What is a transaction? What is a Task?
In CICS, you define a transaction, to run a program. Every transaction has a unique 4-character transaction-id. A task is a running instance of a transaction.

What are CICS Tables - PCT, PPT, FCT, RCT.
CICS maintains special tables to keep a track of different things.
PCT - List of transactions.
PPT - List of Programs.
FCT - List of DD-Names and physical datasets(files) used by program.
RCT - List of DB2 Tables used in a program.

What is BMS? What is Symbolic Map and Physical Map?
BMS stands for Basic Mapping Support. To create a new CICS GUI screen(map), the programmer must code BMS Macros - DFHMSD, DFHMDI and DFMDF.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
INQMAP  DFHMSD TYPE=&SYSPARM,LANG=COBOL,MODE=INOUT,TERM=ALL,           X
               CTRL=FREEKB,STORAGE=AUTO,TIOAPFX=YES
INQMAP  DFHMDI SIZE=(24,80),LINE=1,COLUMN=1
        DFHMDF POS=(5,1),LENGTH=07,ATTRB=(NORM,PROT),COLOR=BLUE,       X
               INITIAL='CUSTNO:'
CUSTNO  DFHMDF POS=(5,9),LENGTH=10,ATTRB=(NORM,UNPROT,IC),COLOR=GREEN, X
               INITIAL='__________'
```

When you assemble this BMS map, you get Symbolic Map and Physical Map.
The physical map is the actual load-module(executable) that is used by Cics to display the GUI screen.
The Symbolic map is a copy-book, that contains cobol variables corresponding to the fields on the map.

```
01 INQMAPI.
   03 FILLER                          PIC X(12).
   03 CUSTNOL                         PIC S9(4) COMP.
   03 CUSTNOF                         PIC X.
   03 FILLER REDEFINES CUSTNOF.
      05 CUSTNOA                         PIC X.
   03 CUSTNOI                         PIC X(6).
01 INQMAPO REDEFINES INQMAPI.
   ...
   ...
```

CUSTNOI is the name of the input-field. CUSTNOO is the name of the output field. CUSTNOL is the name of the length-field. CUSTNOA is the attribute-byte. CUSTNOF represents the Modified data tag(MDT). CUSTNOF contains X'0080' if it is modified, other-wise it contains Low-Values X'0000'. CUSTNOC represents the color.

When you execute EXEC CICS RECEIVE MAP('INQMAP') END-EXEC Command, the user-input is stored in cobol variable CUSTNOI. When you execute EXEC CICS SEND MAP('INQMAP') END-EXEC Command, the data from CUSTNOO field is displayed on the screen.

What is CTRL=FREEKB?
FREEKB option frees(resets) the Keyboard.

What CEDA, CEMT and CECI?
IBM ships some ready-made utility transactions like CEDA, CEMT, CECI and CESF, performing repetitive tasks such as maintaining CICS tables.

CEDA is a ready-made utility transaction by IBM, to DEFINE(Insert) new entries in CICS Tables. For example, to DEFINE a new TRANSACTION in PCT, you will type CEDA DEFINE TRANSACTION('INQ1'). To DEFINE a new PROGRAM in PPT, you will type CEDA DEFINE PROGRAM('PROG01').

CEMT is a ready-made utility transaction by IBM, to ENQUIRE(Search) existing entries in CICS Tables. For example, to ENQUIRE a transaction in PCT, you will type CEMT INQUIRE TRANSACTION('INQ1').

A second more important use of CEMT is to refresh the Load Pointer. Whenever you make changes(modifications) to cics program, you also need to refresh(reset) the pointer in the PCT Table, disconnect it from the old load, and point it to the new load. You type CEMT SET PROGRAM('PROG01') NEWCOPY.

CECI is the CICS Command Level Interpreter. Essentially, when you want to test some CICS Commands, outside of a Cobol host program(without writing a complete Cobol program), you can use the ready-made CECI transaction. For example, after designing a CICS Screen, when you want to test them, and see how they look, you type

```
CECI SEND MAP('INQMAP') MAPSET('INQMAP')
```

What are ERASE and ALARM?

You can code ERASE and ALARM options on the CICS SEND MAP command.

