



InterviewBit

CICS Interview Questions



To view the live version of the page, [click here](#).

© Copyright by Interviewbit

Contents

CICS Interview Questions for Freshers

1. What is CICS?
2. What are the basic functions of CICS?
3. Explain TCT and RCT.
4. Difference between PCT (Program Control Table) and FCT (File Control Table)?
5. What are the main components of a CICS system?
6. What is the difference between a program and a transaction in CICS?
7. Describe CICS Task.
8. How are transactions processed in CICS?
9. What is the difference between a transient and a permanent dataset in CICS?
10. What is the purpose of the CICS Resource Definition Online (RDO), and how is it used?
11. How can we read a CSAM File in CICS?
12. What is "TransID"? Why is it used and where is it defined?
13. How Error Handling is done in CICS?
14. What is COMMAREA and explain its benefits.
15. What are the different types of DFHMDf present in the CICS Set?

CICS Interview Questions for Experienced

16. How does CICS differ from other transaction processing systems?
17. Explain some of the most commonly used CICS Commands.
18. What is the difference between a VSAM KSDS and a VSAM RRDS file? List in detail.

CICS Interview Questions for Experienced (.....Continued)

19. What is EIB? Explain in detail. Also, list the information this contains.
20. How do you ensure data integrity in a CICS program?
21. How Dynamic Memory Allocation is Handled in the CICS Program?
22. List out some points through which you can optimize the CICS performance.
23. What is the difference between a program-controlled task and a system-controlled task in CICS? Explain in Detail.
24. Can you implement a pseudo-conversational program in CICS? Also, Describe some practical examples of how the techniques are used in real-world scenarios.

Tips and Tricks

25. CICS Interview Preparation

Let's get Started

Introduction

CICS (Customer Information Control System) is an online transaction processing system that is widely used by many industries, including banking, insurance, and healthcare. As CICS is an important system, job seekers need to prepare well for interviews related to CICS.



CICS Interview Questions



This article covers a wide range of topics related to CICS, such as CICS commands, transactions, control blocks, and more including Multiple Choice Questions. We have also provided tips and tricks on how to approach CICS interviews and what the interviewer may be looking for in your responses.

So, if you want to excel in your CICS interview, dive into the following sections to get a better understanding of what to expect during your interview!

- [CICS Interview Questions for Freshers](#)
- [CICS Interview Questions for Experienced](#)
- [Tips and Tricks](#)
- [CICS MCQ Questions](#)

CICS Interview Questions for Freshers

1. What is CICS?

CICS full form is (Customer Information Control System). It is a software system that helps organizations manage their day-to-day business operations. Think of it as a central hub that connects various applications and allows them to interact with each other.

For example, if you are using an ATM to withdraw money, CICS would be the software that connects the ATM to your bank account and handles the transaction.

- It is a middleware software system developed by IBM that enables multiple applications to run concurrently and communicate with each other in real time.
- It is used in the mainframe system. CICS mainframe provides a runtime environment for executing business transactions, managing resources, and providing security, scalability, and reliability for the applications it supports.

2. What are the basic functions of CICS?

Some of the basic functions of CICS include the following:

- **Transaction management:** CICS provides facilities to initiate, monitor, and control the execution of transactions, which are the basic units of work in an OLTP system.
- **Task management:** CICS creates and manages tasks that execute programs, handle input and output, and communicate with other tasks.
- **Program management:** CICS provides facilities to load, link, and execute programs written in a variety of programming languages, including COBOL, PL/I, Assembler, and C.
- **Data management:** CICS provides access to various types of data sources, including databases, files, and message queues. CICS also provides facilities to manage data integrity and concurrency.
- **Communications management:** CICS provides facilities to manage communication between applications running in different CICS regions or different systems.
- **Security management:** CICS provides facilities to protect data and resources from unauthorized access and control access to CICS transactions and programs.
- **System management:** CICS provides facilities to manage the CICS system itself, including monitoring system performance, managing system resources, and maintaining system availability.

