

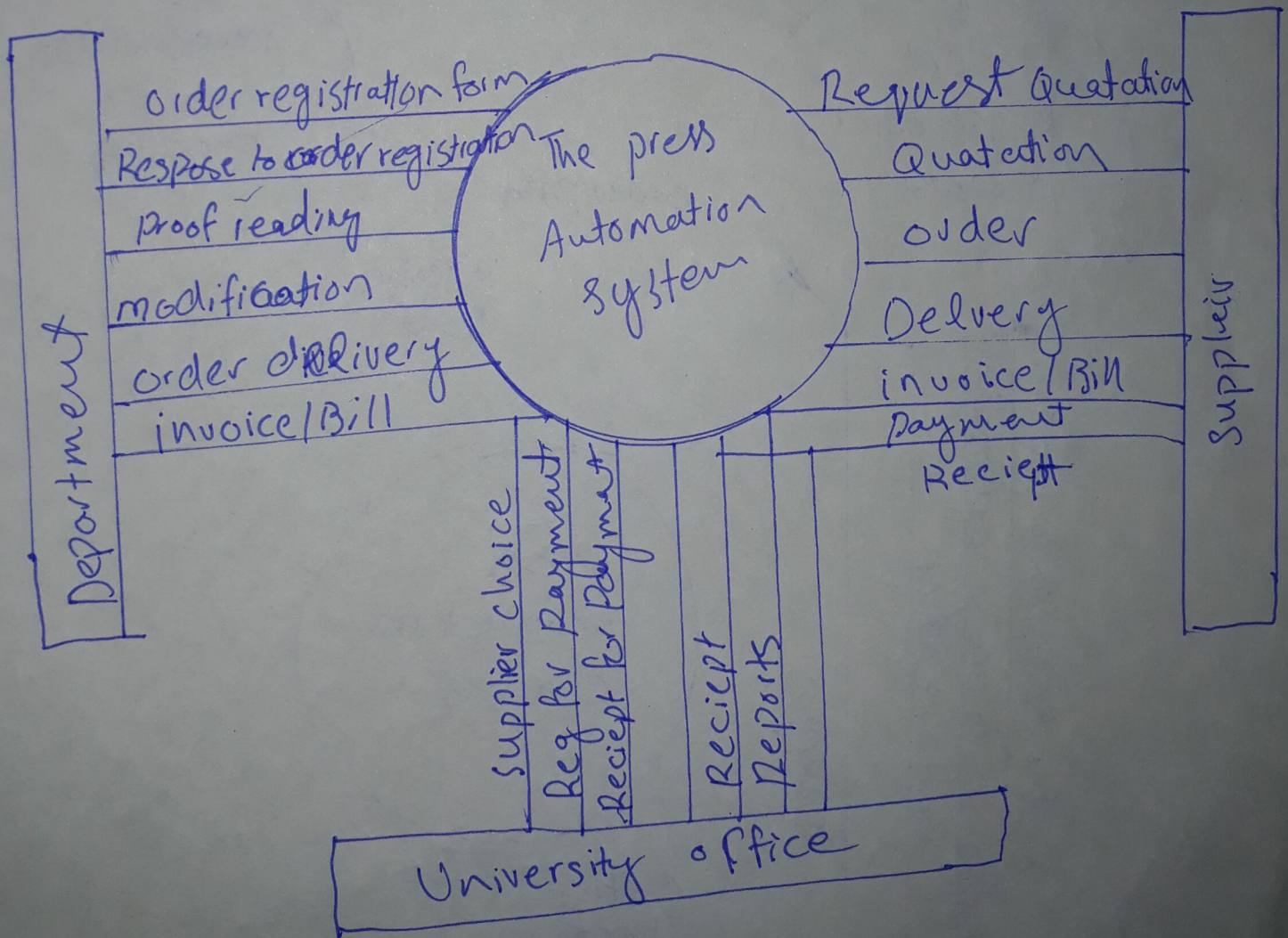
Shiraz Khan Bangash

ID: 14539

Program: BSCS (Software engineering)

