



JCL

Chapter a1 Introduction to JCL

Job Control Language

Chapter a1. Introduction to JCL

Chapter a2. Coding JOB statements

Chapter a3. Coding EXEC statements

Chapter a4. Coding DD statements

Chapter a5. Analyzing job output

Chapter a6. Conditional processing

Job Control Language

Chapter b1. Using special DD statements

Chapter b2. Introducing procedures

Chapter b3. Modifying EXEC parameters

Chapter b4. Modifying DD parameters

Chapter b5. Determining the effective JCL

Chapter b6. Symbolic parameters

Job Control Language

Chapter c1. Nested procedures

Chapter c2. Cataloging procedures

Chapter c3. Using utility programs

Chapter c4. Sample utility application

Introduction to JCL

Chapter a1

Introduction to JCL

Introduction to JCL

Course objectives.

Be able to:

- **Identify JCL statements and their purpose.**
- **Code a JOB statement.**
- **Code an EXEC statement to invoke a program.**
- **Code a DD statement to identify a data set.**
- **Code the library DD statements JOBLIB and STEPLIB.**
- **Identify the JCL statement errors in an output listing and code appropriate corrections.**
- **Code the IF/THEN/ELSE/ENDIF JCL statement construct.**

Introduction to JCL – Introduction.

Job Control Language.

What is JCL?

Job Control Language (JCL) is a control language used to identify a job to an operating system and to describe the job's requirements.

JCL is used to describe the work you want a system using multiple virtual storage (MVS) to perform.

The three main types of JCL statements are:

- **JOB - It marks the beginning of a job and identifies the job name.**
- **EXEC - It marks the beginning of a job step and specifies the name of the program (or procedure) to be executed.**
- **DD - It describes data sets to be used within individual steps.**

Job Control Language.

What is a job?

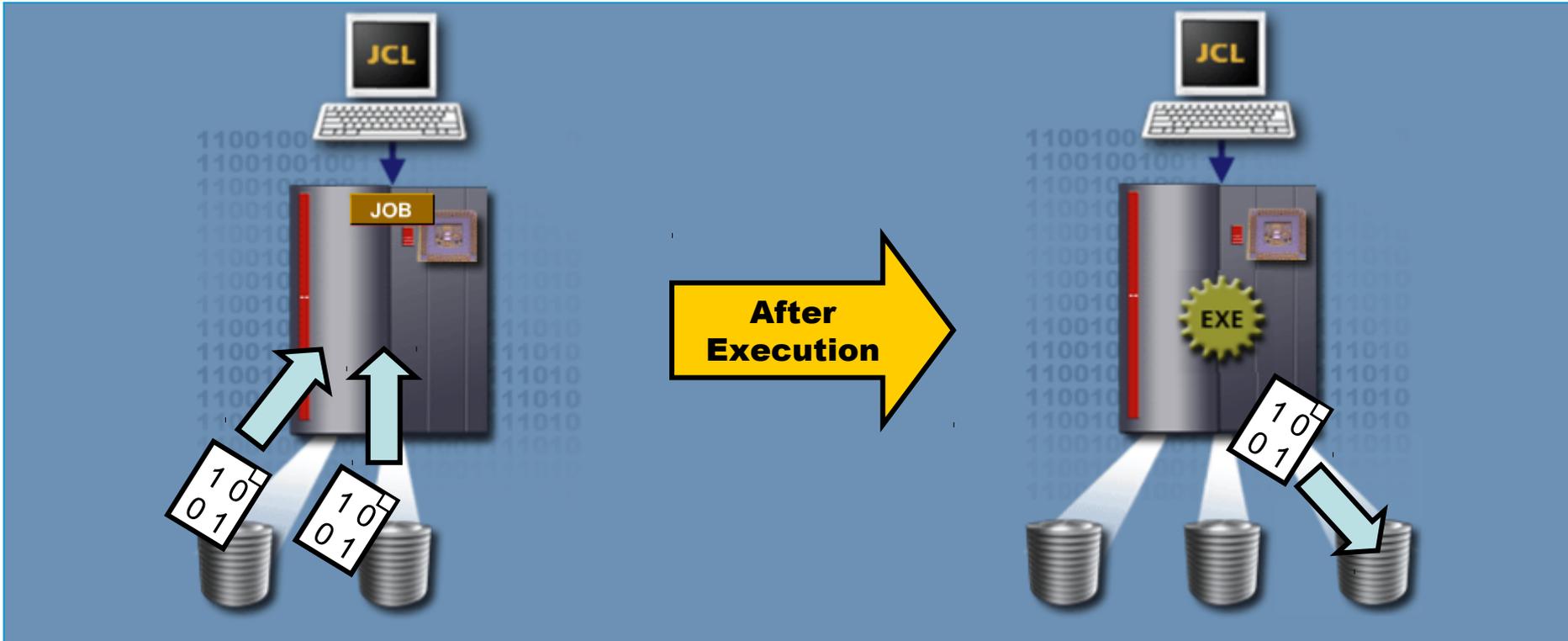
A job is a unit of work defined by a user to be accomplished by a computer, including computer programs, files or control statements.

For example: The request to perform a task of updating one data set with information from another data set is called a job.

JCL provides the operating system with information about the resources, programs, and data necessary to complete the job.

Introduction to JCL – Introduction.

Jobs – an example.



Example: A job may direct the computer to run a program that combines two data sets into one new data set.

Introduction to JCL – Introduction.

JCL job information.

JCL provides information to the system about job details including:

- The programs to execute.
- The location of the required data sets.
- The department to be billed for CPU processing time.
- The job class.



Introduction to JCL – Introduction.

Job stream.

One or more jobs placed in a series and ready to be submitted to the system are referred to as a job stream.

This series of jobs is entered through one input device. You can use JCL to structure a job stream.

The JCL for each job begins with a JOB statement.



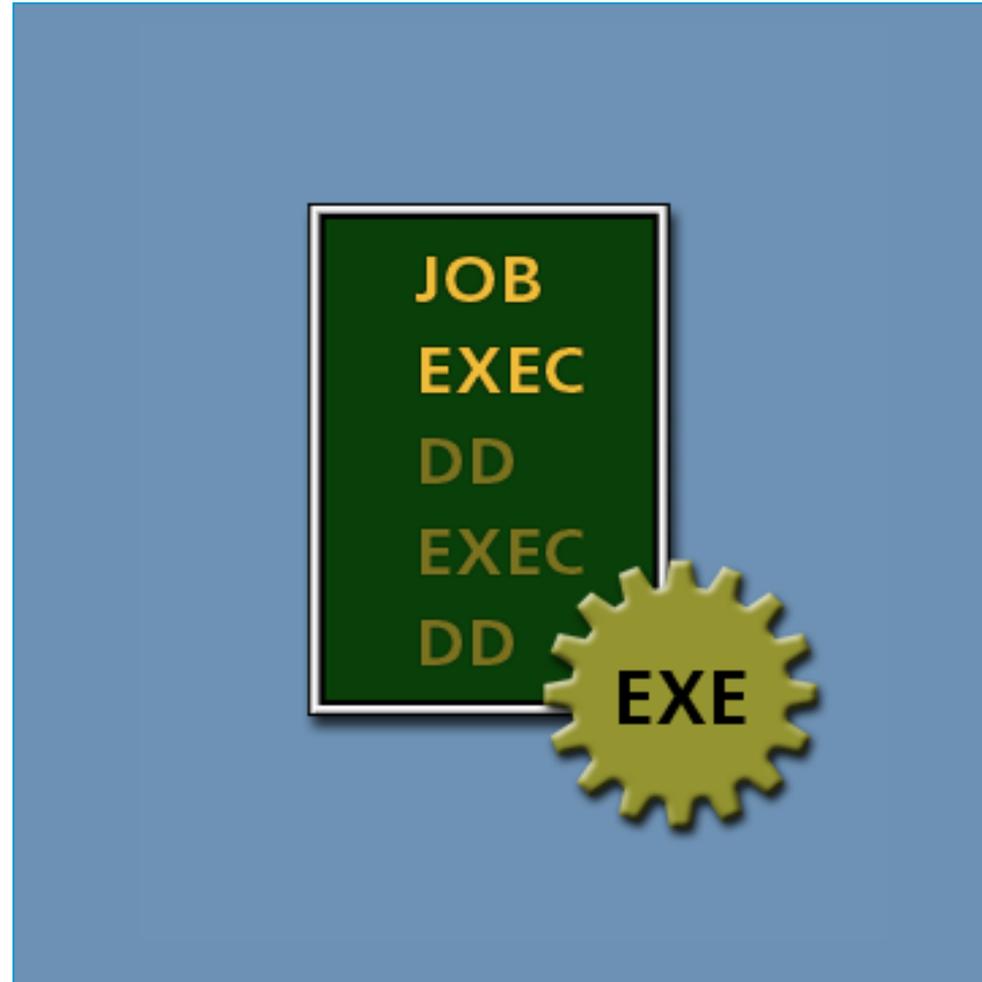
Introduction to JCL – Introduction.

Job steps.

A job consists of one or more individual job steps.

Each job step starts with an EXEC statement and is associated with the execution of one program (or procedure).

The data sets required for each job step can be defined using a DD statement.



Introduction to JCL – Introduction.

Are we on track?

The JCL specifying a collection of jobs to be run is called a(n)

Glossary.

Batch

A group of records or data processing jobs brought together for processing or transmission.

Central Processing Unit (CPU)

The CPU controls the operation of a computer. Units within the CPU perform arithmetic and logical operations and decode and execute instructions.

Data set

Synonym for file. A unit of information that can be stored and retrieved.

DD statement

A JCL statement describing a data set associated with a specific job.

Introduction to JCL – Introduction.

Glossary.

EXEC statement

Marks the beginning of a job step; assigns a name to the step; identifies the program or procedure to be executed in this step.

Job

A unit of work defined by a user to be accomplished by a computer, including computer programs, files or control statements.

Job step

The JCL statements that request and control the execution of a program and request the resources needed to run the program. A job step is identified by an EXEC statement.

Job stream

Jobs placed in a series and entered through one input device.

Multiple Virtual Storage (MVS)

The standard operating system for older IBM mainframes.



Introduction to JCL – Job control statements.

Defining Job control statements.

Job control statements define jobs and job steps.

JCL statements provide the following information:

- **Programs to be executed.**
- **Sequence of program execution.**
- **Data sets required by the programs.**

Introduction to JCL – Job control statements.

Defining Job control statements.

Each job requires you to specify at least the following:

- Job to begin (using a JOB statement).
- Programs or procedures to be executed (using EXEC statements).
- Data sets to be used (using DD statements).



Introduction to JCL – Job control statements.

Are we on track?

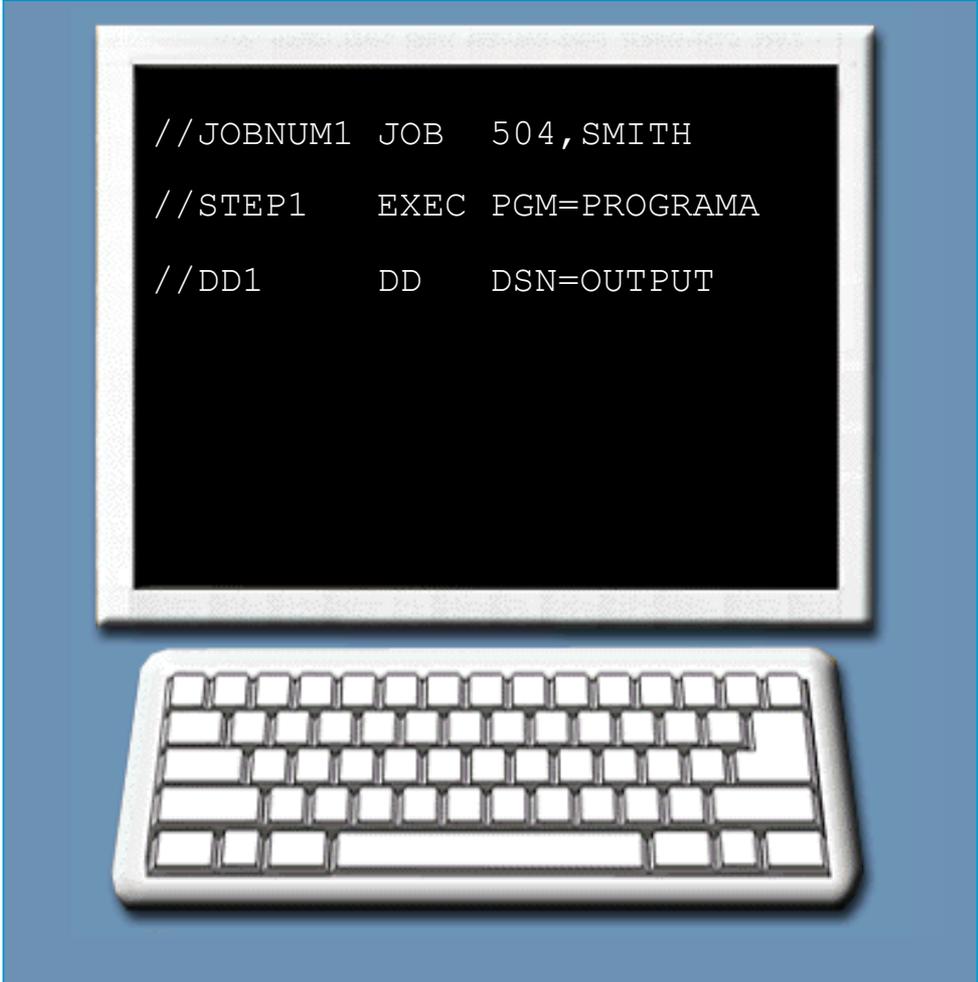
You would use the _____ statement to run a program as part of a job.

Introduction to JCL – Job control statements.

The JOB statement.

The JOB statement is the first control statement in a job. It marks the beginning of a job and specifies the job name.

The JOB statement may also provide details and parameters that relate to the overall job, such as accounting information and conditions for job termination.