3. Explain TCT and RCT.

- **TCT (Terminal Control Table):** The TCT is a table that stores information about each terminal connected to the CICS system. This information includes the terminal device type, device address, and current status. The TCT is used by CICS to manage terminal activity, such as routing transactions to the correct terminal and managing screen displays. When a user initiates a transaction, CICS uses the TCT to determine which terminal the user is using and to route the transaction to that terminal. The TCT also helps to manage terminal resources and ensures that terminals are being used efficiently.
- **RCT (Resource Control Table):** The RCT is a table that stores information about various resources in the CICS system, such as files, programs, and transactions. This information includes the resource name, status, location, and other details. The RCT is used by CICS to manage resource activity, such as allocating and deallocating resources and coordinating resource sharing between multiple users. For example, if multiple users need to access the same file, the RCT will ensure that the file is locked and that users can access it without conflicts. The RCT also helps to manage system resources and ensures that they are being used efficiently.

The TCT and RCT work together to manage CICS resources and control terminals. **For example**, when a user initiates a transaction, CICS uses the TCT to determine which terminal the user is using and to route the transaction to that terminal. CICS then uses the RCT to allocate the necessary resources, such as programs and files, and to coordinate resource sharing between multiple users.

4. Difference between PCT (Program Control Table) and FCT (File Control Table)?

PCT (Program Control Table)	FCT (File Control Table)
It manages programs and transactions in a CICS system.	It manages files and datasets in a CICS system.
It contains information about each program or transaction, such as the program name, transaction ID, and load module name.	It contains information about each file or dataset, such as the file name, record format, and physical location.
It is used by CICS to load and execute programs and transactions and to manage program flow.	It is used by CICS to manage file access and processing, such as reading, writing, and updating files.
It allows CICS to dynamically load and execute programs, and to maintain program versions and libraries.	It allows CICS to manage file access modes, such as read-only, write-only, or update.
It can be updated dynamically by authorized users without requiring a system shutdown.	It requires a system shutdown to update the FCT.
The PCT is typically stored in the main storage (in memory) to improve program execution speed.	The FCT is typically stored in auxiliary storage (on disk) to reduce memory usage.

5. What are the main components of a CICS system?

A CICS system is composed of several key components that work together to process transactions and manage system resources. Some of the main components are:

- **CICS Region:** A CICS region is a logical grouping of CICS resources, including terminals, programs, and files. It is responsible for managing system resources and processing transactions.
- **Terminal Control Program (TCP):** The TCP manages communication between the CICS system and user terminals. It handles input and output operations, manages screen displays, and processes user requests.
- **Program Control Table (PCT):** The PCT is a table that contains information about programs and transactions in the CICS system. It is used to load and execute programs and to manage program flow.
- **File Control Table (FCT):** The FCT is a table that contains information about files and datasets in the CICS system. It is used to manage file access and processing, such as reading, writing, and updating files.
- **Resource Definition Online (RDO):** The RDO is a facility that is used to define and manage CICS resources, such as programs, files, and transactions. It allows authorized users to create, modify, and delete resources in the CICS system.
- **Data Communication System (DCS):** The DCS manages communication between the CICS system and other systems or applications. It provides communication protocols, message formatting, and network connectivity.
- **CICS Commands:** CICS provides a set of commands that are used to manage system resources and perform transactions. These commands are typically entered by users through terminal screens or batch jobs.

6. What is the difference between a program and a transaction in CICS?

- **Program:** A program in CICS is a set of instructions written in a high-level language, such as COBOL or PL/I. It is compiled, linked, and executed within the CICS environment. A program can perform a specific task, such as updating a file or generating a report.
- **Transaction:** A transaction in CICS is a logical unit of work that is processed by the system. It represents a user request, such as a query, an update, or a retrieval of information. A transaction is initiated by a user through a terminal or other input device, and it typically involves a sequence of program executions.

7. Describe CICS Task.

A CICS task is a logical unit of work initiated by a CICS transaction. It consists of a sequence of program calls and operations that are performed by CICS on behalf of the transaction. Each task has its environment and resources, including program work areas, communication buffers, and system control blocks.

Tasks can run in parallel with other tasks within a CICS region and can communicate with each other and exchange data using a variety of techniques. When the task is complete, CICS frees the resources allocated to it and returns control to the transaction.

