|  |  |  |
| --- | --- | --- |
| **Control Flow Constructs** | **Description** | **Freq.** |
| **PROCEDURE DIVISION/USING** | A “Run Unit” – which may or may not have an **ENTRY** statement | **3** |
| * [**ENTRY**](https://www.ibm.com/docs/fi/cobol-aix/5.1?topic=statements-entry-statement) **XYZ** **USING**. Can call to program and to a paragraph using ENTRY statements.
 | Program-level Label – callable by the z/OS or any linked executable. You can code ENTRY anywhere in the PROCEDURE DIVISION. Meaning that you could turn a paragraph into a Linked Module with an **ENTRY PARA-NAME** statement. This is rare. | **2** |
| **Program end or exit:** | Termination of Run Unit | **3** |
| * **CALL**
 | Transfer of control to an external executable – note that there is a z/OS CALL and an EXEC CICS CALL – both of which transfer control | **3** |
| * [**EXEC CICS LINK**](https://www.ibm.com/docs/en/txseries/9.1?topic=SSAL2T_9.1.0/com.ibm.cics.tx.doc/concepts/c_exec_cics_link_exec_cics_xctl_cmnds.htm)
 | CICS control transfer out of the program – return is expected | **2** |
| * [**EXEC CICS XCTL**](https://www.tutorialspoint.com/cics/cics_control_operations.htm)
 | CICS control transfer out of the program – without having to return | **2** |
| * **GOBACK**
 | Ends the current Run Unit and returns control to z/OS – or to a calling module | **3** |
| * [**EXIT**](https://www.ibm.com/docs/en/cobol-zos/6.1?topic=statements-exit-statement)
 | Ends a subroutine and returns control to the calling program | **1** |
| * **EXEC CICS RETURN**
 | Returns control to the module that is LINKED or CALLED | **2** |
| * **STOP RUN**
 | [Ends the current Run Unit and returns control to z/OS](http://www.differencebetween.net/technology/software-technology/difference-between-goback-and-stop-run-in-cobol/#:~:text=STOP%20RUN%20and%20GOBACK%20are,called%20by%20a%20COBOL%20program.) | **2** |
| * **ABEND**
 | Terminates the Run Unit throwing a z/OS error | **1** |
| **Internal program procedural flow** – *It would be useful for the diagram to notate conditional vs. unconditional flow* | PERFORM statements create a “perform chain” – machine code that returns to the NSI (Next Sequential Instruction after the PERFORM) at the logical end of the chain |  |
| **Potential Fall Thru situations** |
| * + Paragraphs coded within COBOL SECTIONS
 | All executable statements within Paragraphs are executed sequentially/physically downward thru the source until the next SECTION Label is hit – ***this includes Paragraph and all code within paragraphs*** | **1** |
| * + **GO TO** statements that transfer control outside of the “Perform Chain”
 | Paragraph labels are ignored and all statements are executed sequentially downward until the physical end of program code is hit, or one of the above Program Exit statements occur. Note that GO TOs are operational | **1** |
| * **PERFORM** <label>
 | There are both **inline** and **external** PERFORM statements. External PERFORM statements reference a paragraph(label). Note that inline PERFORM statements almost always execute non-control flow statements | **3** |
| * + PERFORM n TIMES
		- ***DO WHILE***
 | Branch to a paragraph or SECTION n number of times – when n is reached execute the NSI after the PERFORM statement | **2** |
| * + PERFORM UNTIL
		- ***DO UNTIL***
 | Branch to a paragraph or SECTION until a given condition (IF logic) is met. When the logical condition tests true execute the NSI after the PERFORM statement | **3** |
| * + [PERFORM <paragraph> THRU <exit\_paragraph>](https://www.ibmmainframer.com/cobol-tutorial/cobol-exit-statement/)
 | Branch to paragraph and execute all logic between paragraph and exit\_paragraph. The compiler sets up a “Perform Chain” with Branch instructions | **2** |
| * **GO TO** - Any label anywhere within the PROCEDURE DIVISION
 | Transfer control to any label in the Run Unit – including back to the current paragraph in a procedural DO loop. | **3** |
| * [ALTER](https://www.ibm.com/docs/en/cobol-zos/6.3?topic=statements-alter-statement)
 | Dynamically modify the destination label of a GO TO statement at run time. Almost never seen in a current production stack | **< 1** |
| * **EXIT**
 | Return to the NSI (Next Sequential Instruction) after the PERFORM statement |  |
| * + PARAGRAPH/SECTION
 | Branch back to the NSI at the logical end of the PERFORM chain | **1** |
| * + PERFORM
 | End an inline PERFORM Loop | **1** |
| * + PROGRAM
 | See Program EXIT above | **2** |
| * **SORT/MERGE**
 | Used to collate/merge external file records based on key field(s) |  |
| * + INPUT PROCEDURE
 | Branches to a COBOL SECTION to read & process the input file | **1** |
| * + OUTPUT PROCEDURE
 | Branches to a COBOL SECTION to sort & process the output file | **1** |
| Embedded (**EXEC SQL WHENEVER**) Transfers – using GO TO a given label* SQLERROR
* SQLWARN
* NOT FOUND
 | **EXEC SQL WHENEVER** statements setup a condition (insert machine code) at the end of every EXEC SQL operational statement in a Run Unit – to branch to a label when an error (SQLCODE < 0) or warning (SQLCODE > 0) or NOT FOUND/SELECT (SQLCODE = 100) occurs | **1** |
| [**EXEC CICS – HANDLE CONDITION**](https://www.ibm.com/docs/el/cics-ts/5.1?topic=commands-handle-condition) | “Handle Condition” == “WHENEVER” - Same behavior as the embedded SQL – except found in an EXEC CICS statement. | **2** |
|  |

|  |  |  |
| --- | --- | --- |
| **Use Case programs** | **Description** | **Results** |
| USECASE1 | Sequential Fall Thru Logic – no GOBACK or STOP RUN |  |
| USECASE2 | Sequential Fall Thru Logic – STOP RUN |  |
| USECASE3 | Sequential Fall Thru Logic – GOBACK |  |
| USECASE4 | Call to SUBROUT1 – STOP RUN |  |
| USECASE5 | Call to SUBROUT1 – GOBACK |  |
| USECASE6 | Call to SUBROUT1/PARA3 - ENTRY statement in PARA3 |  |
| USECASE7 | Sequential Logic – PERFORM Statements |  |
| USECASE8 | Sequential Logic – PERFORM chain |  |
| USECASE9 | Sequential Logic – GO TO Statements |  |
| USECASEA | Mix of COBOL SECTIONs and Paragraphs |  |
| USECASEB | Mix of COBOL SECTIONs and Paragraphs |  |
| USECASEC | GO TOs with a **conditional** GOBACK in PARA7 |  |
| USECASED | GO TOs with an **unconditional** GOBACK in PARA7 |  |
| USECASEE | Random PERFORMs |  |
| USECASEF | No STOP RUN or GOBACK (Falls thru) |  |
| USECASEG | Backwards Go Tos |  |
| CNTRLBRK | Sort Statement with INPUT & OUTPUT Sections |  |