A computer monitor with a white frame is centered in the image. The screen is black and displays three lines of white text representing JCL statements. Below the monitor is a white keyboard with a black grid pattern on the keys. The entire scene is set against a light blue background.

```
//JOBNUM1 JOB 504,SMITH
//STEP1 EXEC PGM=PROGRAMA
//DD1 DD DSN=OUTPUT
```

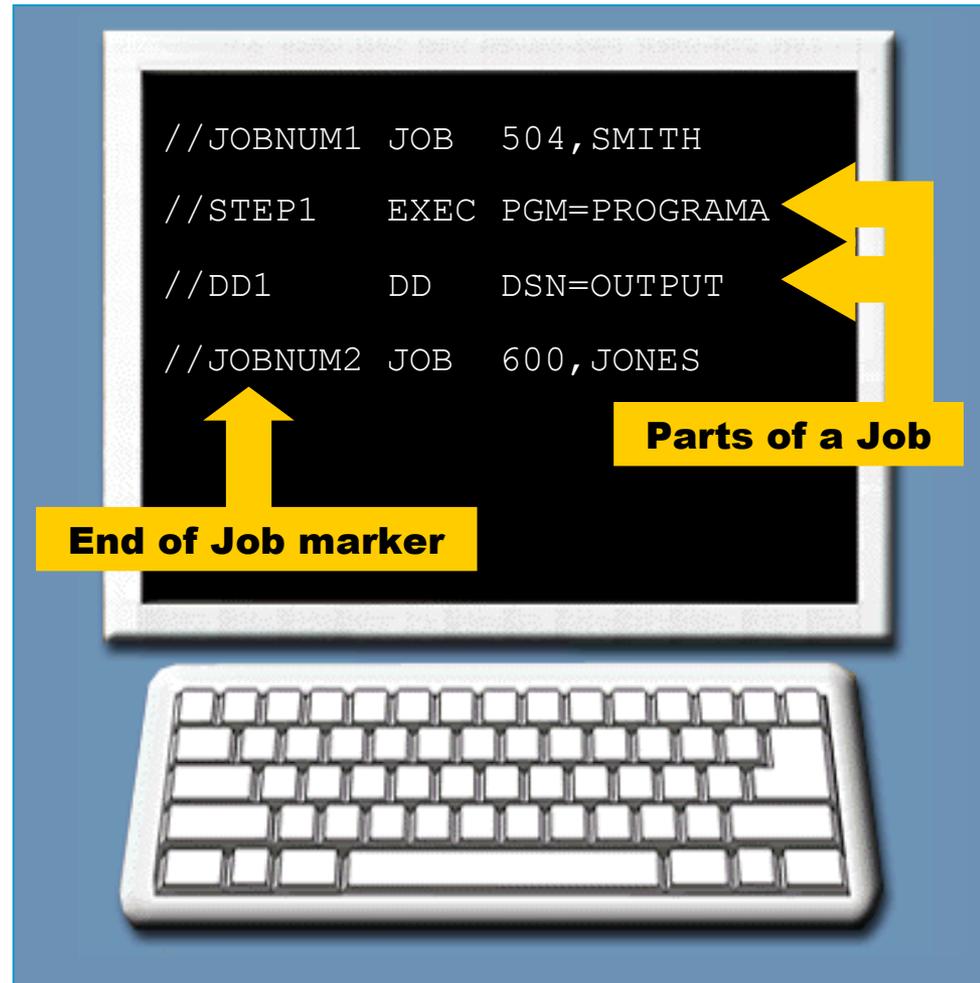
Continued... 

Introduction to JCL – Job control statements.

The JOB statement.

Each job must begin with a JOB statement. All statements that follow up to the next JOB statement, are parts of one job.

The end of a job is marked by either another JOB statement or by a null statement.



Continued... 

Introduction to JCL – Job control statements.

Are we on track?

Which of the following is a function of the JOB statement?

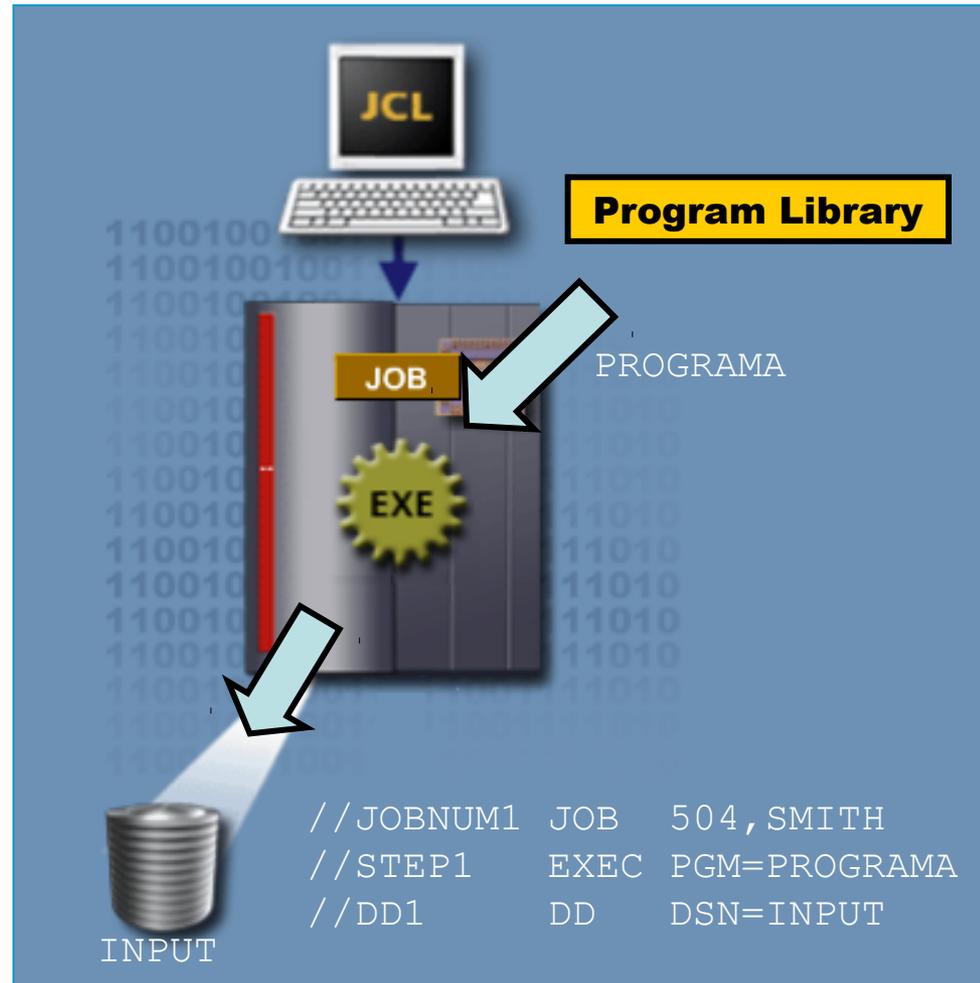
- A. Contains documentation information only.**
- B. Identifies the program to be run.**
- C. Identifies the data sets to be used during the job.**
- D. Identifies the job name and specifies accounting information.**

Introduction to JCL – Job control statements.

The EXEC statement.

An EXEC statement often follows the JOB statement and identifies a program or procedure to be executed.

A job contains one or more EXEC statements, each followed by statements that define one or more data sets needed for that job step.



Continued... 

Introduction to JCL – Job control statements.

Are we on track?

Where are programs referenced by an EXEC statement stored?

A. Sequential data sets.

B. Program libraries.

C. Tape devices.

D. Procedure libraries.

Introduction to JCL – Job control statements.

The DD statement.

DD statements define the data sets the program uses during execution. Each data set used in the job requires at least one DD statement.

The number of DD statements following an EXEC statement depends on the number of data sets required by a program.

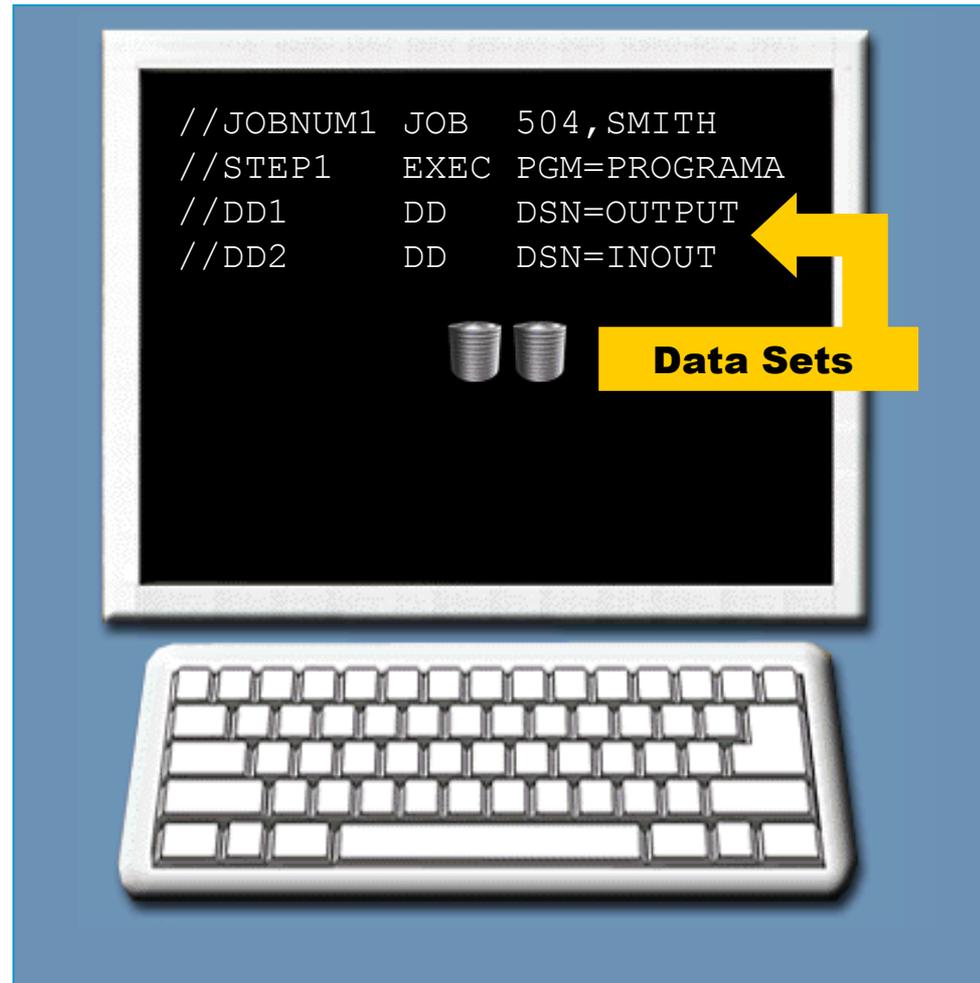
The order of DD statements within a job step is not usually significant.

Introduction to JCL – Job control statements.

The DD statement.

The DD statement describes the data set extensively and can include the following information:

- Give the data set name.
- Specify the initial status and final disposition of the data set.
- Specify volumes.
- Specify record length, format, and blocking.
- Request I/O devices.
- Specify storage allocation for new data sets.



Continued... 

Introduction to JCL – Job control statements.

Are we on track?

The data set requirements for a program are described to the operating system by the _____ statements.

Introduction to JCL – Job control statements.

Are we on track?

Which of the following functions can DD statements perform?

- A. Tell the system which data set to access.**
- B. Request a printer to produce the job output.**
- C. Initiate a new program.**
- D. Indicate where to store new data sets.**

Introduction to JCL – Job control statements.

Grouping programs.

When deciding which programs to group together as a single job, you may consider programs that:

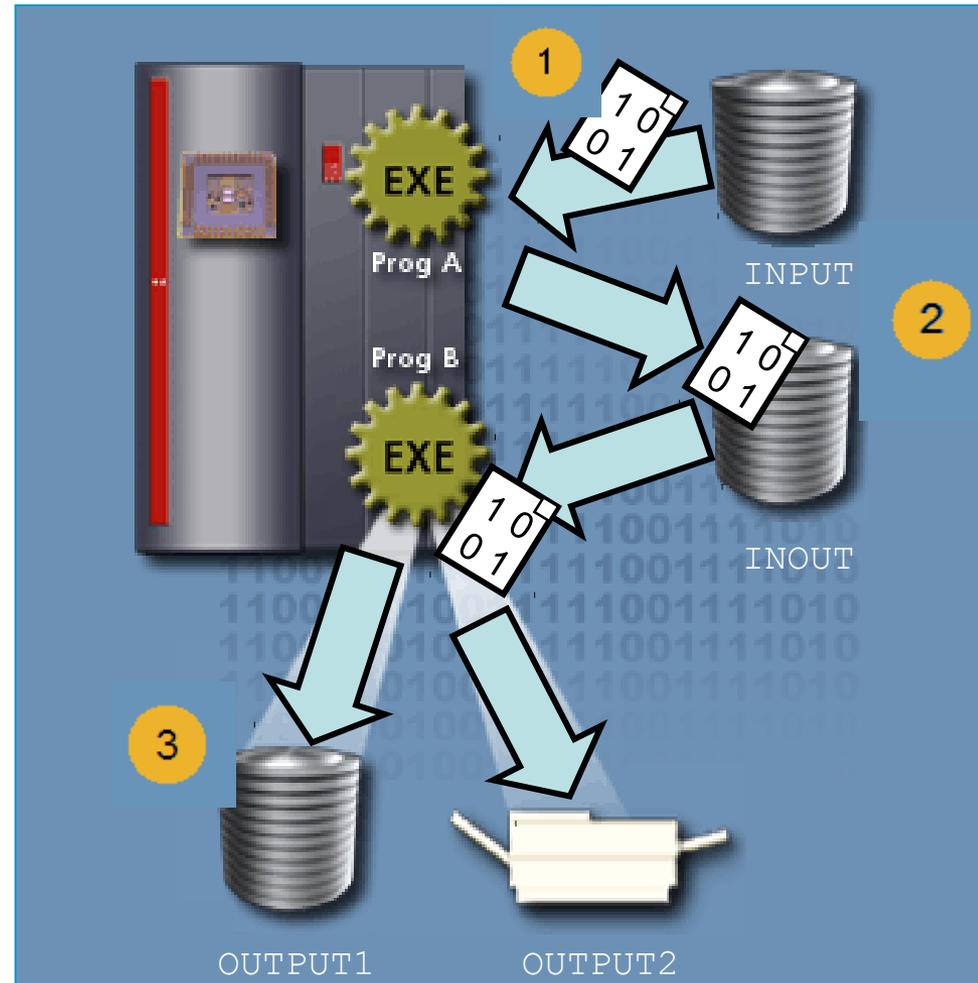
- **Have to run sequentially.**
- **Have the same job accounting information.**
- **Use the same resources, such as data sets.**
- **Are dependent on each other.**

Introduction to JCL – Job control statements.