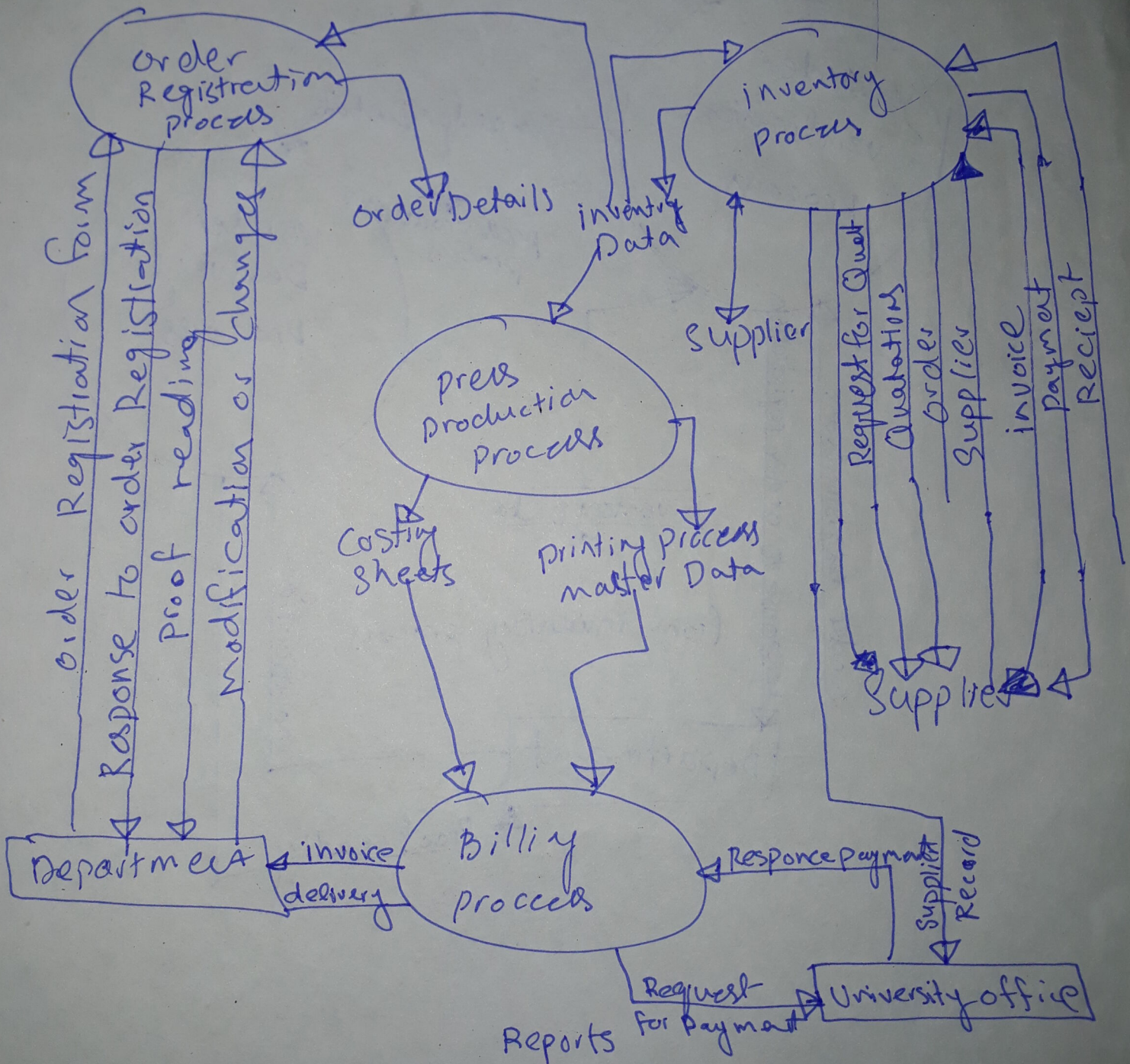
①

1.1 Content diagram for IMU Printing Press





