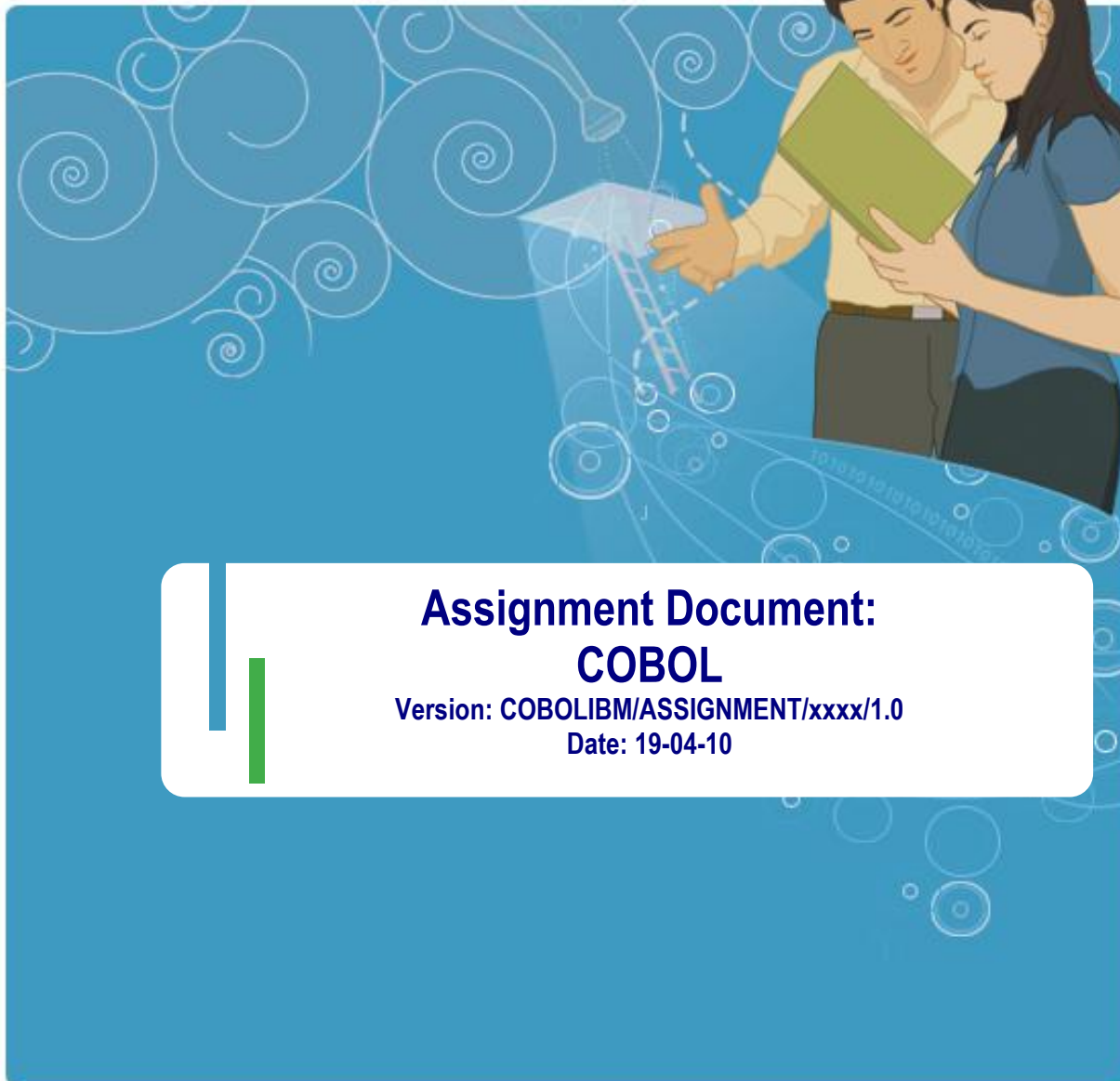




**Cognizant**

Passion for making a difference



## Assignment Document: **COBOL**

Version: COBOLIBM/ASSIGNMENT/xxxx/1.0

Date: 19-04-10

Cognizant  
500 Glen Pointe Center West  
Teaneck, NJ 07666  
Ph:  
[www.cognizant.com](http://www.cognizant.com)