8. How are transactions processed in CICS?

There are certain steps followed while the transaction is being processed. The steps are-

1. **The user initiates the transaction:** A user initiates a transaction by sending a request through a terminal or other input device. The user enters a transaction code, which is associated with a program or set of programs that perform the requested task.
2. **Transaction identification and routing:** The CICS system receives the transaction request and identifies the program or programs that need to be executed. The system also determines the resources, such as files and databases, that the transaction needs to access.
3. **Program execution:** The CICS system loads and executes the required programs to perform the transaction. The programs perform the requested task, such as updating a file or generating a report.
4. **Resource management:** During program execution, the transaction may need to access system resources, such as files or databases. The CICS system manages resource allocation and ensures that the transaction has the necessary permissions to access the resources.
5. **Data storage:** If the transaction updates data, the CICS system stores the data in temporary storage until the transaction is complete. If the transaction completes successfully, the updated data is written to permanent storage.
6. **Transaction completion:** Once the transaction is complete, the CICS system returns control to the user, and the user receives a response or output based on the requested task

9. What is the difference between a transient and a permanent dataset in CICS?

	Transient Dataset	Permanent Dataset
Purpose	Temporary storage for data processing	Permanent storage for data retention
Location	Stored in main storage	Stored in secondary storage (like - disk or tape)
Lifespan	Exists only for the duration of the transaction	Exists until explicitly deleted or the system is shut down
Size limitations	Limited by the amount of available main storage	Limited by the amount of available secondary storage
Access	Directly accessed by programs	Accessed through CICS file control
Recovery	Not recoverable in case of system failure	Recoverable through backup and restore procedures

Let's understand the use case scenario of both datasets.

- **Transient datasets** are typically used to hold intermediate results or temporary data that is not needed beyond the current job or program execution. Transient datasets are automatically deleted when the job or program completes or terminates. One common use case for transient datasets is batch processing.
 - **For example** - a batch job that processes a large amount of data might create temporary datasets to hold intermediate results, such as sorted or filtered data. These temporary datasets can then be passed as input to subsequent steps in the batch job.
- **Permanent datasets** can be accessed and updated by multiple jobs or programs and can be used for long-term storage of important data. A common use case for permanent datasets is in storing transactional data, such as customer orders or financial transactions.

In this scenario, the permanent dataset would be used to store the transactional data for future reference or analysis. Permanent datasets can also be used to store system configuration data, such as application settings or user preferences.

10. What is the purpose of the CICS Resource Definition Online (RDO), and how is it used?

It is a facility that is used to define and manage CICS system resources. The purpose of RDO is to provide a centralized and consistent way to define and manage system resources, such as files, programs, transactions, and terminals.

- **Resource definition:** RDO is used to define system resources and their attributes, such as record layouts for files or transaction codes for programs.
- **Resource maintenance:** RDO is used to maintain system resources, such as changing the attributes of a file or deleting a transaction code.
- **Security:** RDO is used to control access to system resources by defining security profiles for users and groups.
- **Backup and recovery:** RDO is used to manage the backup and recovery of system resources by defining backup and recovery procedures for each resource.
- **Performance monitoring:** RDO is used to monitor the performance of system resources by collecting and analyzing performance statistics.

RDO is used through a set of commands and panels that are accessed through the CICS system console or an interactive terminal session. The RDO commands are used to define, modify, and delete system resources, while the RDO panels provide a graphical interface for resource management and monitoring.

11. How can we read a CSAM File in CICS?

To read a VSAM file in a CICS program, we can use the following steps:

1. Define the file in the CICS program using a FILE control entry (FCT) in the program's PCT.
2. Open the file using the CICS OPEN command with the appropriate options. The options will depend on the access method required for the VSAM file (like - random, or sequential).
3. Read the data from the file using the CICS READ command with the appropriate options. Again, the options will depend on the access method and type of data to be read from the file.
4. After reading the required data, close the file using the CICS CLOSE command.

Consider the below command for the same:

```
EXEC CICS
  OPEN
  FILE(MYVSAMF)
  RIDFLD(MYKEY)
  TYPE(EXCI)
  END-EXEC
```

```
EXEC CICS
  READ
  FILE(MYVSAMF)
  INTO(MYDATA)
  LENGTH(MYLEN)
  END-EXEC
```

```
EXEC CICS
  CLOSE
  FILE(MYVSAMF)
  END-EXEC
```

12. What is “TransID”? Why is it used and where is it defined?

Transaction Identifier (TransID) is a short name or identifier that is used to identify a specific CICS transaction program. It is a unique identifier that is used to invoke a specific CICS transaction program and can be thought of as a program name or entry point.