```
EXEC CICS
   SEND MAP('INQMAP')
        MAPSET('INQMAP')
        FROM(INQMAPO)
        ERASE
        ALARM
END-EXEC
```

ERASE option causes the contents of the screen to be erased before sending the map.
ALARM option causes an alarm-bell to sound, when the map is displayed.

What are MAPONLY and DATAONLY Options?

```
EXEC CICS
   SEND MAP('INQMAP')
        MAPSET('INQMAP')
        FROM(INQMAPO)
        ERASEAUP
        DATAONLY
END-EXEC
```

MAPONLY Option sends only the constant data from the physical-map.
DATAONLY Option sends only the data from the symbolic map(only data-entry fields are transmitted).

What is the difference between ERASE and ERASEAUP?

The ERASEAUP option only causes the Unprotected fields to be erased, while ERASE clears the complete screen. Most of the times, you MOVE LOW-VALUES TO INQMAPI, and then SEND MAP('INQMAP') with ERASEAUP and DATAONLY option. This clears data-entry fields on screen. No data is sent to the terminal(Remember CICS never transmits LOW-VALUES).

```
MOVE LOW-VALUES TO INQMAPI

EXEC CICS
   SEND MAP('INQMAP')
        MAPSET('INQMAP')
        FROM(INQMAPO)
        ERASEAUP
        DATAONLY
END-EXEC
```

```
                                CUSTOMER INQUIRY MAP

CUSTNO
 ________
 ________
 ________
 ________
 ________

 ENTER UPTO 5 CUSTOMER NUMBERS.
```

To create such an array of fields, you can specify the OCCURS option in the BMS Map.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
CUSTNO  DFHMDF POS=(5,9),LENGTH=10,ATTRB=(NORM,UNPROT,IC),COLOR=GREEN, X
               INITIAL='__________',OCCURS=05
```

How to create a repeat a group of fields in the Map as shown below?

```
                                CUSTOMER INQUIRY MAP

CUSTNO      FULL-NAME                 STREET        CITY        PINCODE
 ________   ________________________  ____________  ________    ______
 ________   ________________________  ____________  ________    ______
 ________   ________________________  ____________  ________    ______
 ________   ________________________  ____________  ________    ______
 ________   ________________________  ____________  ________    ______

 ENTER UPTO 5 CUSTOMER NUMBERS
```

You cannot use the OCCURS option to create an array of a set of fields. Instead first you must define all the fields individually, to produce the necessary Physical Map. You can then modify the resulting symbolic map, and replace individual field definitions by an array.

What is Attribute Byte?

On a CICS map, every field is preceded by an attribute byte. Take a look at the map below.

```
                          CUSTOMER INQUIRY MAP

CUSTNO:________

NAME  :________________

STREET:________________CITY:________PINCODE:______


PLEASE ENTER THE CUSTOMER NUMBER AND PRESS ENTER KEY
```

Each of the labels like CUSTOMER INQUIRY MAP, CUSTNO:, NAME :, STREET: etc. are preceded by an attribute byte. When you code LENGTH=07 in a BMS Map, the real length of the field = 07 Bytes + 1 Attribute byte = 8 Bytes. The attributes in an attribute byte determine the characteristic of the field. There are three standard attributes in this byte -
(Intensity,protection,shift). Intensity could be NORM,BRT or DRK. Protection could be UNPROT,PROT or ASKIP. Shift could be NUM. In addition to this you may code IC and FSET.

Unlike Labels, unprotected data-entry fields don't end until the next attribute byte is encountered. Data-entry fields don't end until there's another closing attribute byte. So, CUSTNO: data entry field needs an explicit closing attribute.