Grouping programs.

Consider two programs:

- Program A takes input from the INPUT data set and uses it to create the INOUT data set.
- Program B uses the INOUT data set created by Program A as its input and creates two more data sets called OUTPUT1 and OUTPUT2.



Introduction to JCL – Job control statements.

Grouping programs.

The JCL for the preceding example is shown on the right.

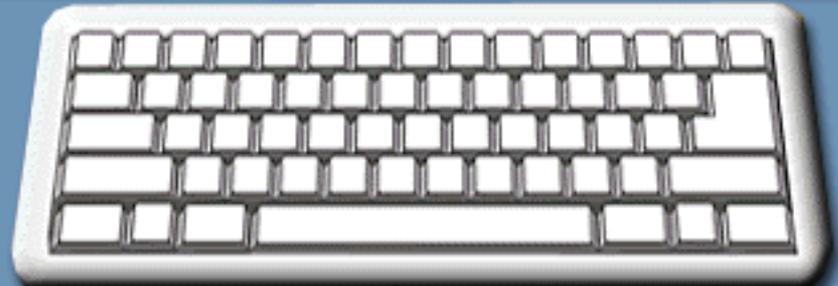
It contains two job steps:
STEP1 and STEP2.

Job step STEP1 invokes a program called PROGRAMA.

Two data definition statements are needed:

- DD1 for the INPUT data set
- DD2 for the INOUT data set

```
//JOBNUM1 JOB 504,SMITH
//STEP1 EXEC PGM=PROGRAMA
//DD1 DD DSN=INPUT,
// DISP=SHR
//DD2 DD DSN=INOUT,
// DISP=(NEW,PASS)
//STEP2 EXEC PGM=PROGRAMB
//DD3 DD DSN=INOUT,
// DISP=(OLD,CATLG)
//DD4 DD DSN=OUTPUT1,
// DISP=(NEW,CATLG,DELETE)
//DD5 DD DSN=OUTPUT2,
// DISP=(NEW,CATLG,DELETE)
```



Continued... 

Introduction to JCL – Job control statements.

Grouping programs.

In the second job step, the STEP2 EXEC invokes PROGRAMB.

Because PROGRAMB uses the output (INOUT) of PROGRAMA as input and then creates two output data sets (OUTPUT1, OUTPUT2), it needs three data definition statements:

- DD3 for the INOUT
- DD4 for the OUTPUT1
- DD5 for the OUTPUT2

```
//JOBNUM1 JOB 504,SMITH
//STEP1 EXEC PGM=PROGRAMA
//DD1 DD DSN=INPUT,
// DISP=SHR
//DD2 DD DSN=INOUT,
// DISP=(NEW,PASS)
//STEP2 EXEC PGM=PROGRAMB
//DD3 DD DSN=INOUT,
// DISP=(OLD,CATLG)
//DD4 DD DSN=OUTPUT1,
// DISP=(NEW,CATLG,DELETE)
//DD5 DD DSN=OUTPUT2,
// DISP=(NEW,CATLG,DELETE)
```

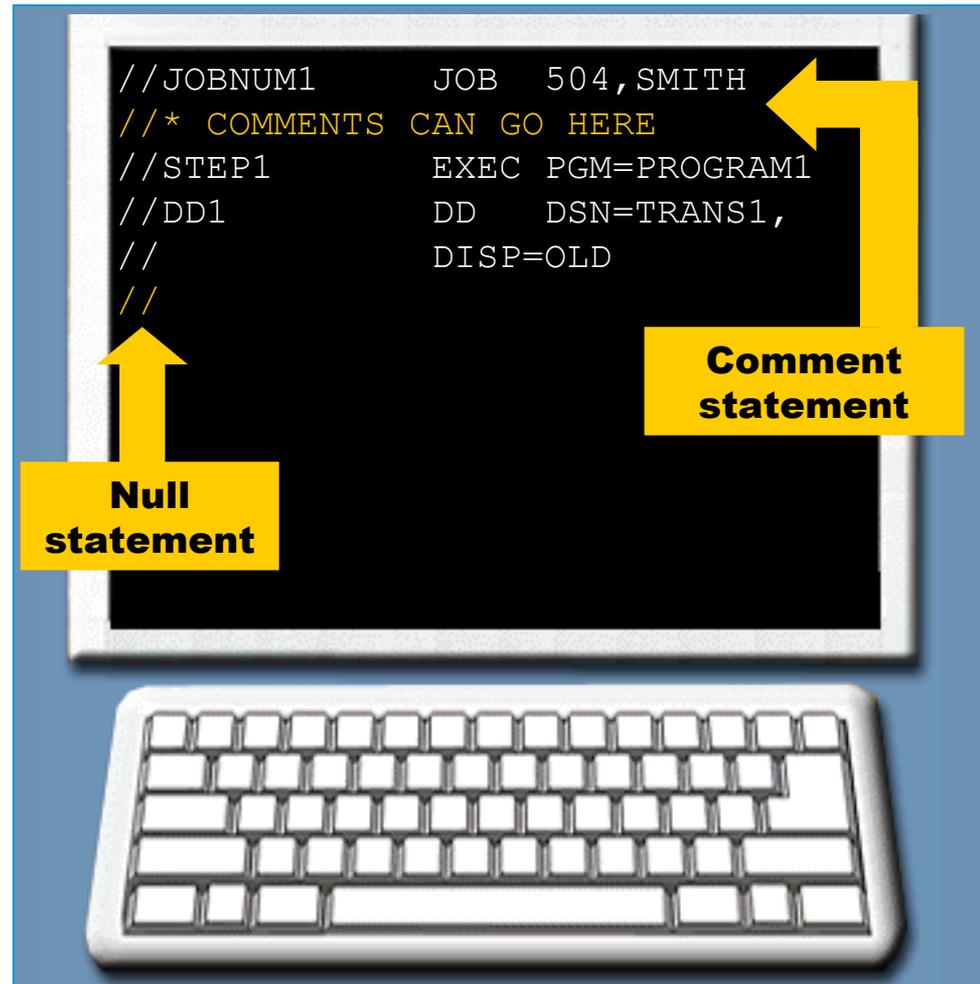


Introduction to JCL – Job control statements.

Other JCL statements.

Besides the JOB, EXEC and DD statements, there are some less frequently used JCL statements. These include:

- Comment statements: You can use these to document JCL statements.
- Null statements: You can use a null statement to mark the end of a job.



Introduction to JCL – Job control statements.

Are we on track?

Which of the following are considerations for determining which programs should be grouped together as a job?

- A. Programs must run sequentially.**
- B. Programs must be dependent on each other.**
- C. Programs must access the same databases.**
- D. Programs must have the same accounting information.**

Introduction to JCL – Job control statements.

Are we on track?

During the course of a job, a program uses the output from a previous job step as its source data, and then creates a new data set and copies it to another storage device. How many DD statements would you find in the JCL for this job step?

- A. Two**
- B. One**
- C. Three**
- D. Four**

Introduction to JCL – Job control statements.

Glossary.

Comment statements

The comment statement is used primarily to document a program and its resource requirements.

Input/Output (I/O)

Transfer of data into a computer and from the computer to the outside world.

Null

A blank statement beginning with a double slash (//) that marks the end of a job.

Introduction to JCL – JCL statement syntax.

JCL statement fields.

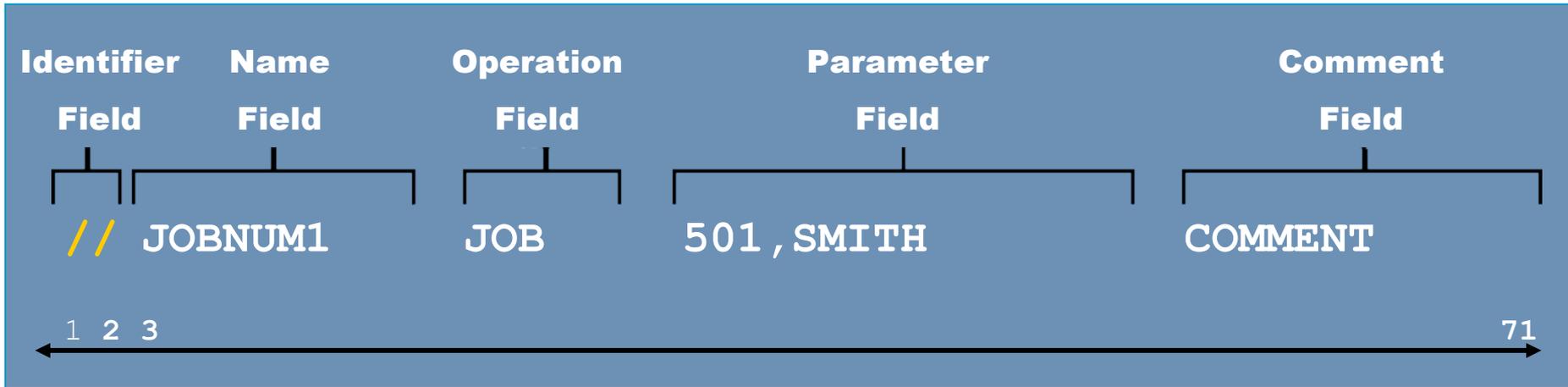
Most JCL statements have the same format consisting of five fields, which are as follows:

- **The identifier field.**
- **The name field.**
- **The operation field.**
- **The parameter field.**
- **The comment field.**

Not all control statements require all five fields.

Introduction to JCL – JCL statement syntax.

The Identifier field.

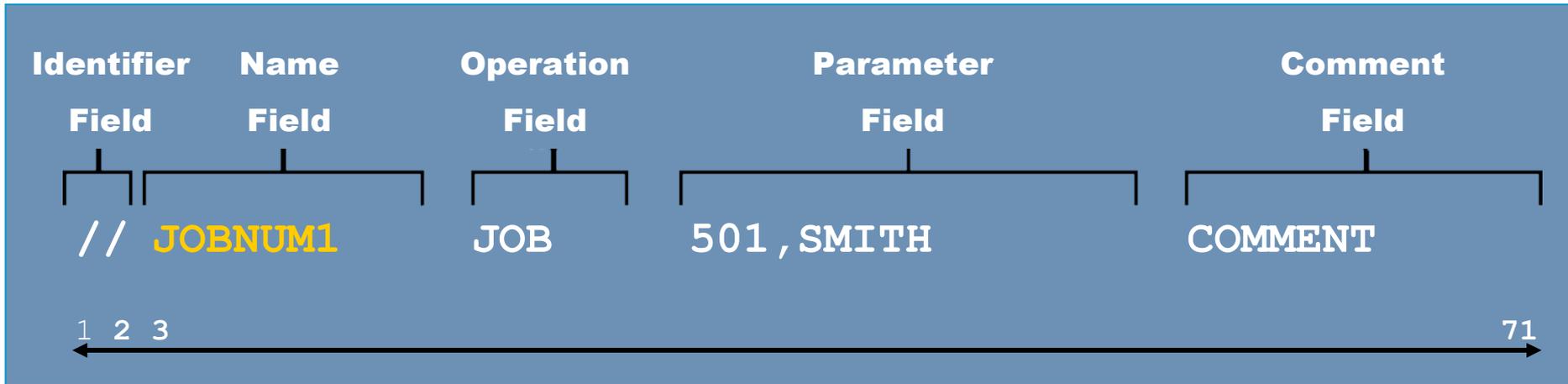


The identifier field occupies positions 1 and 2. It usually contains two forward slashes (//) and indicates that the statement is a JCL statement (as opposed to a data records).

The remaining four fields must be coded from position 3 through position 71.

Introduction to JCL – JCL statement syntax.

The Name field.



The name field must begin in position 3 with no spaces between the identifier and the name fields. It identifies the JCL statement by name so that other JCL statements or the operating system can refer to it.