A “TransID” is typically a 4-character alphanumeric string, and is defined in the CICS system definition (CSD) file. When a user initiates a transaction, they specify the Transid of the program they want to execute, and CICS invokes that program.

“TransIDs” is used extensively in CICS to initiate transactions and execute transaction programs. They are also used for program-to-program communication, where a program can pass control to another program using its TransID.

13. How Error Handling is done in CICS?

CICS provides various mechanisms for error handling to ensure that programs can handle any unexpected errors or exceptions that occur during runtime. Here are some ways in which error handling can be achieved in CICS:

- **Return codes:** CICS programs can return various return codes depending on the success or failure of an operation. These return codes can be checked and handled appropriately in the calling program to take corrective action.
- **Condition handling:** CICS programs can use the EXEC CICS HANDLE ABEND command to handle abnormal termination (ABEND) conditions that occur during runtime. This command allows the program to take corrective action when an unexpected exception occurs.
- **Recovery routines:** CICS programs can define recovery routines that can be executed when an error occurs. These recovery routines can be used to take corrective action and retry the operation or to log the error for analysis.
- **Exception handling:** CICS programs can use the **EXEC CICS HANDLE CONDITION** command to handle specific exceptions that may occur during runtime. This command allows the program to take corrective action when a specific exception occurs.
- **Logging and tracing:** CICS provides various logging and tracing facilities that can be used to track and diagnose errors that occur during runtime. These facilities can be used to identify the root cause of an error and take corrective action.

14. What is COMMAREA and explain its benefits.

COMMAREA stands for Communication Area, which is a shared data area that is used by CICS programs to exchange information with each other. The COMMAREA can be used to pass data between programs that are executed within the same CICS task or between different CICS tasks.

Using COMMAREA has several advantages:

- It helps in efficient and fast data exchange between programs without the overhead of writing data to a file or database.
- It has a standardized format for data exchange between programs that ensures consistent and reliable communication.
- It allows Dynamic data sharing without the need for pre-defined data structures or file layouts.
- It reduces I/O overhead which leads to improved performance of the application.
- It uses Transaction-based processing within a single atomic operation which leads to consistent and accurate data.

15. What are the different types of DFHMDf present in the CICS Set?

DFHMDf stands for CICS Map Definition Facility, which is used to define maps for CICS applications. There are different types of DFHMDf present in the CICS set. Each type serves a specific purpose and is used to define various aspects of CICS applications, such as maps, programs, files, and terminals. Below mentioned some of the DFHMDf sets:

- **BMS:** Basic Mapping Support. This is the standard format for defining CICS maps.
- **PCT:** Program Control Table. This is used to define the properties of the CICS program, such as its name and entry point.
- **FCT:** File Control Table. This is used to define the properties of CICS files, such as the file name, record format, and key length.
- **TCT:** Terminal Control Table. This is used to define the properties of CICS terminals, such as the terminal type, model, and keyboard layout.
- **XCTL:** Transient Program Control. This is used to transfer control from one CICS program to another.
- **MDT:** Map Destination Table. This is used to define the properties of the output destination for a CICS map.
- **MCT:** Map Conversion Table. This is used to define the properties of the data conversion process for a CICS map.

CICS Interview Questions for Experienced

16. How does CICS differ from other transaction processing systems?

Most transaction Processing System offers the same, But here are some ways in which CICS differs from other transaction processing systems:

- **Application flexibility:** It provides a high degree of application flexibility and has support for multiple programming languages, like - **COBOL, PL/I, C, C++,** and **Assembler**. This flexibility makes it easy for developers to write and maintain complex applications that require a variety of programming languages.
- **Scalability:** It is designed to handle large and complex transaction processing environments and can scale to meet the demands of large organizations. It can support thousands of transactions per second and can handle millions of users.
- **Reliability:** It is a highly reliable transaction processing system that offers features such as automatic restart, recovery, and backup. It can recover from system failures and errors quickly and efficiently.
- **Security:** It provides a wide range of security features, like - user authentication, authorization, and encryption. It can also protect against unauthorized access and attacks from external sources.
- **Integration:** It provides a wide range of integration capabilities, including support for various messaging systems, databases, and middleware technologies. This integration allows CICS to interact with other systems and applications seamlessly.