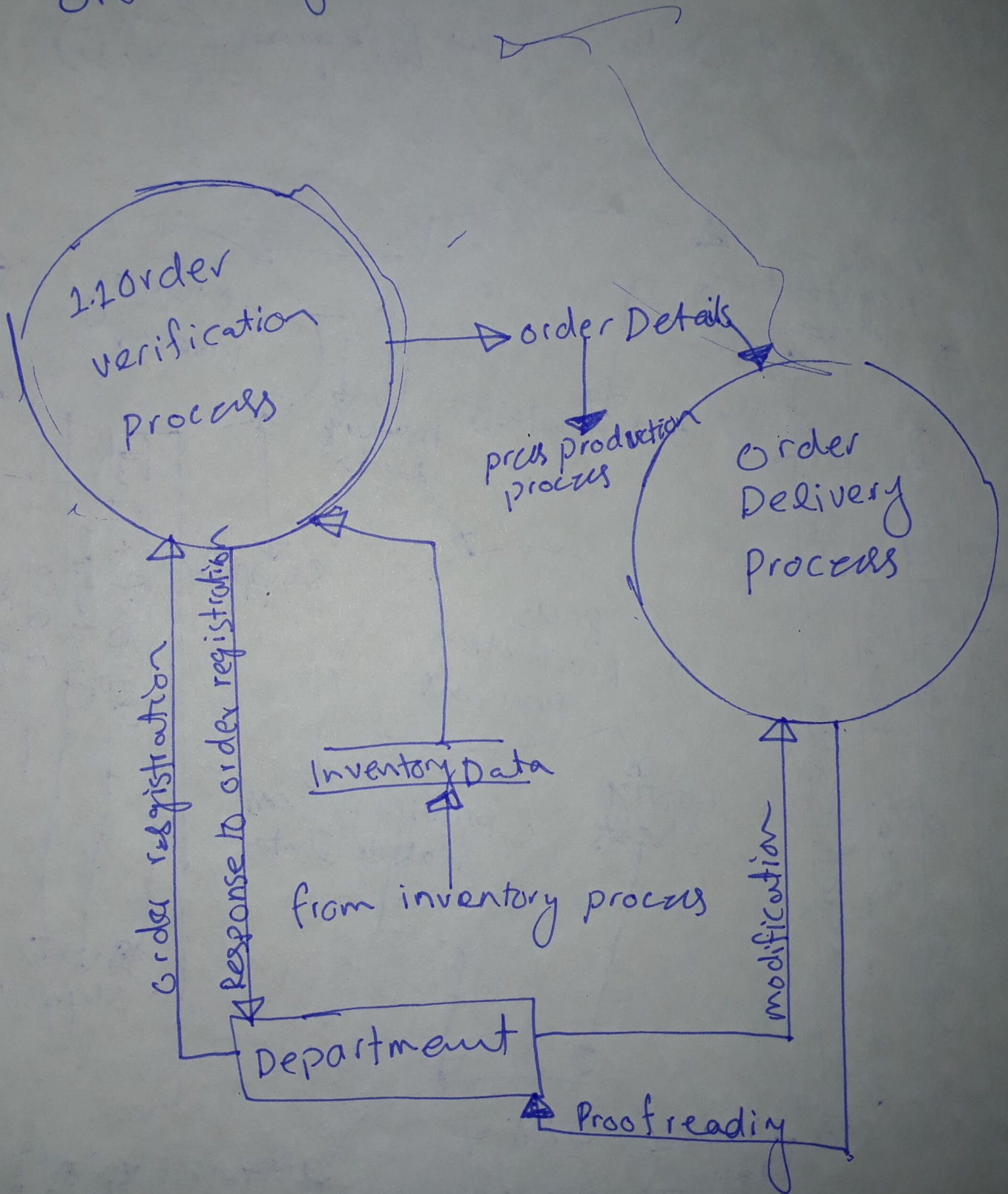
# Q12 level 1 Data Flow Diagram (DFD)