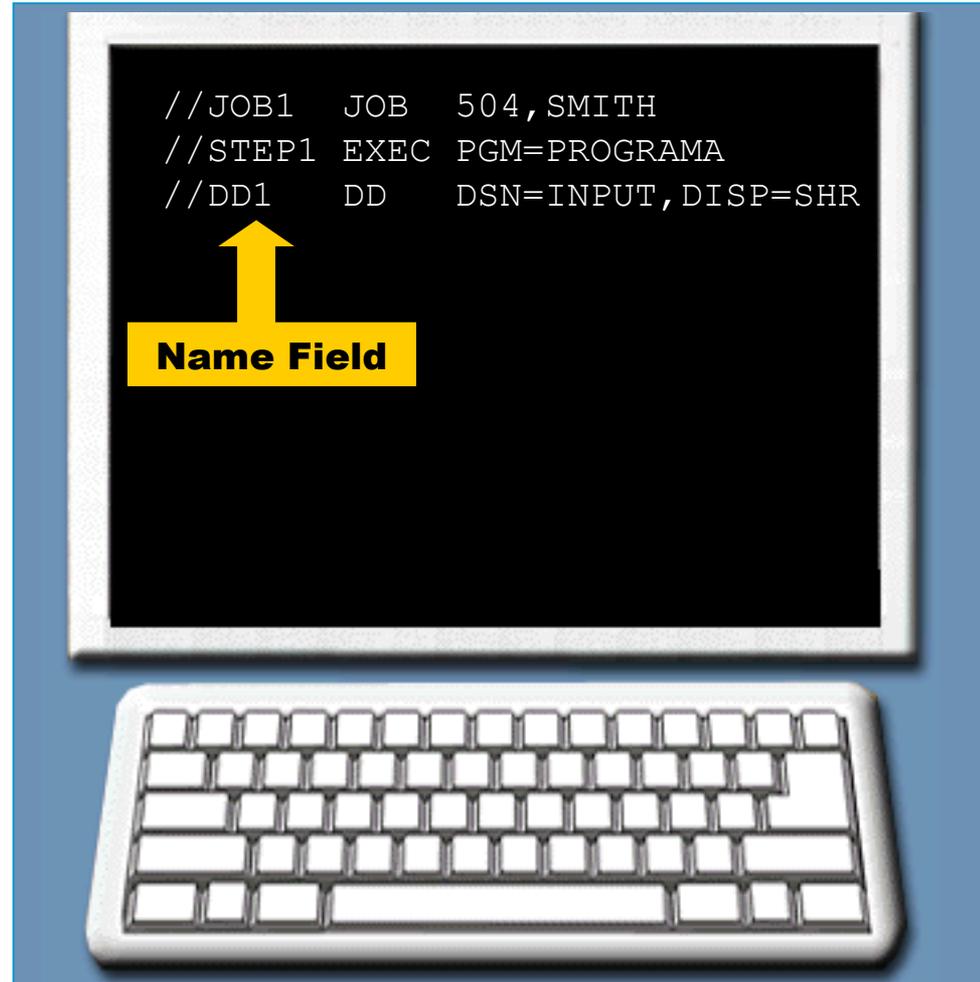
A name is optional for some EXEC statements, but is required on JOB and almost all DD statements.

Introduction to JCL – JCL statement syntax.

Rules for naming JCL statements.

The specifications for JCL statement names are as follows:

- The name must begin in position 3.
- The name can be one to eight characters in length.
- The first character in the name must be alphabetic character (A to Z) or national character (#, @, \$).
- The first character cannot be numeric (0 through 9).

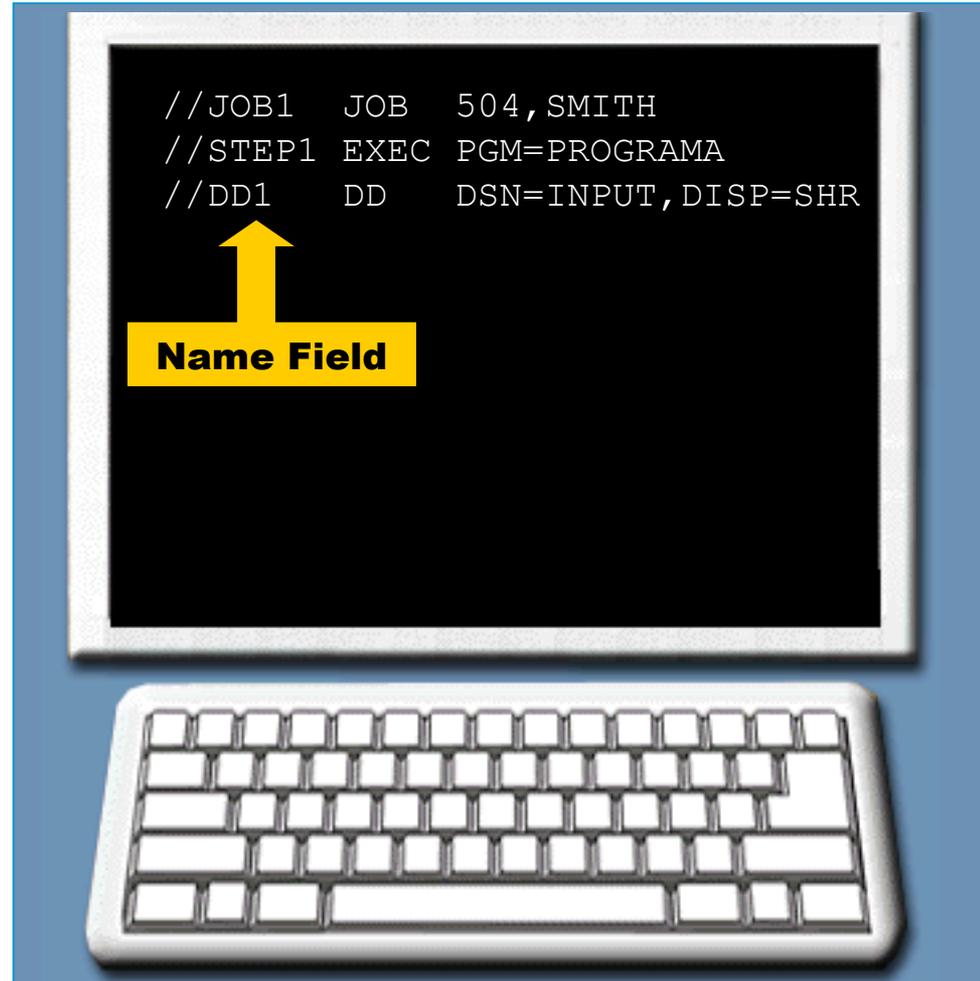


Introduction to JCL – JCL statement syntax.

Rules for naming JCL statements.

The specifications for JCL statement names are as follows:

- The remaining characters can be alpha numeric (A to Z and 0 through 9) or national characters.
- No special characters or spaces can be included in a name.



Introduction to JCL – JCL statement syntax.

Are we on track?

Match each JCL field types with their main function:

- | | |
|---------------------|--|
| 1. Parameter | A. Contains documentation information from the JCL coder. |
| 2. Comments | B. Specifies the type of statement. |
| 3. Operation | C. Contains specific statement values. |
| 4. Name | D. Identifies a JCL statement. |

Introduction to JCL – JCL statement syntax.

Are we on track?

Which of the following is a valid JOB name?

A. //TEST*1

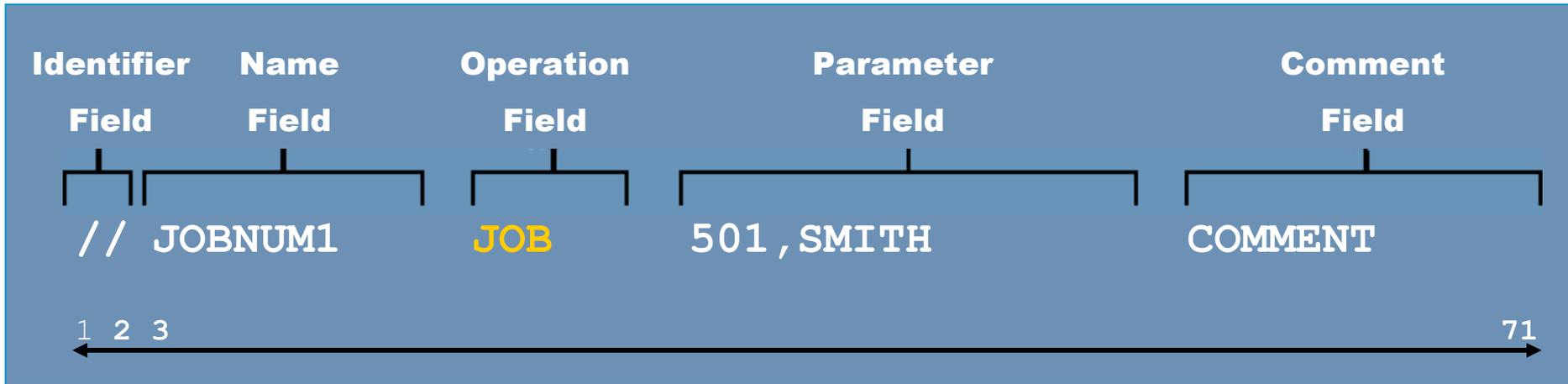
B. // JOB13

C. //\$ABCD

D. //STEPSEVEN

Introduction to JCL – JCL statement syntax.

The Operation field.



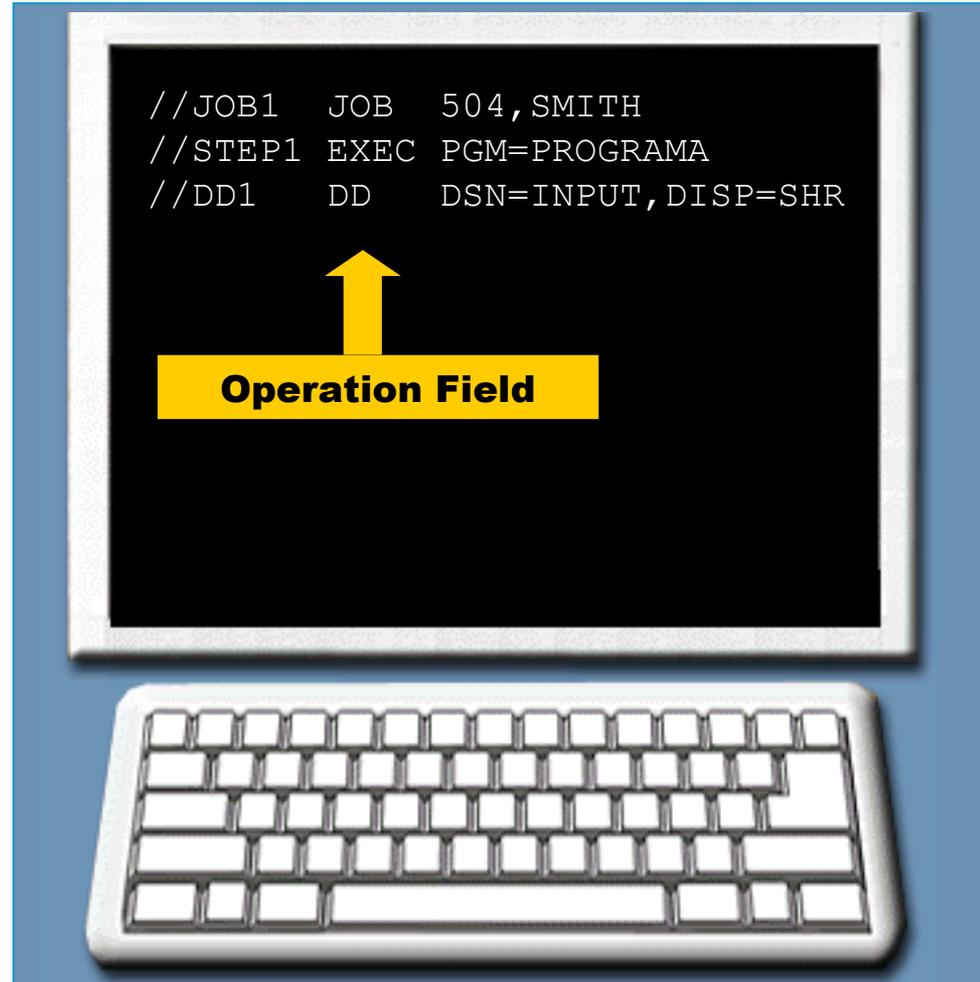
The operation field defines the JCL statement type - JOB, EXEC, or DD.

For every JCL statement, the operation field must follow the name field and it must be preceded by at least one blank space.

Introduction to JCL – JCL statement syntax.

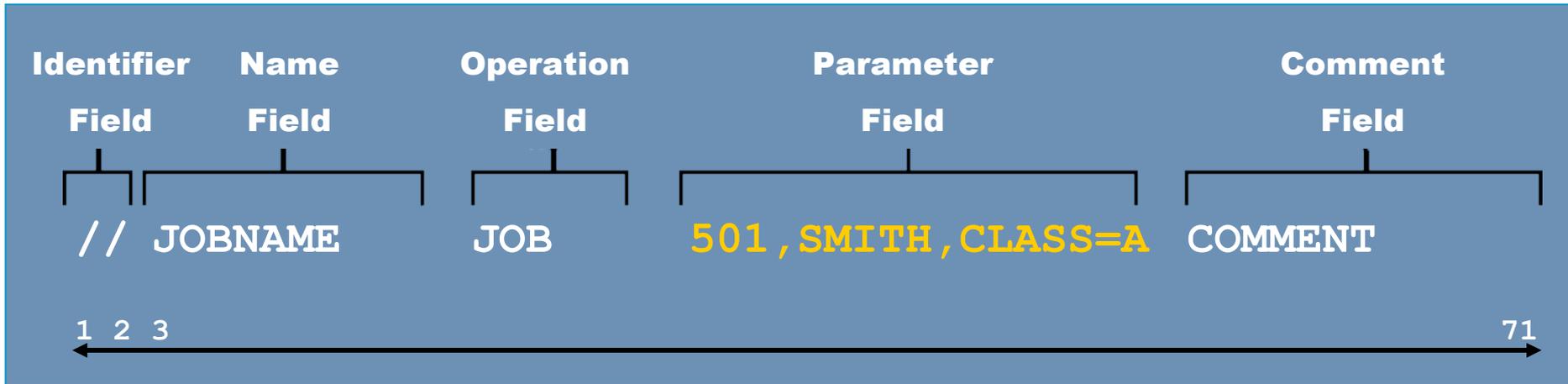
Operation fields: an example.

In the JCL statements that you have seen so far, the operation fields contain the operators JOB, EXEC, and DD.



Introduction to JCL – JCL statement syntax.

The Parameter field.



The parameter field contains one or more parameters separated by commas. Each JOB, EXEC and DD statement has its own set of parameter information.

The parameter field of each JCL statement must:

- Follow the operation field.
- Include one or more blanks between it and the operation field.
- Use commas to separate parameters.
- Have no blanks between parameters.

Introduction to JCL – JCL statement syntax.

Parameter fields: an example.