17. Explain some of the most commonly used CICS Commands.

- **EXEC CICS LINK** - This command is used to invoke a program from within another program in the same CICS region.
- **EXEC CICS XCTL** - This command is used to transfer control from one program to another within the same task.
- **EXEC CICS RETURN** - This command is used to return control from a program to the program that has been called.
- **EXEC CICS SEND** - This command is used to send data to another program or terminal.
- **EXEC CICS RECEIVE** - This command is used to receive data from another program or terminal.
- **EXEC CICS READ** - This command is used to read data from a file.
- **EXEC CICS WRITE** - This command is used to write data to a file.
- **EXEC CICS START** - This command is used to start a task that will run concurrently with the current task.
- **EXEC CICS SYNCPOINT** - This command is used to synchronize changes made to a database during a transaction.
- **EXEC CICS ENDMULTI** - This command is used to end a loop that was started with the EXEC CICS DO command.
- **EXEC CICS ASKTIME** - This command is used to obtain the current date and time from the CICS system clock.
- **EXEC CICS ASSIGN** - This command is used to assign a value to a field or variable.
- **EXEC CICS SET** - This command is used to set the value of a system control block or resource.
- **EXEC CICS INQUIRE** - This command is used to obtain information about a resource or system control block.
- **EXEC CICS GETMAIN** - This command is used to allocate main storage (i.e. memory) for use by a program.
- **EXEC CICS FREEMAIN** - This command is used to release the main storage that was allocated using the GETMAIN command.
- **EXEC CICS XCTLTRAN** - This command is used to transfer control from one task to another within the same CICS region.
- **EXEC CICS LINKPGM** - This command is used to invoke a program that resides in a different CICS region.
- **EXEC CICS RETURNTRANS** - This command is used to terminate a transaction and return control to the terminal or application that initiated it.
- **EXEC CICS SEND MAP** - This command is used to send a screen or map to a

18. What is the difference between a VSAM KSDS and a VSAM RRDS file? List in detail.



	VSAM KSDS	VSAM RRDS
Full form	Key Sequenced Data Set.	Relative Record Data Set.
Access Method	Keyed access method.	Relative byte address access method.
Record Retrieval	Records are retrieved based on the key value.	Records are retrieved based on the relative byte address.
Record Insertion	Records can be inserted in ascending or descending order based on the key value.	Records can be inserted at any available slot in the dataset.
Record Deletion	Records can be deleted using the key value.	Records can be deleted based on the relative byte address.
Record update	Records can be updated using the key value.	Records can be updated using the relative byte address.

19. What is EIB? Explain in detail. Also, list the information this contains.

EIB stands for Execution Interface Block, and it is a data structure used by CICS to communicate information between a CICS task and the CICS system. The EIB is available to all CICS programs and contains information about the current task, the transaction, and the environment in which the task is running.

CICS programs can access the EIB using the **EXEC CICS GET EIB** command to retrieve information about the current task and environment. The EIB can also be updated using the **EXEC CICS SET EIB** command to set values for certain EIB fields.

The EIB contains the following information:

- **Task information:** This includes the task number, the task type (batch or online), and the transaction code that initiated the task.
- **Security information:** This includes the user ID and the security level of the task.
- **Date and time information:** This includes the current date and time.
- **Terminal information:** This includes the terminal ID and the device type of the terminal.
- **Program information:** This includes the program name and the program type.
- **Language environment information:** This includes the language environment ID and the associated data.

20. How do you ensure data integrity in a CICS program?

Ensuring data integrity is an important aspect of any application, and CICS provides various mechanisms to maintain the integrity of data in a transactional environment. Here are some ways through which we can ensure data integrity in a CICS program:

- **Use CICS Syncpoints:** A Syncpoint is a checkpoint in the transaction where CICS ensures that all updates made to the resources are either committed or rolled back. By using Syncpoints, we can ensure that updates are committed only if all updates are successful, and rolled back if any update fails. This ensures that data is consistent and reliable.
- **Use CICS Restart:** CICS Restart is a facility that enables a program to recover from failures. By using CICS Restart, we can ensure that updates made to resources are not lost in the event of a failure, such as a system crash or network outage.
- **Use CICS journaling:** CICS journaling is a facility that records changes made to resources, such as files or databases. By using journaling, we can ensure that changes made to resources can be replayed in the event of a failure, thereby ensuring that data is consistent and reliable.
- **Use CICS program-controlled logging:** Program-controlled logging is a technique where the program explicitly logs changes made to resources. By using program-controlled logging, we can ensure that changes made to resources are logged in a controlled manner, thereby ensuring data integrity.
- **Use CICS Resource Control Table (RCT):** The RCT is a facility that enables you to define rules for accessing resources. By using the RCT, we can ensure that only authorized programs can access resources, thereby ensuring data integrity.

These are just a few examples of how we can ensure data integrity in a CICS program. By using a combination of these techniques, we can create a robust and reliable application that ensures data integrity.

21. How Dynamic Memory Allocation is Handled in the CICS Program?

Dynamic memory allocation in CICS programs is typically handled using the CICS dynamic storage allocation services. These services provide a way for programs to dynamically allocate and deallocate memory at runtime.

There are several CICS dynamic storage allocation services available, Some of them are -

- **GETMAIN:** This service is used to allocate a block of memory of a specified size. The allocated block can then be used by the program for temporary storage.
- **FREEMAIN:** This service is used to deallocate a block of memory that was previously allocated using GETMAIN. This releases the memory back into the system for reuse.
- **PUT** and **GET:** These services are used to allocate and deallocate storage for the duration of a task. PUT is used to allocate storage, while GET is used to deallocate storage.
- **EXEC CICS HANDLE CONDITION:** This service is used to handle storage violations that may occur during dynamic storage allocation. If a storage violation occurs, the program can use this service to handle the error and take appropriate action.

Programs can use these services to allocate memory for variables, tables, and other data structures needed at runtime. The allocated memory can be used for the duration of the program's execution and is automatically deallocated when the program terminates.

22. List out some points through which you can optimize the CICS performance.

- We can write efficient code using appropriate algorithms and data structures to optimize performance.
- We can use caching and buffering techniques to minimize the number of I/O operations.
- Excessive use of dynamic memory allocation can lead to performance problems. So we can allocate memory only when necessary and deallocate it as soon as it's no longer needed.
- We can use appropriate locking strategies and minimize the number of calls to the database.
- We can adjust system parameters to optimize performance. This includes tuning CICS transaction and region parameters.
- We can use CICS connection pooling to reduce the overhead of establishing a new connection for each transaction.
- We can use CICS thread pooling to improve throughput by reusing threads for subsequent requests.
- We can minimize the use of system resources, such as CPU and storage, to improve performance.
- We can use caching techniques to reduce database access and improve response time.
- We can use asynchronous calls instead of synchronous calls where possible to improve response time.
- We can optimize the network settings to improve performance.
- We can use CICS monitoring tools to identify bottlenecks and areas for improvement.

23. What is the difference between a program-controlled task and a system-controlled task in CICS? Explain in Detail.

Feature	Program-Controlled Task	System-Controlled Task
Who initiates task execution?	The application program initiates execution.	CICS initiates execution.
How is the task created?	Using the START command in the application program.	Using the CICS task control facilities, such as transaction initiation or event processing.
When is the task terminated?	When the program calls the EXEC CICS RETURN command.	When CICS determines that the task has been completed.
Who controls task execution?	The application program controls task execution.	CICS controls task execution.
Can it be dispatched by CICS?	No, the application program is responsible for dispatching the task.	Yes, CICS can dispatch the task.
Can it access system resources directly?	Yes, the application program can access system resources directly.	No, the system-controlled task must request access to system resources through CICS facilities.
Examples	Batch programs that run under CICS.	CICS transactions are initiated by users or by events.

24. Can you implement a pseudo-conversational program in CICS? Also, Describe some practical examples of how the techniques are used in real-world scenarios.

Here is the implementation of a pseudo-conversational program in CICS in the COBOL Programming language.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PSEUDOCONV.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT MYFILE ASSIGN TO SYSOUT.

DATA DIVISION.
FILE SECTION.
FD MYFILE.
01 MYREC PIC X(80).