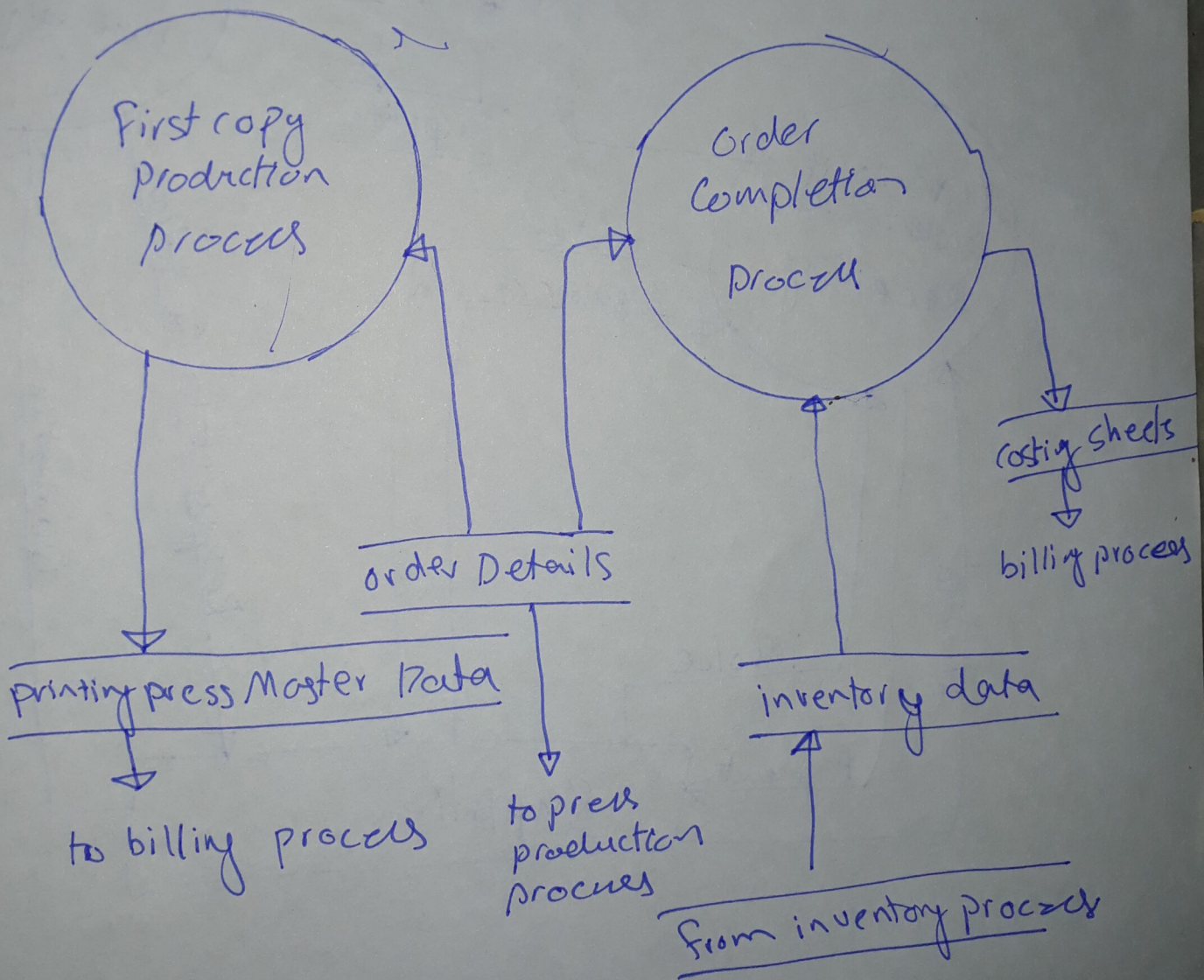
Q 1.3

# Order Registration Process





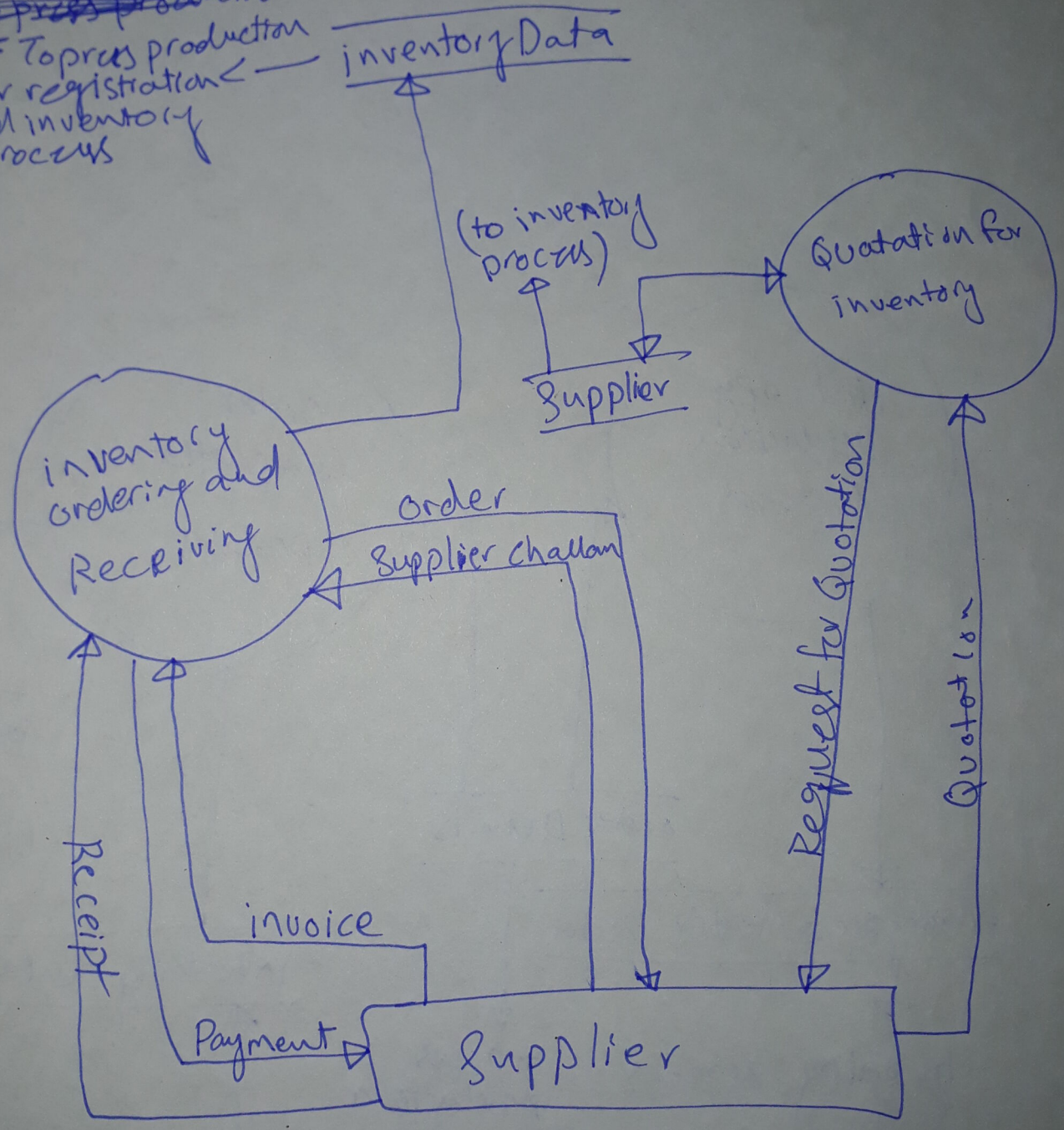