The JOB contains:

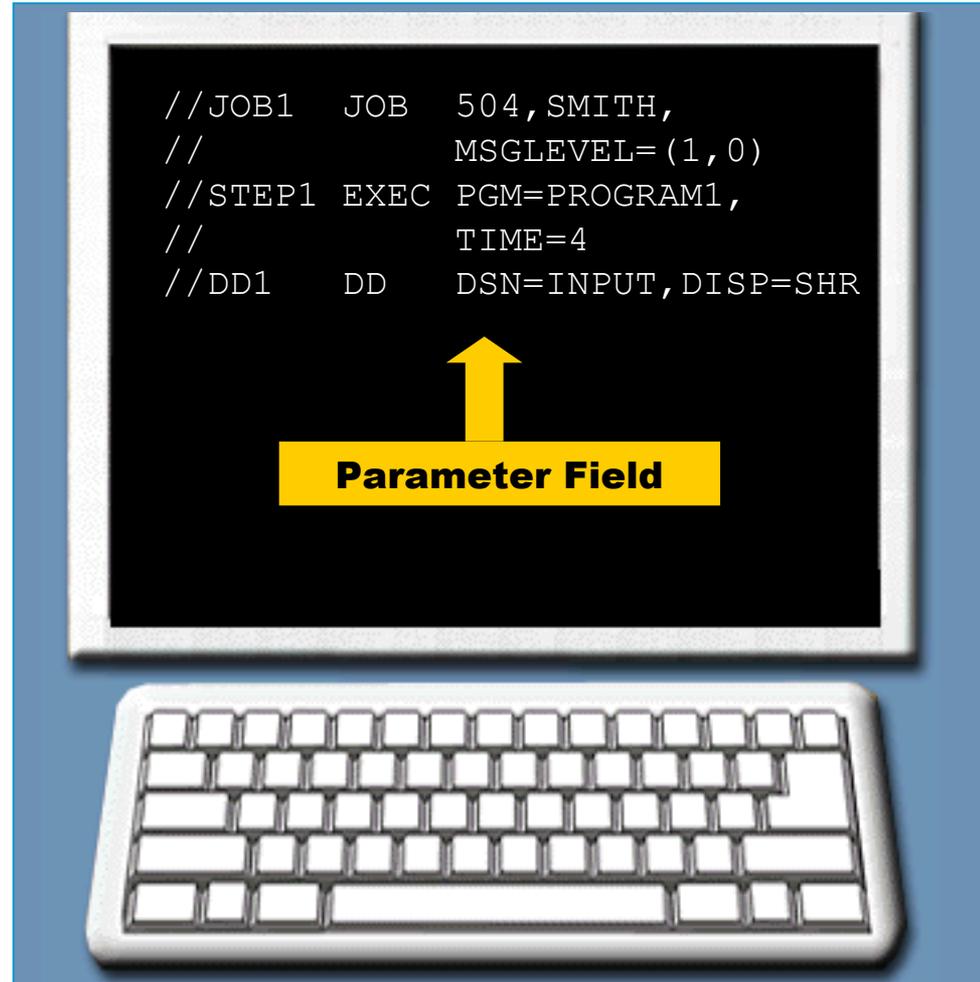
- Accounting information (504).
- The JCL programmer's name (SMITH).
- MSGLEVEL (message level) parameter.

The EXEC contains:

- The first one specifies the program name (PROGRAM1).
- The second one specifies the maximum CPU time for the program (4 minutes).

The DD contains:

- The first one specifies the data set name INPUT.
- The second one specifies a status of SHR (shared).



Introduction to JCL – JCL statement syntax.

Are we on track?

Place the following fields in the order in which they appear in a JCL statement (from first to last).

A. Name

B. Comment

C. Operation

D. Parameter

E. Identifier

Introduction to JCL – JCL statement syntax.

Are we on track?

Match the following parameters with their descriptions:

- | | |
|---------------|--------------------------------------|
| 1. SMITH | A. Maximum run time for the program. |
| 2. TIME=4 | B. Data set name. |
| 3. DSN=TRANS1 | C. Programmer's name. |

Introduction to JCL – JCL statement syntax.

Continuation statement.

You can continue a JCL statement on one or more separate lines if:

- **The JCL statement requires more than 71 positions to specify all the necessary parameters.**
- **You want to make it easier to read all the parameters. For example: You could code one parameter per line.**

Introduction to JCL – JCL statement syntax.

Procedure for continuing a JCL statement.

The procedure for coding a JCL statement on more than one line is as follows:

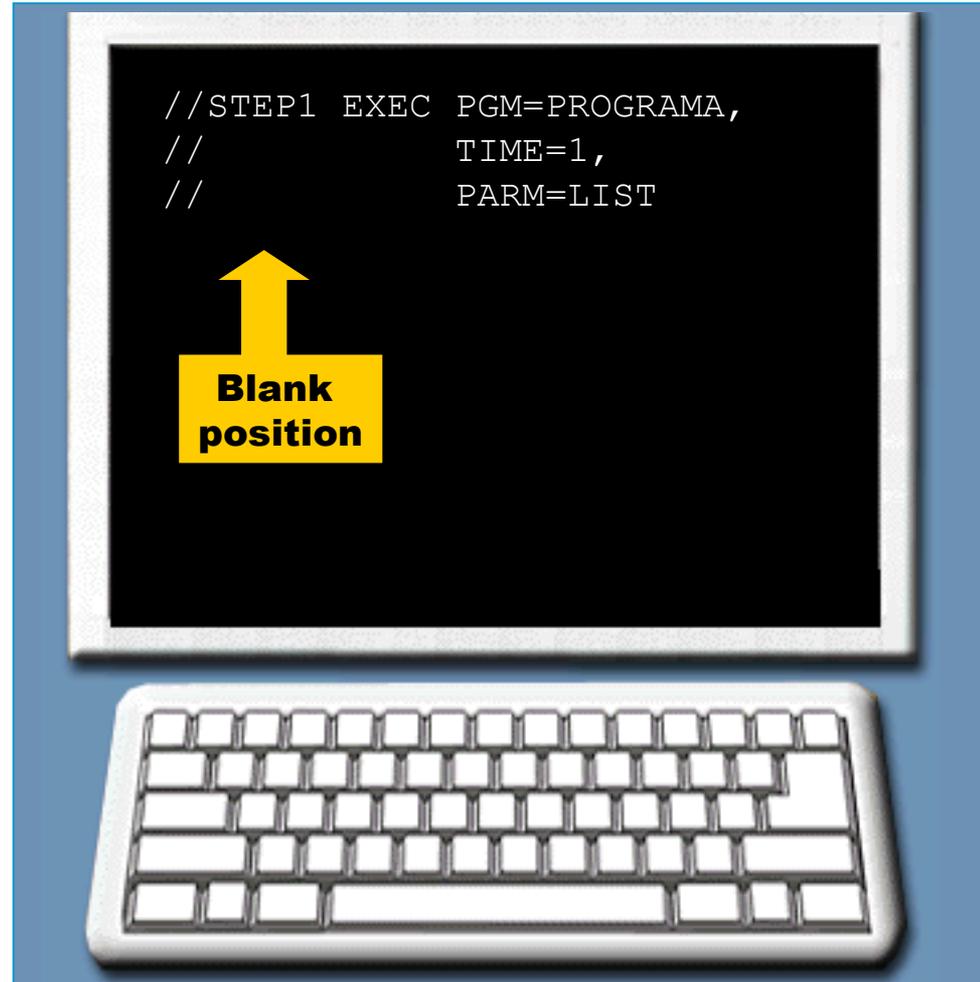
- 1. Interrupt the coding of a field after a complete parameter or subparameter, including the comma that follows it, before reaching position 72.**
- 2. Code // in positions 1 and 2 of the following statement.**
- 3. Continue the parameters of the interrupted field, beginning in any position from 4 to 16.**

Introduction to JCL – JCL statement syntax.

Procedure for continuing a JCL statement.

Following is an example of a single JCL statement that continues on 2 additional lines. All 3 lines comprise one JCL statement:

1. In the parameter field of the JCL statement, end the line before reaching position 72, but after coding a complete parameter and the comma that follows it.



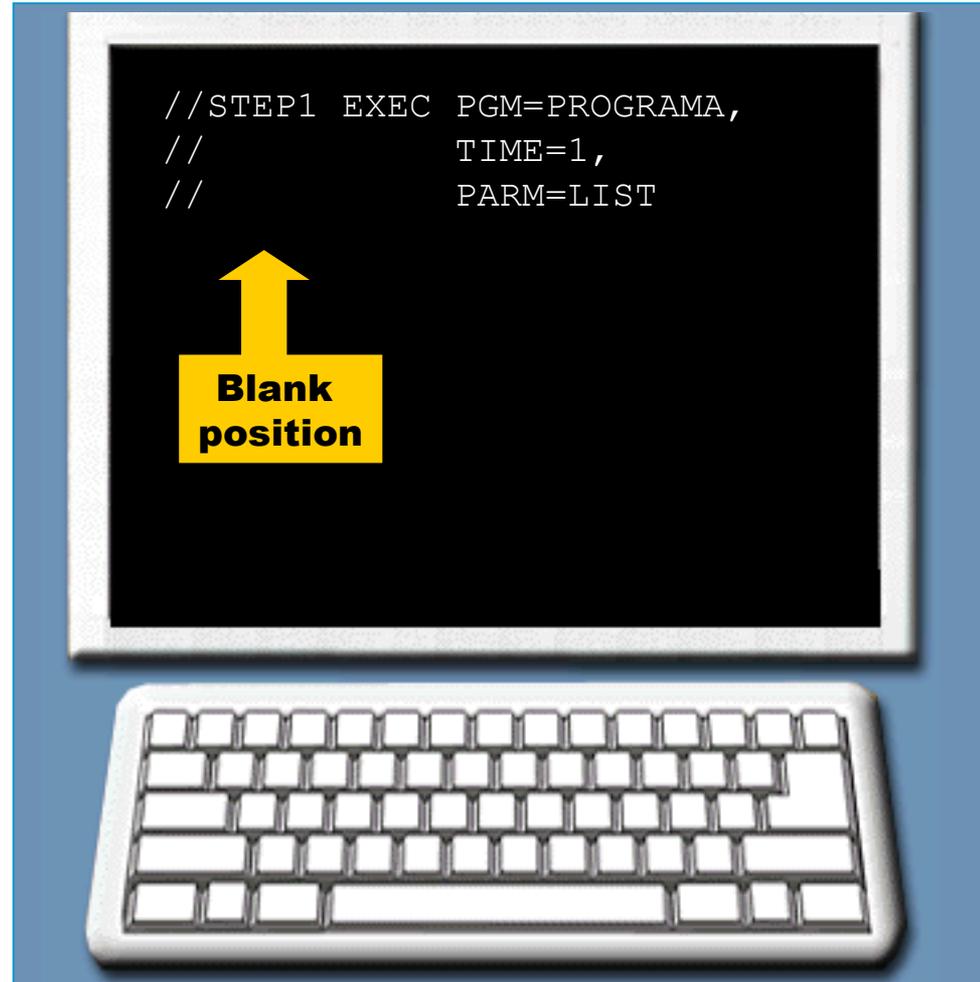
Continued... 

Introduction to JCL – JCL statement syntax.

Procedure for continuing a JCL statement.

Following is an example of a single JCL statement that continues on 2 additional lines. All 3 lines comprise one JCL statement:

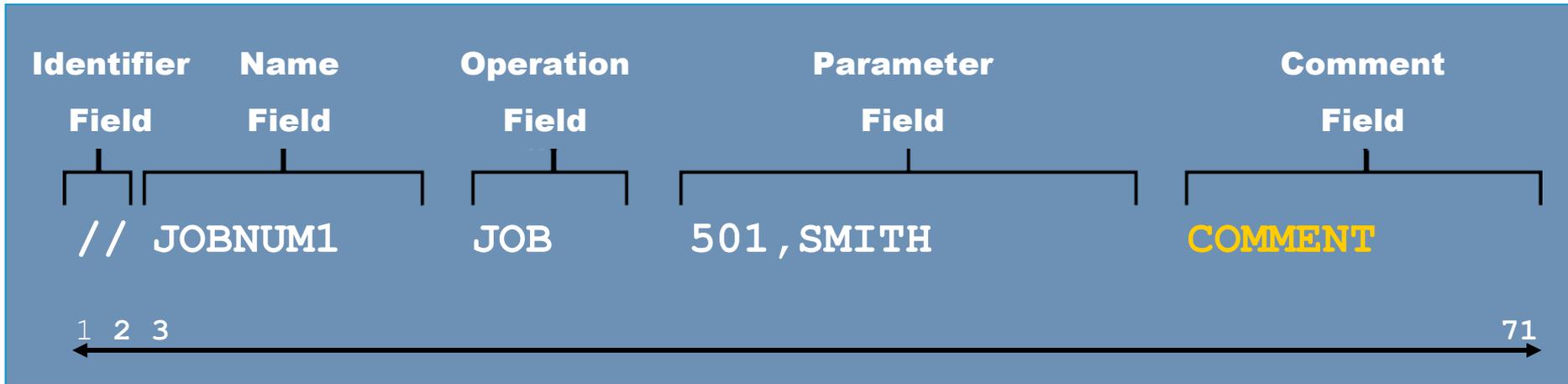
2. On the next line, code two forward slashes (//) in positions 1 and 2, and a blank in position 3.
3. Beginning in position from 4 to 16, resume coding the JCL statement with the next complete parameter.



Continued... 

Introduction to JCL – JCL statement syntax.

The Comment field.