201-801-0233

## Contents

<b>SESSION 06 TO 13 : PROGRAM CONTROL-1.....</b>	<b>3</b>
HANDS-ON EXERCISE 1.1: STRING HANDLING .....	3
Objective: .....	3
Sample code: .....	3
<b>SESSION 14 TO 18 : PROGRAM CONTROL-2.....</b>	<b>3</b>
HANDS-ON EXERCISE 2.1: CONDITION NAME CONDITIONS .....	4
Objective: .....	4
Sample code: .....	4
HANDS-ON EXERCISE 2.2: PERFORM STATEMENT.....	4
Objective: .....	4
Sample code: .....	5
<b>SESSION 19 TO 30 : FILE HANDLING .....</b>	<b>6</b>
HANDS-ON EXERCISE 3.1: KSDS WITH AIX .....	6
Objective: .....	6
Sample code: .....	6
HANDS-ON EXERCISE 3.2: RRDS IN COBOL.....	7
Objective: .....	7
Sample code: .....	7
<b>SESSION 31 TO 32 : TABLE HANDLING .....</b>	<b>7</b>
HANDS-ON EXERCISE 4.1: BINARY SEARCH IN COBOL TABLES.....	8
Objective: .....	8
Sample code: .....	8
<b>SESSION 33 TO 37 : INTERPROGRAM COMMUNICATION .....</b>	<b>8</b>
HANDS-ON EXERCISE 5.1: DYNAMIC CALLS.....	9
Objective: .....	9
Sample code: .....	9

## Session 06 to 13: Program Control-1

### Hands-On Exercise 1.1: String Handling

Estimated Completion Time: 10 Minutes

(10 Marks)

#### Objective:

Learn COBOL String Handling functions

Write a COBOL program that Performs the below listed string operations

- Displays the number of occurrences of character 'B' in string 'ABBCC BF'
- Replaces first occurrence of 'ABC' with 'XYZ' in string 'ABCABCDE'
- Given string (ABBCC BF) delimited by size with "test"
- Given strings (STRING1:TESTC BF , STRING2:XYZABCDE ) are concatenated with space as delimiter
- Given string (TESTC BF ) is split with space as delimiter

#### Sample code:



STRNGPGM.TXT



EXECJCL.TXT

Note: To view the Embedded document click on the Attachment Icon on the left panel

#### Evaluation Rubrics

Parameters	Weightage
Completeness	3
Accuracy	3
Clarity of understanding	2
Presentation	2
Total	10

## Session 14 to 18: Program Control-2

## Hands-On Exercise 2.1: Condition Name Conditions

Estimated Completion Time: 10 Minutes

(10 Marks)

### Objective:

Learn usage of COBOL Condition Name Conditions

Write a COBOL program to calculate bonus for Employees based on their marital status and employment status. 'Conditional variables (88 level variables)' and 'Evaluate' are used to verify the marital status and Employment status.

Marital Status	Employment status	Bonus
Married	Permanent	40000
Married	Temporary	20000
Single	Permanent	50000
Single	Temporary	25000

### Sample code:



Note: To view the Embedded document click on the Attachment Icon on the left panel

### Evaluation Rubrics

Parameters	Weightage
Completeness	3
Accuracy	3
Clarity of understanding	2
Presentation	2
Total	10

## Hands-On Exercise 2.2: PERFORM Statement

Estimated Completion Time: 10 Minutes

(10 Marks)

### Objective:

Learn usage of COBOL PERFORM statement

Write a COBOL program that takes customer details (Customer no., Credit-code, Qty, and Price) as input and calculates the Net amount after applying the discounts. Use 'PERFORM' statement to accept 5 records as input. If the Customer number input is not equal to 'C' then an error message 'customer number error' needs to be displayed. When the credit code is other than A, B, C, D and E, then error message 'credit code error' needs to be displayed.

Display the output in the given format:

CUST-NO-OUT	X (3).
FILLER	X (10).
CREDIT-CODE-OUT	X.
FILLER	X (2).
QTY-OUT	9(4).
FILLER	X (2).
RATE-OUT	9(3).
FILLER	X (3).
PRICE-OUT	9(6).
FILLER	X (2).
ACT-DIS-OUT	9(6).
FILLER	X (2).
NET-AMT-OUT	9(7).