If a data-entry field like ones for Street and City are immediately followed by another field, the attribute byte for that field closes the entry field.

Examples of Attribute bytes
ATTRB=(NORM,PROT) – Normal, protected field. A Constant Label like CUSTNO:
ATTRB=(NORM,UNPROT) – Normal, unprotected field. An alpha-numeric Data-entry field.
ATTRB=ASKIP – Skipper field. Closing attribute byte for a data-entry field.
ATTRB=(NORM,NUM) – Numeric data-entry field.
ATTRB=(DRK,PROT,FSET) – A darkened Dummy field, which is always transmitted, to avoid mapfail condition.

What is the MDT Bit? What are FSET and FRSET options?
The last bit of an attribute byte is called the Modified Data Tag(MDT). The MDT bit for field is set automatically, when the user keys in data into that field. Whenever a field's MDT Bit is ON, it is transmitted to the program(stored in the symbolic map CUSTNOI Variable).

Sometimes, the user may not type/modify any field on the screen, and press ENTER AID Key. Since, no data is transmitted on RECEIVE MAP Command, the program abends with the CICS MAPFAIL Condition or ABM0 abend.

To avoid the MAPFAIL condition, you can create a 01-Byte dummy field on the map as follows.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
DUMMY   DFHMDF POS=(24,79),LENGTH=01,ATTRB=(DRK,PROT,FSET),INITIAL=' '
```

The FSET option always turns the MDT Bit ON, so this 1-byte dummy field, which is hidden and protected will always be transmitted, irrespective of whether user enters modifies or types on any of the fields on the screen or not.

The FRSET option turns the MDT Bit OFF.

What is the difference between conversational and pseudo-conversational programming?
Conversational programs wait and sit idle for user input.

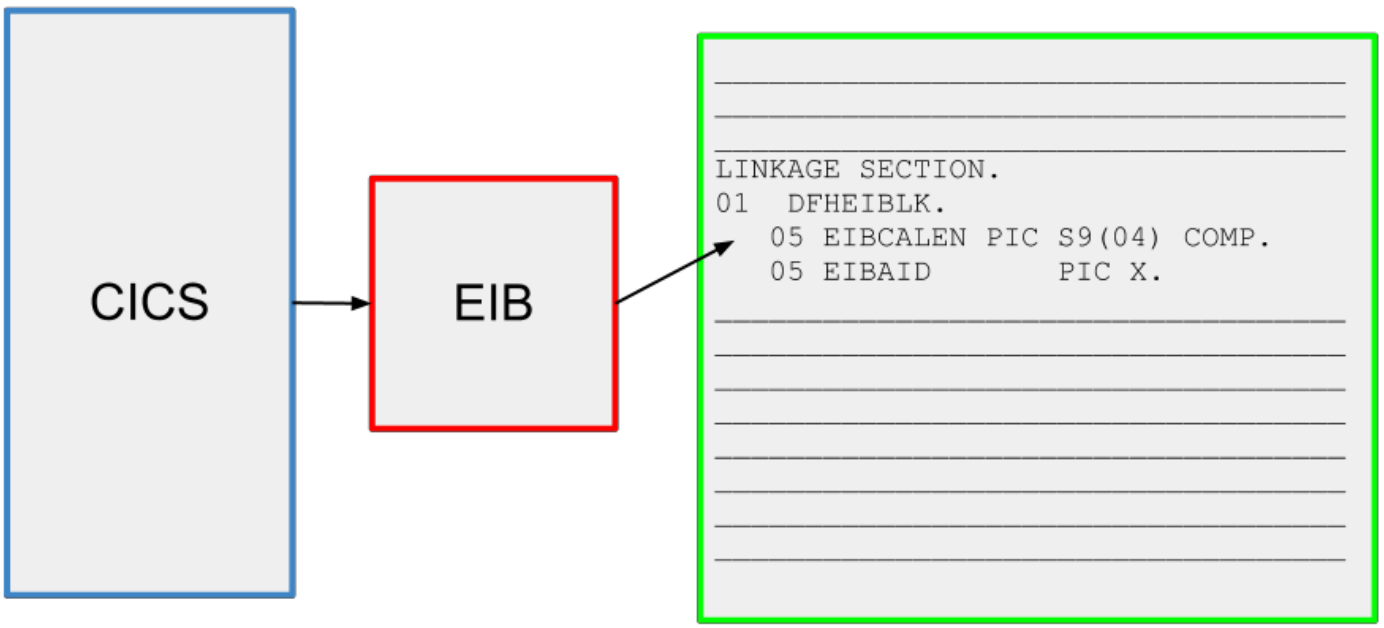
Pseudo-conversational programs end (they are killed) after the output is displayed on the screen. Once the user has keyed in all inputs, and presses a terminal attention-identifier key, only then the program wakes up. CICS restarts the program once again. This gives a false impression to the user that it is conversational, hence pseudo conversational.

What is Quasi-Re-entrancy?
A Quasi-reentrant program is the one, which does not modify itself, so that after an interruption by CICS, the control can re-enter and resume normal processing.

In a multi-threaded environment, several tasks may need to access the same program-code. A program which allows this activity harmlessly is called quasi-reentrant. To make a program quasi-reentrant, you must compile the Cobol program with RENT option.

What is EIB(Execute Interface Block)?
EIB(Execute Interface Block) is a shared area between CICS and your application program. CICS stores important control information such as the length of the comm- area in bytes, the AID Key the user pressed, the date and time the transaction started, the transaction-id, the response-code of the last cics-command in the Execute Interface Block.

How to access the data of EIB in the Cobol program?