WORKING-STORAGE SECTION.
01 WS-FIELD PIC X(80).
01 WS-FIRST-TIME VALUE 'Y'.
01 WS-CONV-IND PIC X(1) VALUE 'N'.

PROCEDURE DIVISION.
MAIN-PROCEDURE.
    PERFORM UNTIL WS-CONV-IND = 'X'
        IF WS-FIRST-TIME = 'Y'
            DISPLAY 'Enter input:'
            ACCEPT WS-FIELD
            MOVE 'N' TO WS-FIRST-TIME
        ELSE
            EXEC CICS SEND TEXT='Enter input:' RESPONSE=(WS-FIELD)
            EXEC CICS RECEIVE MAPSET('MYMAP') MAP('MYMAP') RESPONSE=(WS-FIELD)
        END-IF
        EXEC CICS WRITE FILE('MYFILE') FROM(MYREC)
        EXEC CICS SEND MAP('MYMAP') MAPSET('MYMAP')
        EVALUATE TRUE
            WHEN WS-FIELD = 'END' SET WS-CONV-IND TO 'X'
        END-EVALUATE
    END-PERFORM.
EXEC CICS RETURN.
```

In this example, the program uses a loop to repeatedly prompt the user for input until they enter **'END'**. The program starts by accepting input from the user and writing it into a file. If this is the first iteration of the loop, the program simply accepts input using the **'ACCEPT'** statement. However, if it is not the first iteration, the program uses **'EXEC CICS SEND'** and **'EXEC CICS RECEIVE'** commands to send and receive input using a CICS map.

The **'EVALUATE'** statement is used to check the user's input. If the user enters **'END'**, the program sets the **'WS-CONV-IND'** field to **'X'**, which causes the loop to exit.

Pseudo-conversational programming is used in real-world scenarios. Like -

- **Online Banking:** The user may need to perform multiple transactions like - balance inquiry, fund transfer, and bill payment. So, using pseudo-conversational programming, the system can ask the user to enter each transaction and then return to the main menu after the completion of the transaction. This approach makes sure that the user doesn't lose any entered data in case of any errors or the case when the system may fail.
- **Airline Reservation System:** Suppose the user needs to perform multiple tasks like - searching for flights, selecting a seat, booking a ticket, etc. So, pseudo-conversational programming helps here by asking the user for each task and then returning to the main menu after completing each of the tasks. This makes sure that the user doesn't need to start again if they face any errors in the process.

In both of the above scenarios, pseudo-conventional programming helps the user in a smooth experience when performing multiple tasks.

Conclusion

CICS is a powerful system used for processing transactions on mainframe computers. During a job interview, you might be asked questions about your experience with CICS, how it works, and how you develop and maintain applications using CICS. Some of the topics that could be discussed during the interview include the CICS commands, how to handle files, designing programs, and ways to improve performance.

To prepare for a CICS interview, it's important to review the concepts and best practices related to CICS development. It's also important to know the programming languages and tools used in the job you're applying for. By showing your expertise and ability to think critically about CICS development challenges, you can increase your chances of getting a job in this field.

Tips and Tricks

25. CICS Interview Preparation

Some tips and tricks to answer questions during your CICS Interview -

- **Know the basics of CICS:** Make sure you have a solid understanding of the fundamentals of CICS, including its architecture, programming languages, and tools used in CICS development. Familiarize yourself with the various CICS commands and their usage.
- **Be prepared to give specific examples from your experience:** Be ready to provide concrete examples of how you have used CICS in your experience. This could include specific programming tasks you have completed, challenges you faced and how you overcame them, and the impact of your work on the organization.
- **Practice problem-solving skills:** CICS development often involves problem-solving, so it's important to demonstrate your ability to approach and solve complex problems. Be ready to discuss your thought process and methodology when approaching a problem.
- **Demonstrate your communication skills:** Communication is critical in CICS development, so be prepared to discuss how you have communicated with team members, stakeholders, and customers. This could include discussing how you have documented your work, how you have provided updates on progress, and how you have collaborated with others on projects.
- **Be honest about your abilities and show a willingness to learn:** It's important, to be honest about your strengths and weaknesses in CICS development. Don't be afraid to admit when you don't know something, but be prepared to show a willingness to learn and grow in your role.

By following these tips, you can increase your chances of doing well in a CICS interview and landing a job in this field.

Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)