Second level DFD for Press Production process





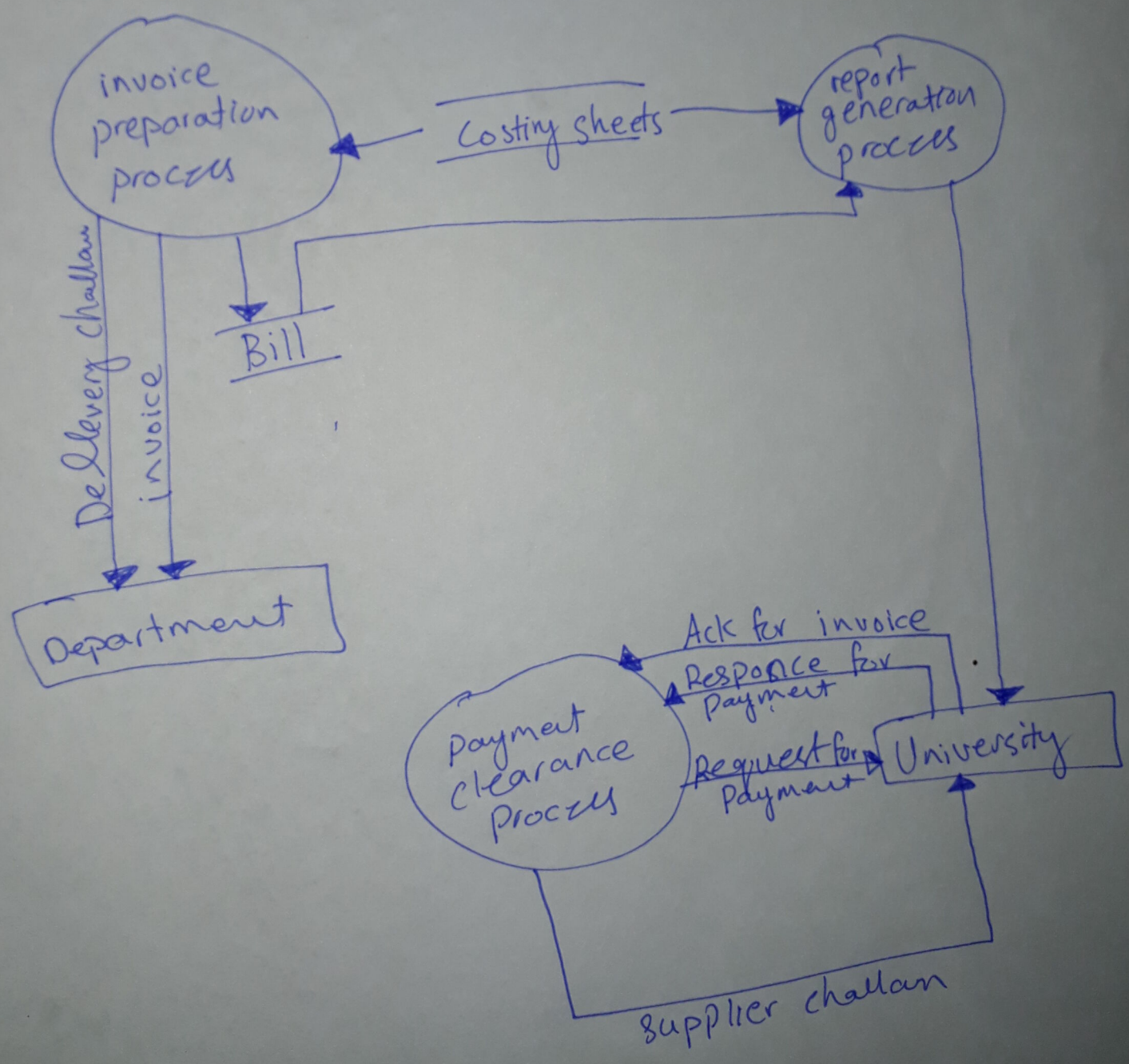
# Second level DFD for inventory process