Discount to be applied based on Credit code:

Credit code	Discount %
A	50
B	40
C	30
D	20
E	00

Calculations Involved:

$\text{PRICE-OUT} = \text{RATE-OUT} * \text{QTY-OUT}$   
 $\text{ACT-DIS-OUT} = \text{PRICE-OUT} * \text{DIS-OUT} / 100$   
 $\text{NET-AMT-OUT} = \text{PRICE-OUT} - \text{ACT-DIS-OUT}$

**Sample code:**



**Note:** To view the Embedded document click on the Attachment Icon on the left panel

### Evaluation Rubrics

Parameters	Weightage
Completeness	3
Accuracy	3
Clarity of understanding	2
Presentation	2
Total	10

## Session 19 to 30: File handling

### Hands-On Exercise 3.1: KSDS with AIX

Estimated Completion Time: 10 Minutes

(10 Marks)

#### Objective:

Learn to use KSDS file with AIX in COBOL Programs

Write a COBOL program that uses a KSDS file as Input. The structure of the KSDS file is as given below:

EMP-NUM(1-4)	DEPT-NO(6-11)	EMP-NAME(13-20)
1001	DEPT01	THUSHARA
1002	DEPT02	RAM
1003	DEPT03	SUPRI
1004	DEPT04	SHYAM
1005	DEPT12	ANNIE

Define an Alternate index for this KSDS file with DEPT-NO as the Alternate Key. In the COBOL program accept DEPT-NO as input and do a Skip sequential read based on the Department Number.

#### Sample code:



P8KSAIRD.TXT



J8KSAIRD.TXT

Note: To view the Embedded document click on the Attachment Icon on the left panel

### Evaluation Rubrics

Parameters	Weightage
Completeness	3
Accuracy	3

Clarity of understanding	2
Presentation	2
Total	10

### Hands-On Exercise 3.2: RRDS in COBOL

Estimated Completion Time: 10 Minutes

(10 Marks)

#### Objective:

Learn to use VSAM in COBOL Programs  
Write a COBOL program that uses a RRDS file as Input. The structure of the RRDS file is as given below:

EMP-NUM(1-4)	DEPT-NO(6-11)	EMP-NAME(13-20)
1001	DEPT01	THUSHARA
1002	DEPT02	RAM
1003	DEPT03	SUPRI
1004	DEPT04	SHYAM
1005	DEPT12	ANNIE

Accept the record number and Delete the record randomly based on the input.

#### Sample code:



P5RRRRAN.TXT



J5RRRRAN.TXT

Note: To view the Embedded document click on the Attachment Icon on the left panel

#### Evaluation Rubrics

Parameters	Weightage
Completeness	3
Accuracy	3
Clarity of understanding	2
Presentation	2
Total	10

## Session 31 to 32: Table handling

**Hands-On Exercise 4.1: Binary search in COBOL Tables****Estimated Completion Time: 10 Minutes****(10 Marks)****Objective:**

Learn to use Loop control structure and Binary search in COBOL Programs

Write a COBOL program that takes Employee details (Employee-Id,Name,Department) as input and writes it into a one dimensional table. Do a binary search on the Employee details table to verify if there is an employee named 'Ramya' and also if a department called Health exists.

The sample data is:

EMPLOYEE-ID	NAME	DEPARTMENT
190588	THUSHARA	DEPT1
200120	RAM	DEPT2
200121	SHYAM	DEPT3
201121	RAMYA	DEPT3
201122	SOWMYA	DEPT1

**Sample code:**

Note: To view the Embedded document click on the Attachment Icon on the left panel

**Evaluation Rubrics**

Parameters	Weightage
Completeness	3
Accuracy	3
Clarity of understanding	2
Presentation	2
Total	10

**Session 33 to 37: Interprogram Communication**



### Hands-On Exercise 5.1: Dynamic calls

Estimated Completion Time: 10 Minutes

(10 Marks)

#### Objective:

Learn Dynamic calls to subprograms in COBOL

Write a COBOL program 'DYNMAIN' that takes 2 numbers (NUM1, NUM2) and a subprogram name (DYN SUB) as input and calls the subprogram dynamically to calculate the sum (NUM3). Use Call by reference for the NUM1 and NUM2 and call by content for NUM3. Display NUM1, NUM2 and NUM3 in 'DYNMAIN'

#### Sample code:



DYNMAIN.TXT



DYNSUB.TXT



COBCOMP.TXT



JDYNMAIN.TXT

Note: To view the Embedded document click on the Attachment Icon on the left panel

#### Evaluation Rubrics

Parameters	Weightage
Completeness	3
Accuracy	3
Clarity of understanding	2
Presentation	2
Total	10