CICS automatically inserts the code for the EIB(Execute Interface Block) in the LINKAGE SECTION of the Cobol Program. This establishes addressability between the cobol program and the EIB area in Memory. Using linkage fields EIBCALEN, EIBAID, EIBTIME, EIBTRNID, EIBRESP you can access useful information of the EIB area in your Cobol program



- What is DFHBMSCA? How to change an attribute to Dark dynamically?

What is the Communication area DFCOMMAREA? How do you store data in COMMAREA? How do you check the length in COMMAREA? What is it used for? What is the maximum size of the communication area?

CICS provides a special communication area called DFHCOMMAREA. The communication area is used to pass(communicate) data to the next program. To use the communication area you provide two definitions – a working storage definition WS-COMMAREA and a linkage section definition DFHCOMMAREA. When a cics program starts the data in communication area is available in DFHCOMMAREA, and can be used to determine the processing to be done. When the program completes, you do EXEC CICS RETURN TRANS(INQ1) COMMAREA(WS-COMMAREA) END-EXEC control back to CICS and specify the WS-COMMAREA, so that CICS takes WS- data and stores it in the communication area for the next program.

When a CICS program is invoked for the first time, no data is passed to it. Therefore, the communication area is empty. CICS always stores the length of the communication area in bytes in the EIB Field EIBCALEN. You can test IF EIBCALEN = 0, to check for first time execution.

What are SEND MAP and RECEIVE MAP Commands?

EXEC CICS SEND MAP END-EXEC Command is used to send map to the CICS Terminal. This displays the data in CUSTNOO, NAMEO, ADDRESSO, ...variables. The EXEC CICS RECEIVE MAP END-EXEC receives the user-input from the CICS Terminal and stores it in CUSTNOI, NAMEI, ADDRESSI, ... variables.

What is the design/skeleton of any basic CICS-Cobol application program?

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
       IDENTIFICATION DIVISION.
       PROGRAM-ID. PROG01.

       ENVIRONMENT DIVISION.

       DATA DIVISION.
       WORKING-STORAGE SECTION.

      ************************SYMBOLIC MAP COPYBOOK*********************
       COPY INQMAP.
      ******************************************************************

      *******************************AID KEYS***************************
       COPY DFHAID.
      ******************************************************************

       01  WS-COMMAREA.
           05 WS-CONTEXT-AREA                PIC X.

       LINKAGE SECTION.
       01  DFHCOMMAREA                       PIC X.

       PROCEDURE DIVISION.
       0000-MAIN.
           IF EIBCALEN = 0
              PERFORM 1000-SEND-MAIN-MENU-MAP
              THRU    1000-EXIT
           ELSE
              EVALUATE TRUE
                 WHEN EIBAID = DFHPF3
                      ...
                 WHEN EIBAID = DFHPFENTER
                    PERFORM 2000-PROCESS-MAIN-MENU-MAP
              END-EVALUATE
           END-IF

           EXEC CICS
              RETURN
           END-EXEC
```

