

## Assing1. Q4.

### FORM – DRIVEN METHODOLOGY:

HIPO is a forms-driven technique. Here standard documents are used to document the information. It consists of a hierarchy chart and an associated set of input/process/output charts. HIPO uses top-down decomposition. It describes the data input and output from processes and defines the data flow composition. The objectives are

1. Provide a structure by which the functions of a system can be understood.
2. State the functions to be performed by the program rather than specifying the program statements to be used to perform the functions.
3. Provide a visual description of input to be used and output to be produced for each level of the diagram.

HIPO makes the transformation of input to output data visible.

HIPO uses easy to draw symbols. The procedure to construct the HIPO chart is as follows

1. Begin at the highest level of abstraction.
2. Identify the processing steps that convert the input to the output.
3. Document each element using the HIPO diagram notation and a treelike structure.
4. Identify sub processes and their inputs and outputs. Continue decomposition until the processes cannot be decomposed further.

There are two tools used for drawing the HIPO diagrams

- HIPO Worksheet ,GX20-1970
- HIPO template , GX20- 1971

The template contains the symbols for the HIPO diagrams. HIPO format consist of **Visual table of contents** shows the structure of the diagram and the relationships of the functions in a hierarchical manner. It also has a legend to show how symbols are to be used.

**Overview diagrams** describe the major functions and reference the major details diagrams needed to expand the functions adequately. They provide the following

- a) The input section that contains the data items used by the process steps.
- b) The output section that contains the data items created by the process steps.
- c) Process section that contains numbered steps that describe the functions to be performed. Arrows connect then to the output steps and input/output data items.
- d) The extended description refers to non-HIPO documentation and code

**Detail diagram** contains an extended description section that amplifies the process steps and references the code associated with each process steps.

## IPO-

The **input–process–output (IPO) model**, or **input-process-output** pattern, is a widely used approach in systems analysis and software engineering for describing the structure of an information processing program or other process. Many introductory programming and systems analysis texts introduce this as the most basic structure for describing a process.<sup>[1][2][3][4]</sup>

A computer program is useful for another sort of process using the input-process-output model receives inputs from a user or other source, does some computations on the inputs, and returns the results of the computations.<sup>[1]</sup> In essence the system separates itself from the environment, thus defining both inputs and outputs as one united mechanism.<sup>[5]</sup> The system would divide the work into three categories:

- A requirement from the environment (input)
- A computation based on the requirement (process)
- A provision for the environment (output)

In other words, such inputs may be materials, human resources, money or information, transformed into outputs, such as consumables, services, new information or money.

As a consequence, an Input-Process-Output system becomes very vulnerable to misinterpretation. This is because, theoretically, it contains all the data, in regards to the environment outside the system. Yet, in practice, the environment contains a significant variety of objects that a system is unable to comprehend, as it exists outside the system's control. As a result, it is very important to understand where

the boundary lies between the system and the environment, which is beyond the system's understanding. Various analysts often set their own boundaries, favoring their point of view, thus creating much confusion.

## **Assign1. Q5**

Several development activities are carried out during structured design. They are database design, implementation planning, system test preparation, system interface specification, and user documentation.

**1.Database design:** This activity deals with the design of the physical database. A key is to determine how the access paths are to be implemented. A physical path is derived from a logical path. It may be implemented by pointers, chains or other mechanisms.

**2.Program design:** In conjunction with database is a decision on the programming language to be used and the flowcharting, coding and debugging procedure prior to conversion. The operating system limits the programming languages that will run on the system. When the system design is done, the plans and test cases for implementation are required. So there must be detailed schedules for system testing and user training.

**3.System and Program test preparation:** Each aspect of the system has a separate test requirement. System testing is done after all programming and testing is completed. Acceptance testing is another testing that convinces the user that the candidate system will meet the stated requirements. It is conducted in the presence of users, audit representatives or the entire staff.

**4.System interface specification:** This phase specifies for the user the way in which information should enter and leave the system. The designer offers the user various options.

Before the system is ready for implementation, user documentation in the form of a user or operator's manual must be prepared. The manual provides instructions on how to access, update, or retrieve information, how to display or print output, in what format and so on.

**Personnel Allocation:** The structured approach is useful in planning process. A completed structure chart gives an idea of the work to be done. Programmers are

the assigned appropriately. Programmers are assigned subsystems that are strongly cohesive and loosely coupled. Once modules are allocated, roles are allocated within each team and the designer oversees all the work. Assigning modules are very important. Modules at the bottom are important because they represent the user interface. So a team with specialized skills should be assigned to such a module.