~~to press production~~  
order to press production  
and inventory  
process





# Second level DFD for Billing Process





Q2.1

A:- Test can detect only presence of errors not their absence because the main goal of the testing is to observe of the behavior of the particular software and to check whether it meets 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 anomalously in order to detect an anomaly to be later fixed. Testing cannot demonstrate the fault other than specified in every circumstance.

Q2.2

A Unit Testing

Unit Testing is a level of software testing where individual units/components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.



## BLACK BOX Testing

Black box testing is a testing strategy that ignores the internal mechanism of a system or component and focuses solely on output generated in response to selected inputs and execution conditions.

Testing technique in which functionality of the application under test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal path of software.

## White box Testing:-

White box testing is a testing of a software solution internal structure, design and coding. The term "white box" was used because of the see-through box concept. The clear box or white box name symbolizes the ability to see through the software's outer shell or box into its inner working.



Q3.1

9

## Three Types of software maintenance

**Fault repairs:** Coding errors are usually relatively cheap to correct, design errors are more expensive as they may involve rewriting several program components. Requirement ~~are~~ errors are the most expensive to repair because at the expensive system redesign which be necessary.

### Environmental adaptation:-

This type of maintenance is required when some aspect of the system's environment. Such as the hardware, the platform operating system, or other support software changes the application system must be modified to adapt it to cope with these environmental changes.

### Functionality addition:-

This type of maintenance is necessary when the system requirements change in response to organizational or business changes. The scale of the change required to the software is often much greater than for the other types of maintenance.

### difficult to differentiate:-

In practice, there is not a clear-cut distinction between these types of maintenance when the system adapts to new environment, then add functionality to take advantage of new environment features. Software faults are often exposed because users use the system in unanticipated ways. These types of maintenance are recognized but a different person sometimes gives them different names.



Q3.2

The cost of re-engineering depends on the extent of the work that is carried out. Cost increases from left to right so that source code translation is the cheapest option and re-engineering as part of the architectural migration is the most expensive.

Factors that affect re-engineering costs are:

1. The quality of software to be re-engineered. The lower the quality of the software and its associated documentation, if any, the higher the re-engineering costs.

The tools support available for re-engineering. The use of CASE tools to automate most of the program changes is normally cost effective to re-engineer a software.

The extent of data conversion required: if re-engineering requires large volumes of data to be converted, this significantly increases the process cost.

Re-engineering process:

The input to the process is a legacy program and the output is the modularized version of the same program. As the same time as program re-engineering, the data for the system may also be re-engineered. The activities in this re-engineering process are



Source code, Translation, Reverse Engineering, Program structure Improvement, Program Modularization, Data Re-engineering.  
The Re-engineering process diagram.