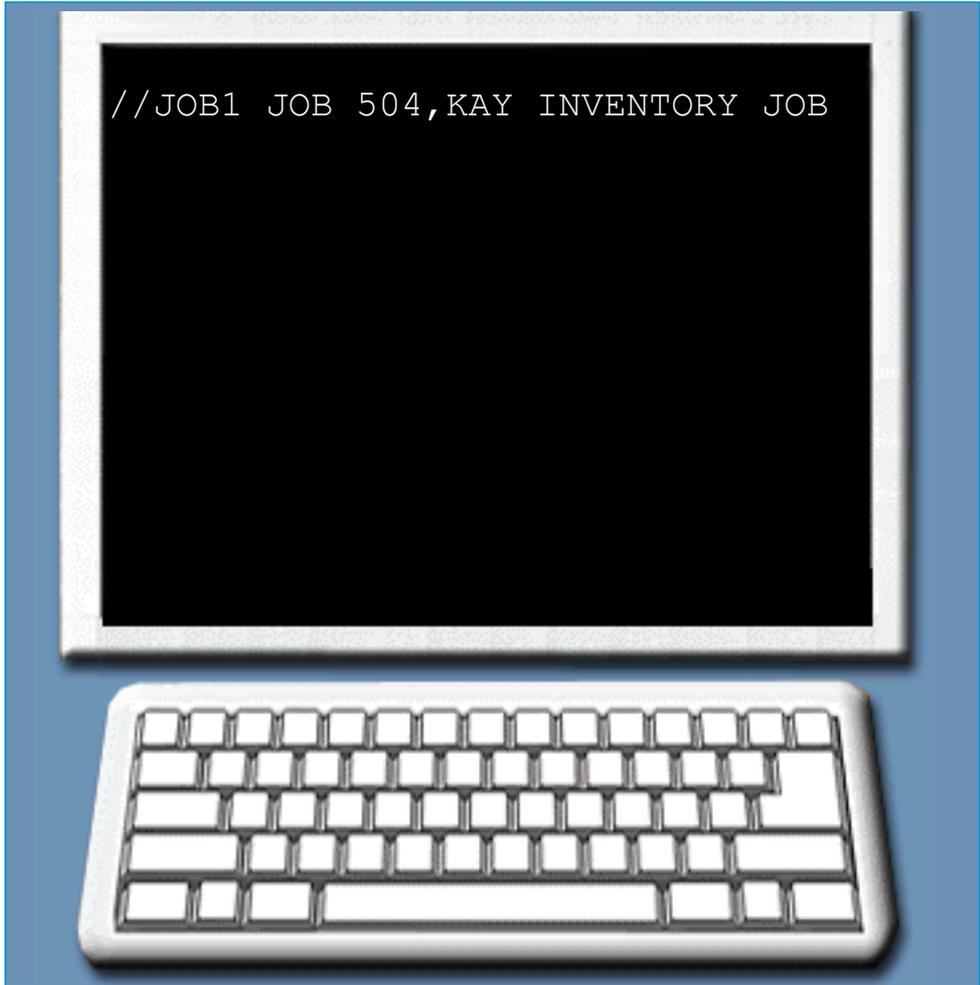
A comment field contains extra documentation which you may choose to include. It contains documentation relating to the code or other messages from the programmer.

A comment field must follow the parameter field and must be preceded by one or more blanks.

Introduction to JCL – JCL statement syntax.

Comment field: an example.

The following example shows a JCL statement that contains a comment 'INVENTORY JOB'. A blank separates the parameter field from the words INVENTORY JOB.



```
//JOB1 JOB 504,KAY INVENTORY JOB
```

Continued... 

Introduction to JCL – JCL statement syntax.

Are we on track?

Place the following JCL statements in the correct order.

A. //STEP1 EXEC PGM=PROG1

B. //PAY JOB 03948,FRASER=(1,0)

C. //XYZ DD DSN

Introduction to JCL – JCL statement syntax.

Rules for writing JCL statements.

You should remember the following rules while writing a JCL statement:

- **The identifying characters (//) must be in positions 1 and 2.**
- **The name field must begin in position 3.**
- **Each subsequent field must be preceded by one or more spaces.**
- **A continuation line begins with //.**
- **Continuing parameters must start in positions 4 through 16.**
- **Do not embed blanks within any field. For example, do not code a blank to separate parameters in the parameter field.**
- **Do not code past position 71.**

Introduction to JCL – JCL statement syntax.

Glossary.

Alphabetic Characters
Capital A to capital Z.

Comment Field

The comment field contains documentation relating to the code or other messages from the programmer.

Identifier Field

The identifier field indicates to the system that a JCL statement rather than data.

Name Field

The name field identifies a particular statement so that other statements and the system can refer to it.

National Characters

The national characters are the at sign [@], the dollar sign [\$], or the pound sign [#].

Introduction to JCL – JCL statement syntax.

Glossary.

Numeric Characters

Numbers 0 through 9.

Operation Field

The operation field specifies the type of statement, or, for the command statement, the command.

Parameter Field

The parameter field contains statement parameters separated by commas.

Special Characters

These include ampersand [&]; exclamation mark [!]; asterisk [*]; comma [,]; equal sign [=]; hyphen [-]; period [.]; plus sign [+]; slash [/]; left parenthesis [(]; right parenthesis [)]

Introduction to JCL – JCL statement syntax.

Unit summary.

Now that you have completed this unit, you should be able to:

- **Identify the three major JCL statements and the purpose of each.**
- **Arrange JCL statements for a specific sequence of execution.**
- **Select three major considerations for grouping work into jobs.**
- **Arrange JCL statement field into the correct format.**
- **Identify properly coded JCL statements.**
- **Depending on the type of JCL statement, the parameter field may contain positional parameters, keyword parameters, and a subparameters.**

JCL

Chapter a1 Introduction to JCL

See „z/OS MVS JCL Reference“ manual.

Job Control Language

Chapter a1. Introduction to JCL

Chapter a2. Coding JOB statements

Chapter a3. Coding EXEC statements

Chapter a4. Coding DD statements

Chapter a5. Analyzing job output

Chapter a6. Conditional processing

Job Control Language

Chapter b1. Using special DD statements

Chapter b2. Introducing procedures

Chapter b3. Modifying EXEC parameters

Chapter b4. Modifying DD parameters

Chapter b5. Determining the effective JCL

Chapter b6. Symbolic parameters

Job Control Language

Chapter c1. Nested procedures

Chapter c2. Cataloging procedures

Chapter c3. Using utility programs

Chapter c4. Sample utility application

Introduction to JCL

Chapter a1

Introduction to JCL

5

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Introduction to JCL

Course objectives.

Be able to:

- **Identify JCL statements and their purpose.**
- **Code a JOB statement.**
- **Code an EXEC statement to invoke a program.**
- **Code a DD statement to identify a data set.**
- **Code the library DD statements JOBLIB and STEPLIB.**
- **Identify the JCL statement errors in an output listing and code appropriate corrections.**
- **Code the IF/THEN/ELSE/ENDIF JCL statement construct.**

Introduction to JCL – Introduction.

Job Control Language.

What is JCL?

Job Control Language (JCL) is a control language used to identify a job to an operating system and to describe the job's requirements.

JCL is used to describe the work you want a system using multiple virtual storage (MVS) to perform.

The three main types of JCL statements are:

- **JOB - It marks the beginning of a job and identifies the job name.**
- **EXEC - It marks the beginning of a job step and specifies the name of the program (or procedure) to be executed.**
- **DD - It describes data sets to be used within individual steps.**

JCL consists of a series of statements, each of which provides specific instructions or information for batch processing jobs. These statements tell z/OS where to find the appropriate input, how to process that input (that is, what program or programs to run), and what to do with the resulting output.

Introduction to JCL – Introduction.

Job Control Language.

What is a job?

A job is a unit of work defined by a user to be accomplished by a computer, including computer programs, files or control statements.

For example: The request to perform a task of updating one data set with information from another data set is called a job.

JCL provides the operating system with information about the resources, programs, and data necessary to complete the job.

Introduction to JCL – Introduction.

Jobs – an example.



Example: A job may direct the computer to run a program that combines two data sets into one new data set.

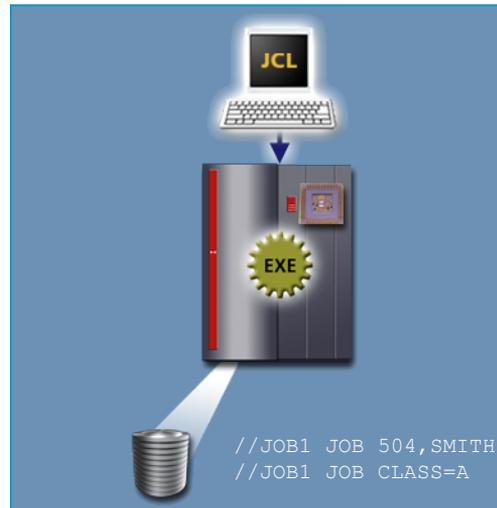
A *job* is a separately executable collection of work defined by a user, and run by a computer. This representation of a collection of work consists of one program or a set of programs, files, and control statements.

Introduction to JCL – Introduction.

JCL job information.

JCL provides information to the system about job details including:

- The programs to execute.
- The location of the required data sets.
- The department to be billed for CPU processing time.
- The job class.



Introduction to JCL – Introduction.

Job stream.

One or more jobs placed in a series and ready to be submitted to the system are referred to as a job stream.

This series of jobs is entered through one input device. You can use JCL to structure a job stream.

The JCL for each job begins with a JOB statement.



11

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

A *job stream*, or *input stream*, consists of one or more jobs that are submitted to the system in a sequence. On z/OS, you can submit jobs to the system through several facilities, including the TSO/E and its menu-driven interface ISPF.

See MCOE.EDU.JCL.JCL(JOBSTR).

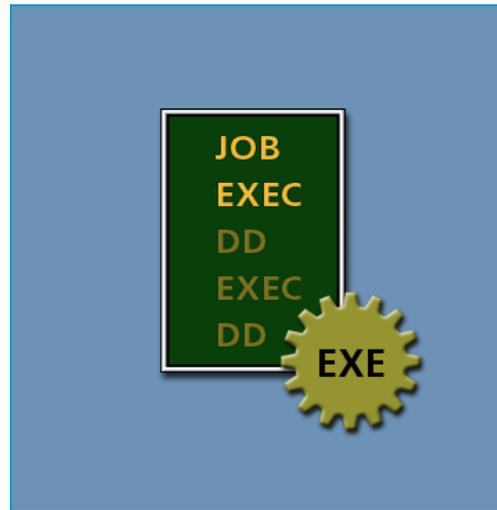
Introduction to JCL – Introduction.

Job steps.

A job consists of one or more individual job steps.

Each job step starts with an EXEC statement and is associated with the execution of one program (or procedure).

The data sets required for each job step can be defined using a DD statement.



12

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

You then submit each job for processing as part of the job stream.

A job is identified by a JOB statement. Each job consists of one or more individual job steps. You define each job step with an EXEC statement, which is associated with the execution of one program or procedure. You define the data sets required for each job step by using a DD statement.

Introduction to JCL – Introduction.

Are we on track?

The JCL specifying a collection of jobs to be run is called a(n)

13

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is Job stream.

Glossary.

Batch

A group of records or data processing jobs brought together for processing or transmission.

Central Processing Unit (CPU)

The CPU controls the operation of a computer. Units within the CPU perform arithmetic and logical operations and decode and execute instructions.

Data set

Synonym for file. A unit of information that can be stored and retrieved.

DD statement

A JCL statement describing a data set associated with a specific job.

Glossary.

EXEC statement

Marks the beginning of a job step; assigns a name to the step; identifies the program or procedure to be executed in this step.

Job

A unit of work defined by a user to be accomplished by a computer, including computer programs, files or control statements.

Job step

The JCL statements that request and control the execution of a program and request the resources needed to run the program. A job step is identified by an EXEC statement.

Job stream

Jobs placed in a series and entered through one input device.

Multiple Virtual Storage (MVS)

The standard operating system for older IBM mainframes.

Introduction to JCL – Job control statements.

Defining Job control statements.

Job control statements define jobs and job steps.

JCL statements provide the following information:

- **Programs to be executed.**
- **Sequence of program execution.**
- **Data sets required by the programs.**

JCL statements may provide other kinds of information as well. For example, a JCL statement can specify how the system should process a job by providing accounting information.

Introduction to JCL – Job control statements.

Defining Job control statements.

Each job requires you to specify at least the following:

- Job to begin (using a JOB statement).
- Programs or procedures to be executed (using EXEC statements).
- Data sets to be used (using DD statements).



17

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

You can also add a null statement to mark the end of a job and comment statements to document your JCL statements.

In fact – it is not necessary to specify DD statement, there is no JCL error if you omitt DD.

Introduction to JCL – Job control statements.

Are we on track?

You would use the _____ statement to run a program as part of a job.

18

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

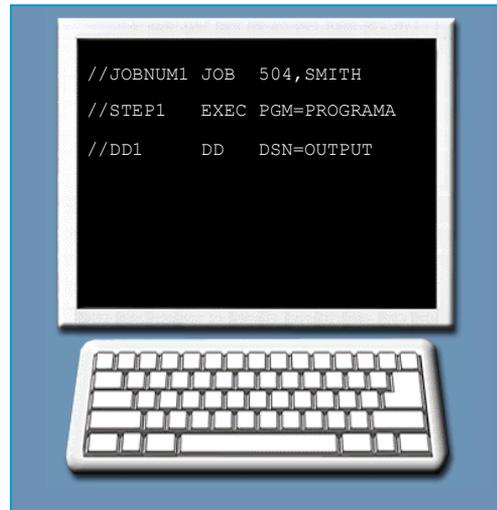
The correct answer is 'EXEC'.

Introduction to JCL – Job control statements.

The JOB statement.

The JOB statement is the first control statement in a job. It marks the beginning of a job and specifies the job name.

The JOB statement may also provide details and parameters that relate to the overall job, such as accounting information and conditions for job termination.



Continued... ➡

19

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

The JOB statement typically provides accounting information and details about the person who is submitting the job, including name, user ID, and password. This information applies to all job steps within the job. The JOB statement also may contain any comments that help describe the statement.

In this diagram, the first JOB statement, named JOBNUM1, includes the accounting number (504) and programmer name (SMITH).

Introduction to JCL – Job control statements.

The JOB statement.

Each job must begin with a JOB statement. All statements that follow up to the next JOB statement, are parts of one job.

The end of a job is marked by either another JOB statement or by a null statement.



Null statement is a blank statement beginning with a double slash - // that marks the end of a job.

Introduction to JCL – Job control statements.

Are we on track?

Which of the following is a function of the JOB statement?