What is CICS translator used for?

When you code a cobol-cics program, the program contains EXEC CICS cics-command END-EXEC commands embedded inside the Cobol program. The Cobol compiler only translates Cobol statements to machine code, it does not understand CICS commands.

A CICS Translator software is needed. The CICS Translator software translates any EXEC CICS commands, into Cobol MOVE and CALL statements, which can be compiled by the Cobol Compiler. After you translate your program, you will compile and link-

edit it just as any other normal cobol program, to produce a load-module(executable).

How will you solve APCT cics abend?
APCT Abend - Make an entry in the CICS PCT Table for this transaction.

What is AEY9 abend?
AEY9 abend happens, when CICS is unable to establish a link with DB2, and happens especially when DB2 is down.

What is ASRA abend?
ASRA is similar to S0C4 or S0C7 abends in batch. It is basically a protection exception, and we need to find the offending instruction that causes the ASRA abend. You use debugging tools like xpeditor, CEDF.

What are CICS LINK, CICS XCTL commands and Cobol CALL statement used for? How do they differ?
When a program is invoked by entering a Transaction-id, it executes at one logical-level below CICS. Thus when a program ends, by issuing RETURN command, the control returns back to CICS.

When a Program A LINKS to another Program B, the program B executes at one logical-level below the calling program A. The calling program A stays in main-memory all the while. When the program B completes and issues a RETURN Command without any options, the control returns back to the calling program A. Each time a LINK Command is used to invoke a program, a fresh copy of the working storage is obtained.

LINK is generally useful for sub-programs.

When a Program B XCTLs to another Program C, the control is transferred permanently to program C, the program B is moved out from main storage, and program C takes its place at the same logical level. Now, if the program C issues a RETURN Command without any options, the control returns back to the next higher level program A.

XCTL Command is very useful in complex Menu Programs.

The cobol CALL statement works just like the LINK Statement. It is used for sub-programs. There is a subtle difference between CICS LINK Command and Cobol CALL statement.

LINK - Each time a fresh copy of WS-Variables is obtained.
CALL - Memory is allocated and WS-Variables are initialized only once for the first time.

When a sub-program is repetitively executed for a certain task, CALL is more efficient over LINK.

What are the CICS File I/O commands?
In Cics, you can process VSAM KSDS, ESDS and RRDS Files.

Random Processing : READ, WRITE, REWRITE, DELETE
The key field is specified using RIDFLD Option. If you want to update a record, you must first READ with UPDATE option and then REWRITE.

What are different CICS Exception conditions?
DUPREC - Duplicate record found.
INVREQ - Invalid request(for example, you are trying to REWRITE, before READ).
NOTFND - Record not found.
MAPFAIL - Happens during RECEIVE MAP Command, when no data was entered by the user.
PGMIDERR - Happens during XCTL/LINK, when the program-id is not found in PPT.

How to check response code of any CICS command?

```
IF EIBRESP = DFHRESP(NORMAL)
   ...
ELSE
IF EIBRESP = DFHRESP(NOTFND)
   ...
END-IF
```

What is a TSQ? How to create, read, write and delete a TSQ?

