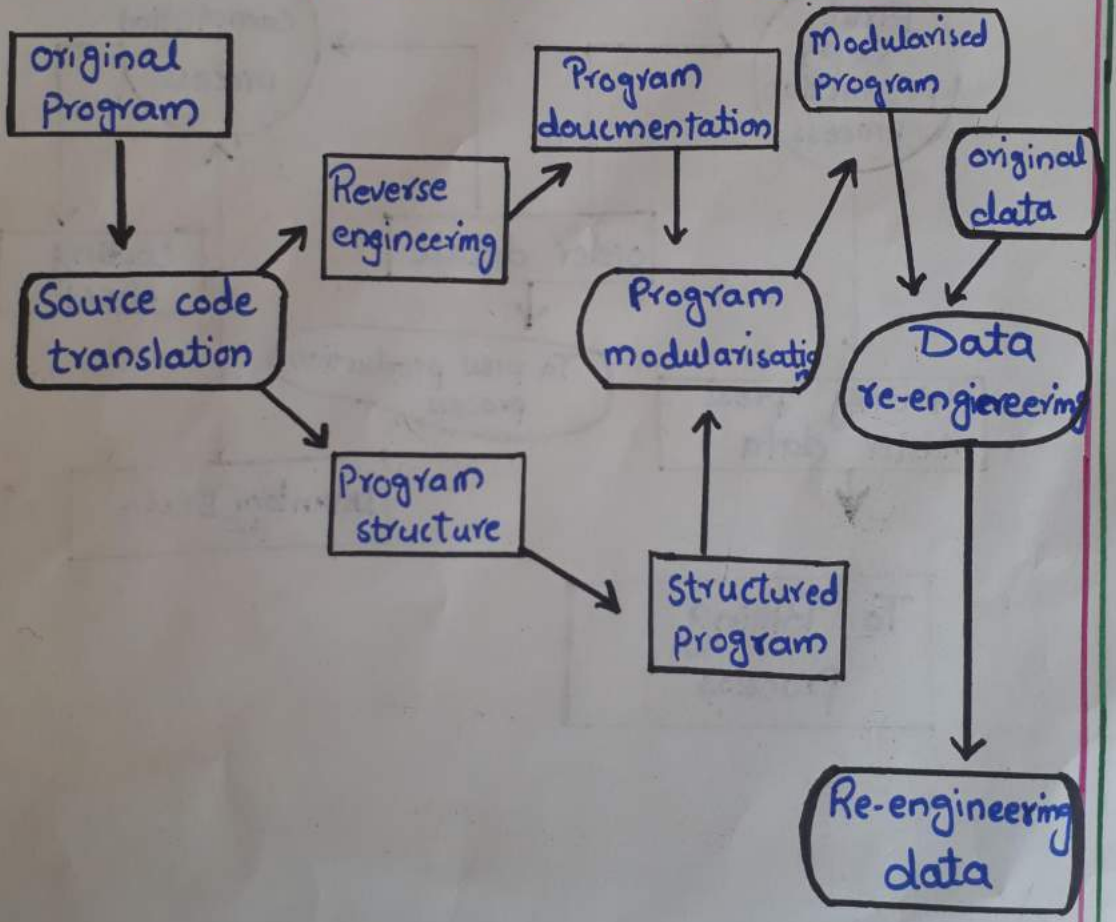
* **Reduced risk:** There is a high risk in new software development. There may be development problems, staffing problems and specification problems.
* **Reduced Cost:** The cost of re-engineering is often significantly less than the costs of developing new software.

Re-engineering Process activities:

- Source code translation.
  * Convert code to a new language.
- Reverse engineering.
  * Analyse the program to understand it.
- Program structure improvement.
  * Restructure automatically for understandibility.

- Program modularisation.
  - * Reorganise the program structure.
- Data reengineering
  - * Clean up and restructure system data.

Diagram of re-engineering Process:



★★ ★★

END OF PAPER