- A. Contains documentation information only.**
- B. Identifies the program to be run.**
- C. Identifies the data sets to be used during the job.**
- D. Identifies the job name and specifies accounting information.**

21

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

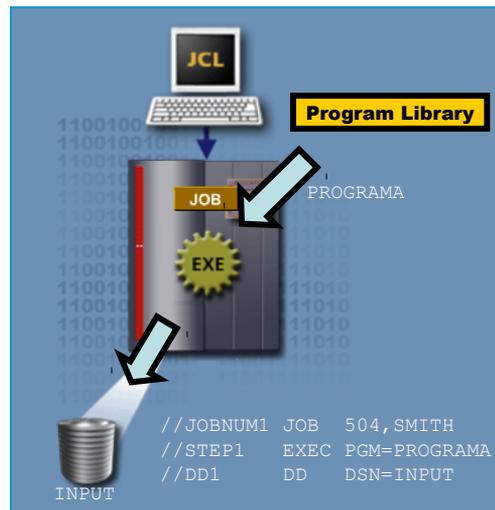
The correct answer is D.

Introduction to JCL – Job control statements.

The EXEC statement.

An EXEC statement often follows the JOB statement and identifies a program or procedure to be executed.

A job contains one or more EXEC statements, each followed by statements that define one or more data sets needed for that job step.



22

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Continued... ➡

A job step ends with the next EXEC statement, the JOB statement for the next job, or a null statement.

Procedures are named collections of partial JCL, usually one or more EXEC statements and DD statements, that perform frequently used functions, such as sorting data.

Programs and cataloged procedures are stored in specific data sets, which are called program or procedure libraries, respectively. They are retrieved by the operating system when they are referenced by an EXEC statement.

Introduction to JCL – Job control statements.

Are we on track?

Where are programs referenced by an EXEC statement stored?

- A. Sequential data sets.**
- B. Program libraries.**
- C. Tape devices.**
- D. Procedure libraries.**

23

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is B.

Introduction to JCL – Job control statements.

The DD statement.

DD statements define the data sets the program uses during execution. Each data set used in the job requires at least one DD statement.

The number of DD statements following an EXEC statement depends on the number of data sets required by a program.

The order of DD statements within a job step is not usually significant.



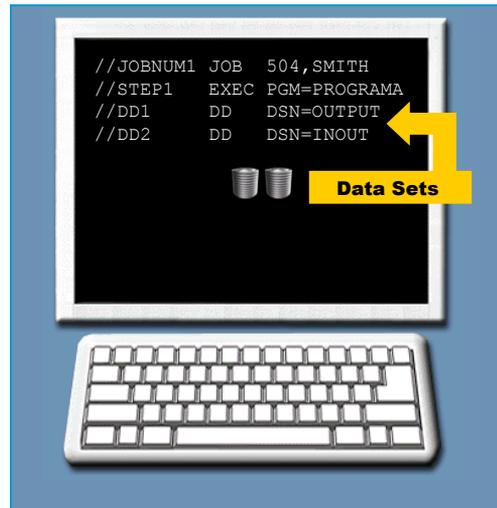
You also can use DD statements to request I/O devices or specify storage allocation for new data sets.

Introduction to JCL – Job control statements.

The DD statement.

The DD statement describes the data set extensively and can include the following information:

- Give the data set name.
- Specify the initial status and final disposition of the data set.
- Specify volumes.
- Specify record length, format, and blocking.
- Request I/O devices.
- Specify storage allocation for new data sets.



25

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Continued... ➡

A DD can include the following information:

- The name that the program uses to refer to the data set, known as the ddname.
- The actual name of the data set and its location.
- Physical characteristics of the data set, such as record format.
- The initial and final status of the data set, known as its disposition.

Introduction to JCL – Job control statements.

Are we on track?

The data set requirements for a program are described to the operating system by the _____ statements.

26

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is DD.

Introduction to JCL – Job control statements.

Are we on track?

Which of the following functions can DD statements perform?

- A. Tell the system which data set to access.**
- B. Request a printer to produce the job output.**
- C. Initiate a new program.**
- D. Indicate where to store new data sets.**

27

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A., B., and D.

Introduction to JCL – Job control statements.

Grouping programs.

When deciding which programs to group together as a single job, you may consider programs that:

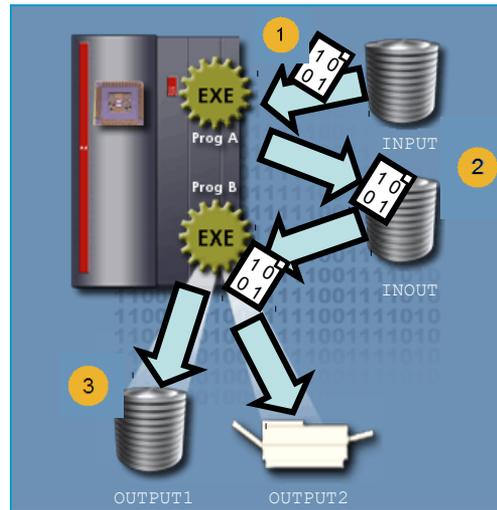
- **Have to run sequentially.**
- **Have the same job accounting information.**
- **Use the same resources, such as data sets.**
- **Are dependent on each other.**

Introduction to JCL – Job control statements.

Grouping programs.

Consider two programs:

- Program A takes input from the INPUT data set and uses it to create the INOUT data set.
- Program B uses the INOUT data set created by Program A as its input and creates two more data sets called OUTPUT1 and OUTPUT2.



29

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Can you group Program A and Program B?

Since these programs must run sequentially and Program B is dependent on Program A, they can be grouped into a single job.

Introduction to JCL – Job control statements.

Grouping programs.

The JCL for the preceding example is shown on the right.

It contains two job steps:
STEP1 and STEP2.

Job step STEP1 invokes a program called PROGRAMA.

Two data definition statements are needed:

- DD1 for the INPUT data set
- DD2 for the INOUT data set



Continued... ➡

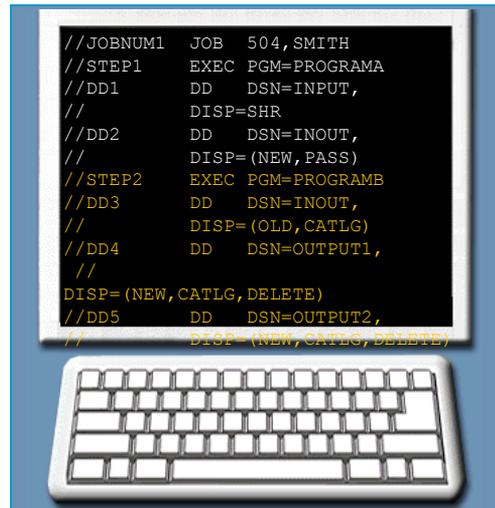
Introduction to JCL – Job control statements.

Grouping programs.

In the second job step, the STEP2 EXEC invokes PROGRAMB.

Because PROGRAMB uses the output (INOUT) of PROGRAMA as input and then creates two output data sets (OUTPUT1, OUTPUT2), it needs three data definition statements:

- DD3 for the INOUT
- DD4 for the OUTPUT1
- DD5 for the OUTPUT2

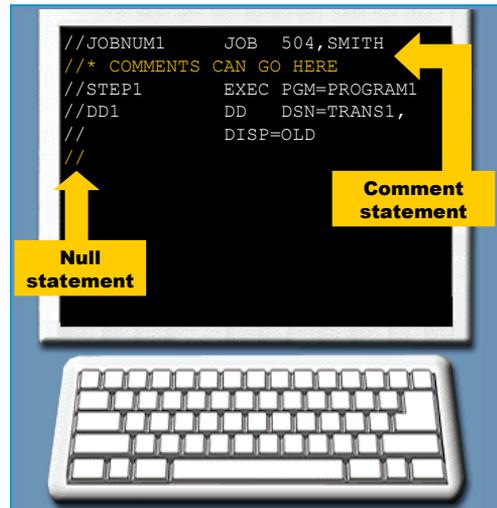


Introduction to JCL – Job control statements.

Other JCL statements.

Besides the JOB, EXEC and DD statements, there are some less frequently used JCL statements. These include:

- Comment statements: You can use these to document JCL statements.
- Null statements: You can use a null statement to mark the end of a job.



32

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

Comment statements:

- A comment statement is coded with two forward slashes and an asterisk (//*) in the first 3 positions. The comment text is coded in positions 4 through 71.

Null statements:

- It is coded with two forward slashes (//) in positions 1 and 2 and no other characteristics.

Introduction to JCL – Job control statements.

Are we on track?

Which of the following are considerations for determining which programs should be grouped together as a job?

- A. Programs must run sequentially.**
- B. Programs must be dependent on each other.**
- C. Programs must access the same databases.**
- D. Programs must have the same accounting information.**

33

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is A, B and D.

Introduction to JCL – Job control statements.

Are we on track?

During the course of a job, a program uses the output from a previous job step as its source data, and then creates a new data set and copies it to another storage device. How many DD statements would you find in the JCL for this job step?

- A. Two**
- B. One**
- C. Three**
- D. Four**

The correct answer is C.

Introduction to JCL – Job control statements.

Glossary.

Comment statements

The comment statement is used primarily to document a program and its resource requirements.

Input/Output (I/O)

Transfer of data into a computer and from the computer to the outside world.

Null

A blank statement beginning with a double slash (//) that marks the end of a job.

For every job that you submit, you need to tell z/OS where to find the appropriate input, how to process that input (that is, what program or programs to run), and what to do with the resulting output. You use JCL to convey this information to z/OS through a set of statements. Most jobs can be run using a very small subset of these control statements.

Introduction to JCL – JCL statement syntax.

JCL statement fields.

Most JCL statements have the same format consisting of five fields, which are as follows:

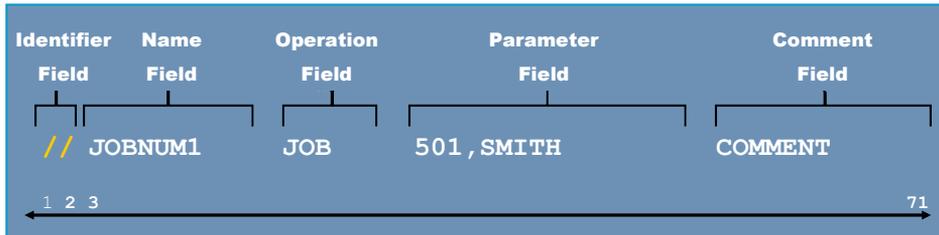
- **The identifier field.**
- **The name field.**
- **The operation field.**
- **The parameter field.**
- **The comment field.**

Not all control statements require all five fields.

This learning module describes the elements of and syntax rules for coding JCL statements.

Introduction to JCL – JCL statement syntax.

The Identifier field.



The identifier field occupies positions 1 and 2. It usually contains two forward slashes (//) and indicates that the statement is a JCL statement (as opposed to a data records).

The remaining four fields must be coded from position 3 through position 71.

37

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The *identifier* field tells the z/OS system what type of information to expect on a record in the JCL. Almost all JCL statements begin with an identifier field that contains the characters // (double slash).

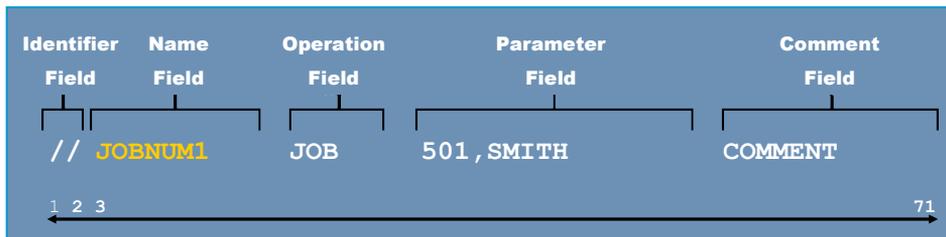
Two exceptions to this syntax rule are the delimiter statement, and comment statements:

- The delimiter statement contains one forward slash and an asterisk (the symbols /*) in the first two positions. A delimiter statement marks the end of data.
- A comment statement contains two forward slashes and an asterisk (the symbols /**) in the first 3 positions. The comment text is coded in positions 4 through 71.

Another JCL statement, the null statement, consists of only the two forward slashes (the symbols //) in the identifier field. The null statement marks the end of a job.

Introduction to JCL – JCL statement syntax.

The Name field.



The name field must begin in position 3 with no spaces between the identifier and the name fields. It identifies the JCL statement by name so that other JCL statements or the operating system can refer to it.

A name is optional for some EXEC statements, but is required on JOB and almost all DD statements.

38

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The *name* field provides a label for a particular statement so that the z/OS system, and other statements in the JCL, can refer to that particular statement.

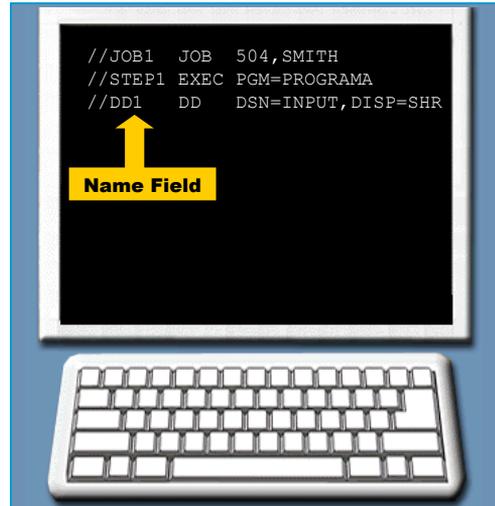
Choose meaningful names because they may help to locate problems later.

Introduction to JCL – JCL statement syntax.

Rules for naming JCL statements.

The specifications for JCL statement names are as follows:

- The name must begin in position 3.
- The name can be one to eight characters in length.
- The first character in the name must be alphabetic character (A to Z) or national character (#, @, \$).
- The first character cannot be numeric (0 through 9).