The communication area is one way to pass Data between executions of a program. The communication area can accommodate upto 32,763 bytes of data(Max = 32 KB). The other way to save data between program executions is TSQ(Temporary Storage Queue).

A TS Queue has a name 1-8 characters. As a good practise, to ensure that TS Queue names are unique, you code the terminal-id as a part of the TSQ. Each TS Queue contains one or more records. Each record or item is assigned an item-number. The first item has item number 1,the second has item number 2 and so on.

You can retrieve records from a TS Queue either randomly or sequentially.

```
01 QUASAR CHUNAWALA
02 CHINTAN BHAVSAR
03 NEHA RAWAT
04 NAFISA CHUNAWALA
05 SHABBIR CHUNAWALA
```

To read the next record from the TSQ,

```
EXEC CICS
   READQ TS QUEUE(Q001)
            INTO(WS-EMPLOYEE-RECORD)
END-EXEC
```

To retrieve a record randomly from the TSQ,

```
MOVE 03 TO WS-ITEM-NUMBER

EXEC CICS
   READQ TS QUEUE(Q001)
            INTO(WS-EMPLOYEE-RECORD)
            ITEM(WS-ITEM-NUMBER)
END-EXEC
```

To write a new record to the TSQ,

```
MOVE 06 TO WS-ITEM-NUMBER
MOVE 'SANKET PATIL' TO WS-EMPLOYEE-RECORD

EXEC CICS
   WRITEQ TS QUEUE(Q001)
             FROM(WS-EMPLOYEE-RECORD)
             ITEM(WS-ITEM-NUMBER)
```

To update an existing record in TSQ, use the REWRITE Option of the WRITEQ TS QUEUE Command.

What are Transient Data Queues(TDQs)?
CICS provides a scratchpad facility, that may be used as an alternative to the TEMPORARY STORAGE. The main differences are :

- Transient Data Queues must be pre-defined to CICS by the Systems Programmer in the special CICS Table DCT.
- These queues are strictly sequential, no random access is available. Their main use is exchange of data between CICS applications, or CICS region and batch region. TDQs are often used for printing.
- TD Queues are read-destructive. You cannot therefore re-read an item.

TD Queues can be defined to be CICS-only, or they may be defined to be written in CICS and read by other regions e.g. batch.

- Those within CICS are called Intra-partition TDQs. They are disk-only.
- Those outside CICS are called Extra-partition TDQs.

Intra-partition TD Queues may be used to initiate asynchronous tasks. A TDQ may be defined with a trigger level : when the number of items on the queue reaches a specified trigger, a separate task is initiated.

What is Automatic Task Initiation? How do you start a new background task in CICS?
CICS allows multi-tasking. Multiple tasks can run under the same CICS address space. CICS allows you to start a new task(transaction), from your current program using the START command.

```
EXEC CICS
   START TRANSID(PCTS)
END-EXEC.
```

CANCEL is used to cancel a task.

How to dynamically allocate memory in CICS? What are GETMAIN and FREEMAIN commands?
The function of GETMAIN command is to obtain a certain amount of storage, on-the-fly. Once storage is allocated, you can set a pointer to it. This pointer is generally a Linkage section variable.

```
WORKING-STORAGE SECTION.
01  INITIAL-VALUE PIC X LOW-VALUE.

LINKAGE SECTION.

01  LS-PARM-AREA.
    05 MEMORY-POINTER PIC S9(08) COMP.

01  LS-MEMORY.
    05 FIELD-1 PIC X(100).
    05 FIELD-2 PIC X(200).

PROCEDURE DIVISION.
    EXEC CICS
       GETMAIN SET(MEMORY-POINTER)
               LENGTH(300)
               INITIMG(INITIAL-VALUE)
    END-EXEC.
```

At the completion of the above GETMAIN command, the storage of 300 Bytes will be acquired, and initialised to LOW-VALUES.