39

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

The following special characters can not be included:

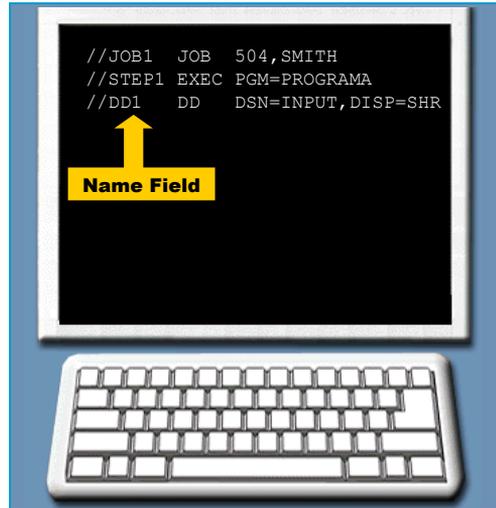
& (ampersand); ! (exclamation mark); * (asterisk); , (comma); = (equal sign); - (hyphen); .(period); + (plus sign); / (slash); ((left parenthesis);) (right parenthesis).

Introduction to JCL – JCL statement syntax.

Rules for naming JCL statements.

The specifications for JCL statement names are as follows:

- The remaining characters can be alpha numeric (A to Z and 0 through 9) or national characters.
- No special characters or spaces can be included in a name.



Introduction to JCL – JCL statement syntax.

Are we on track?

Match each JCL field types with their main function:

- | | |
|---------------------|--|
| 1. Parameter | A. Contains documentation information from the JCL coder. |
| 2. Comments | B. Specifies the type of statement. |
| 3. Operation | C. Contains specific statement values. |
| 4. Name | D. Identifies a JCL statement. |

The correct answer is 1-C, 2-A, 3-B and 4-D.

Introduction to JCL – JCL statement syntax.

Are we on track?

Which of the following is a valid JOB name?

A. //TEST*1

B. // JOB13

C. //\$ABCD

D. //STEPSEVEN

42

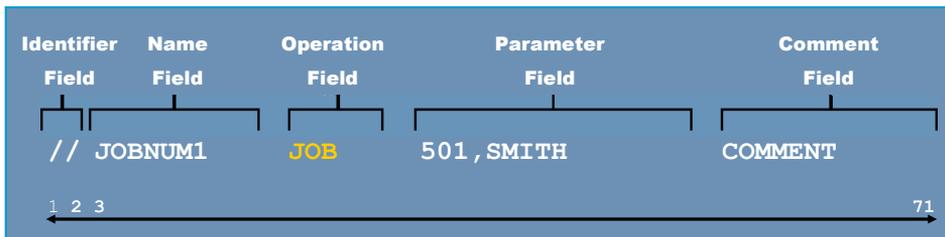
Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is C.

- A. Is invalid as it contains an asterisk in the name field.
- B. Is invalid as it contains a blank between identifier and the name field.
- D. Is invalid as the length of the name field is 9 characters which is greater than 8, the maximum allowed.

Introduction to JCL – JCL statement syntax.

The Operation field.



The operation field defines the JCL statement type - JOB, EXEC, or DD.

For every JCL statement, the operation field must follow the name field and it must be preceded by at least one blank space.

43

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

JOB, EXEC, and DD are main types of JCL statements, so you commonly see these values in the operation field.

In this diagram, the JCL statement type, JOB, appears in the operation field.

Introduction to JCL – JCL statement syntax.

Operation fields: an example.

In the JCL statements that you have seen so far, the operation fields contain the operators JOB, EXEC, and DD.



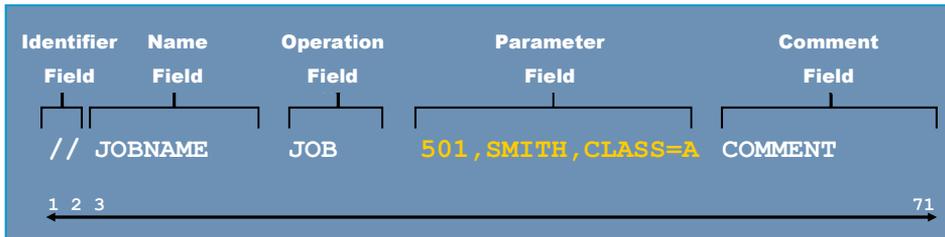
44

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

The first statement is a JOB statement, which marks the beginning of a job. The second statement is an EXEC statement, which marks the beginning of a job step and specifies the program to be executed (PROGRAMA). The third statement is a DD statement, which identifies a data set that PROGRAMA uses during processing.

Introduction to JCL – JCL statement syntax.

The Parameter field.



The parameter field contains one or more parameters separated by commas. Each JOB, EXEC and DD statement has its own set of parameter information.

The parameter field of each JCL statement must:

- Follow the operation field.
- Include one or more blanks between it and the operation field.
- Use commas to separate parameters.
- Have no blanks between parameters.

45

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

The *parameter* field, also referred to as the operand field, contains parameters that further describe the operation or resource defined in the JCL statement.

The parameter field may contain *positional* parameters or *keyword* parameters.

The CLASS parameter, shown in this diagram, illustrates the general format of parameters:

A specific combination of characters followed by an equals sign and an appropriate value. In some cases, however, the value you can specify for a positional parameter or keyword parameter consists of more than one item, called a *subparameter list*.

Introduction to JCL – JCL statement syntax.

Parameter fields: an example.

The JOB contains:

- Accounting information (504).
- The JCL programmer's name (SMITH).
- MSGLEVEL (message level) parameter.

The EXEC contains:

- The first one specifies the program name (PROGRAM1).
- The second one specifies the maximum CPU time for the program (4 minutes).

The DD contains:

- The first one specifies the data set name INPUT.
- The second one specifies a status SHR (shared).



General rules for coding positional and keyword parameters:

Positional parameters must appear in the order in which the z/OS system expects to find them. In this diagram, both 504 and SMITH are positional parameters, indicating the accounting number for this job, and the programmer's name, respectively.

Keyword parameters do not have to be coded in any particular order, but they must follow any positional parameters in the JCL statement. In this diagram, the MSGLEVEL keyword parameter follows both of the positional parameters on the JOB statement.

Introduction to JCL – JCL statement syntax.

Are we on track?

Place the following fields in the order in which they appear in a JCL statement (from first to last).

A. Name

B. Comment

C. Operation

D. Parameter

E. Identifier

47

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

The correct order is E., A., C., D., and B.

Introduction to JCL – JCL statement syntax.

Are we on track?

Match the following parameters with their descriptions:

- | | |
|----------------------|---|
| 1. SMITH | A. Maximum run time for the program. |
| 2. TIME=4 | B. Data set name. |
| 3. DSN=TRANS1 | C. Programmer's name. |

48

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct answer is 1-C, 2-A, and 3-B.

Introduction to JCL – JCL statement syntax.

Continuation statement.

You can continue a JCL statement on one or more separate lines if:

- **The JCL statement requires more than 71 positions to specify all the necessary parameters.**
- **You want to make it easier to read all the parameters. For example: You could code one parameter per line.**

Introduction to JCL – JCL statement syntax.

Procedure for continuing a JCL statement.

The procedure for coding a JCL statement on more than one line is as follows:

- 1. Interrupt the coding of a field after a complete parameter or subparameter, including the comma that follows it, before reaching position 72.**
- 2. Code // in positions 1 and 2 of the following statement.**
- 3. Continue the parameters of the interrupted field, beginning in any position from 4 to 16.**

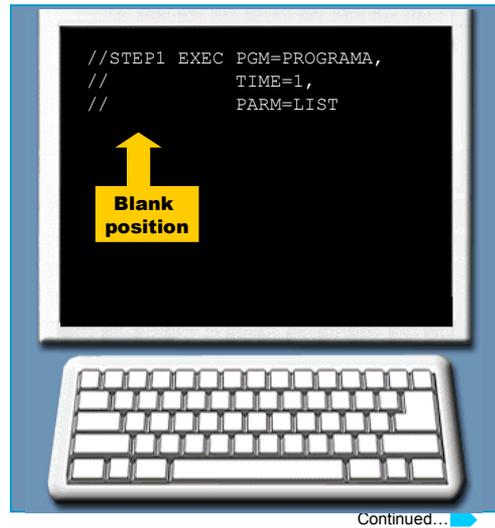


Introduction to JCL – JCL statement syntax.

Procedure for continuing a JCL statement.

Following is an example of a single JCL statement that continues on 2 additional lines. All 3 lines comprise one JCL statement:

1. In the parameter field of the JCL statement, end the line before reaching position 72, but after coding a complete parameter and the comma that follows it.

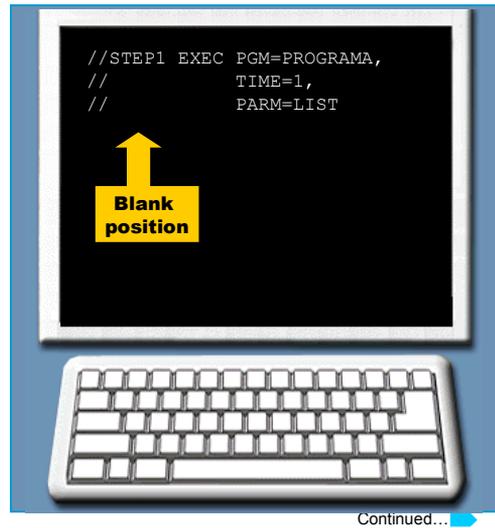


Introduction to JCL – JCL statement syntax.

Procedure for continuing a JCL statement.

Following is an example of a single JCL statement that continues on 2 additional lines. All 3 lines comprise one JCL statement:

2. On the next line, code two forward slashes (//) in positions 1 and 2, and a blank in position 3.
3. Beginning in position from 4 to 16, resume coding the JCL statement with the next complete parameter.



52

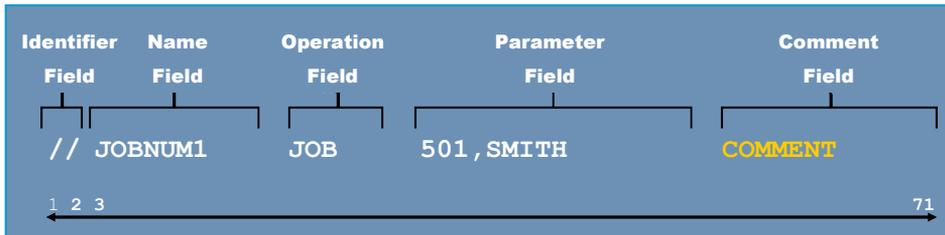
Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

To continue a JCL statement on one or more separate lines in this example:

- The first line ends after the complete specification of the PGM keyword, its value, and the comma to indicate that additional keywords follow.
- The second and the third lines each have a blank in position 3, after the identifier field (//). The blank in position 3 indicates a continuation of the EXEC statement.
- The continued EXEC statement parameters (on the second and third lines) begin in the same column, for improved readability.

Introduction to JCL – JCL statement syntax.

The Comment field.



A comment field contains extra documentation which you may choose to include. It contains documentation relating to the code or other messages from the programmer.

A comment field must follow the parameter field and must be preceded by one or more blanks.

53

Copyright © 2006 CA. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies.

You may place comments following the parameter field, or on a separate comment statement.

It is good programming practice to put comments in your JCL statements whenever possible. Comments help other programmers who might have to read and maintain your JCL code.

Introduction to JCL – JCL statement syntax.

Comment field: an example.

The following example shows a JCL statement that contains a comment 'INVENTORY JOB'. A blank separates the parameter field from the words INVENTORY JOB.



Continued... ➔

Introduction to JCL – JCL statement syntax.

Are we on track?

Place the following JCL statements in the correct order.

A. //STEP1 EXEC PGM=PROG1

B. //PAY JOB 03948,FRASER=(1,0)

C. //XYZ DD DSN

55

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

The correct order is B., A., and C.

Introduction to JCL – JCL statement syntax.

Rules for writing JCL statements.

You should remember the following rules while writing a JCL statement:

- **The identifying characters (//) must be in positions 1 and 2.**
- **The name field must begin in position 3.**
- **Each subsequent field must be preceded by one or more spaces.**
- **A continuation line begins with //.**
- **Continuing parameters must start in positions 4 through 16.**
- **Do not embed blanks within any field. For example, do not code a blank to separate parameters in the parameter field.**
- **Do not code past position 71.**

56

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.



Apart from these rules, JCL coding is free form. Fields (except for the name field) and parameters do not have to begin or end in any particular position.

Introduction to JCL – JCL statement syntax.

Glossary.

Alphabetic Characters
Capital A to capital Z.

Comment Field
The comment field contains documentation relating to the code or other messages from the programmer.

Identifier Field
The identifier field indicates to the system that a JCL statement rather than data.

Name Field
The name field identifies a particular statement so that other statements and the system can refer to it.

National Characters
The national characters are the at sign [@], the dollar sign [\$], or the pound sign [#].

57

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Glossary.

Numeric Characters

Numbers 0 through 9.

Operation Field

The operation field specifies the type of statement, or, for the command statement, the command.

Parameter Field

The parameter field contains statement parameters separated by commas.

Special Characters

These include ampersand [&]; exclamation mark [!]; asterisk [*]; comma [,]; equal sign [=]; hyphen [-]; period [.]; plus sign [+]; slash [/]; left parenthesis [(]; right parenthesis [)]

Introduction to JCL – JCL statement syntax.

Unit summary.

Now that you have completed this unit, you should be able to:

- **Identify the three major JCL statements and the purpose of each.**
- **Arrange JCL statements for a specific sequence of execution.**
- **Select three major considerations for grouping work into jobs.**
- **Arrange JCL statement field into the correct format.**
- **Identify properly coded JCL statements.**
- **Depending on the type of JCL statement, the parameter field may contain positional parameters, keyword parameters, and a subparameters.**

59

Copyright © 2006 CA. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Each JCL statement is logically divided into five fields: Identifier, name, operation, parameter, and comment. The following general rules apply:

- The identifying characters (//) must be in positions 1 and 2.
- The name field must begin in position 3.
- Each subsequent field must be preceded by one or more spaces.
- Do not code past position 71. If you need more than 71 positions to complete a statement, continue the statement on additional lines.
- To identify a continuation line, begin with two forward slashes and at least one blank space (//) in positions 1 through 3. Continuing parameters must start in positions 4 through 16.