Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ABAP/4
Module2 Lab Book

Table of Contents

Getting Started

1.1 Overview

This lab book is a guided tour for learning SAP ABAP. It comprises of assignments to be done. Refer the demos and work out the assignments given by referring the case studies which will expose you to work with Java applications.

1.2 Setup Checklist for SAP ABAP

Here is what is expected on your machine in order to work with lab assignment.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 7 or higher.
- Memory: (1GB or more recommended)

Please ensure that the following is done:

- SAP GUI is installed
- Connection to the SAP Server is present

Lab 1-1 Introduction to ABAP Progarmming

Goals	• Create simple programs in ABAP editor
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1.1. Write a program to display the Empid, EmpName, Emp_addr in chain statement using write.

Hint: All the variable should be in declared using DATA statement.

1.2 . Write a program to display the value of System Date, System Time, Current User ID,Current report name by using single write statement.

1.3. Take the values of two numbers from user. If contents of both variables are negative then convert them into positive values then divide them by 2. If it is not divisible by two then truncate it and then swap them by using move statement.

1.4. Write a program to display the first 10 customers from customer master KNA1 table and consider any 10 fields to display the output.

1.5. Write a program to display the vendor master data from LFA1 table, vendors should be filterd by country code and order by city by accepting the selection screen range and consider any 10 fields to display the output.

Lab 2-1 Data Dictionary

Goals	• Create Tables, Views and Search Helps
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1. Create the tables with the following structure. Name of the table as z_empcode_emp.

z_nnnn_emp

Field Name	Data Element	Domain	Domain Data Type and lenght
EMPNO (PK)	Z_XX_DEEMPNO	Z_XX_DEMPNO	NUMC 4
ENAME	Z_XX_DEENAME	Z_XX_DENAME	CHAR 10
JOB	Z_XX_DEJOB	Z_XX_DJOB	CHAR 9
MGR	Z_XX_DEMGR	Z_XX_DMGR	NUMC 4
HIREDATE	Z_XX_DEHIREDATE	Z_XX_DHIREDATE	DATS
SAL	Z_XX_DESAL	Z_XX_DSAL	DECIMAL 7 2
COMM	Z_XX_DECOMM	Z_XX_DCOMM	DECIMAL 7 2
DEPTNO (FK)	Z_XX_DEDEPTNO	Z_XX_DDEPTNO	NUMC 2

z_nnnn_dept

Field Name	Data Element	Domain	Domain Data Type and length
DEPTNO(PK)	Z_XX_DEDEPTNO	Z_XX_DEDEPTNO	NUMC 2
DNAME	Z_XX_DEDNAME	Z_XX_DEDNAME	CHAR 14
LOC	Z_XX_DELOC	Z_XX_DELOC	CHAR 13

Note: Data Element and Domains for deptno field is the same in both tables.

z_nnnn_salgrade

Field Name	Data Element	Domain	Domain Data Type and length
GRADE	Z_XX_DEGRADE	Z_XX_DGRADE	NUMC 2
LOSAL	Z_XX_DELOSAL	Z_XX_DLOSAL	DECIMAL 7 2
HISAL	Z_XX_DEHISAL	Z_XX_DHISAL	DECIMAL 7 2

Where:

1) NNNN is your empcode
2) XX: are the names of your initials.

This is to ensure uniqueness of your table name, domain and data element name.

2. Create an ABAP program 'Z_XX_ASSG_4A' where XX denotes Group Id of your assignment group.

1. ABAP Inputs:

The ABAP program will have parameters for employee number , employee name and salary with the definitions as per the database table fields.

2. Screen Validations

Mandatory input for all parameters. Employee number can contain only numbers. Employee name should not be blank and Salary cannot be zero.

3. Main Processing Logic :

Read the employee table created above using the Employee number, which has been input.

If matching entry exists, then check the input name and salary. If at least one of them is different than db record, then record should be modified with changed values.

If matching entry does not exist, new entry should be created in table with input values.

4. Output Format:

If record has been modified, display 'Record Updated'.

If record has been added, display 'Record added'.

5. Test Conditions

Report should be run with values for which entries exists in the db and also new values for which entries do not exist, so that both addition and modification of records can be tested..

Error checking: If employee code contains characters other then numbers, then suitable error message should be displayed. (Define Text elements for message text).

Hints

For this ABAP make use of following:

SELECT, INSERT, UPDATE.

Create a copy of the ABAP 'Z_XX_PASSG_4B' and as a variation use only MODIFY instead of INSERT/UPDATE.

3. Create tables and Data base views.

Step # 1: Create two ztables (Eg: ZEKKO_TAB, ZEKPO_TAB) by using predefined table fields of EKKO AND EKPO select at least 5 fields from each table and load some data into that two ztables.

Note: To enter the data/entries to ZTables Ref: EKKO and EKPO tables content/entries.

Step # 2: Maintain foreign key between the two ztables.

Step # 3: Create the data base view for the above two ztables.

Step # 4: Write an abap code and get the data from the database view (from the step # 3) and display the output.

4 Create help view and search help.

Step # 1: Create two ztables (Eg: ZVBAK_TAB, ZVBAP_TAB) by using predefined table fields of VBAK AND VBAP select at least 5 fields from each table and load some data into that two ztables.

Step # 2: Maintain foreign key between the two ztables.

Step # 3: Create the help view for the above two ztables.

Step # 4: Create the elementary search help and provide the help view name in the selection method of search help.

Step # 5: Attach the search help to the primary keys of the both the ztables.

Lab 3-1 String Operations

Goals	• Create an ABAP Program for String Operations
Time	90 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting SAP server

1 Create an ABAP 'Z_XX_ASSG_1' where XX denotes Group Id of your assignment group.

ABAP will have following parameters.

P_STRING1 - input string. The input should be taken in LOWERCASE.
FLAGS should be displayed as radiobutton. Depending upon the value of the selected FLAG, corresponding function will be performed on the input string.

T_FLAG – to convert the string to UPPERCASE
L_FLAG – to return the length of the string
S_FLAG – to remove leading zeroes in the string
O_FLAG – to return subset of string starting from offset P_OFFSET and having length P_LENGTH.

P_OFFSET and P_LENGTH are parameters to take in the offset and length of the substring.

1. Screen Validations

Offset and Length should be numeric.

In case O_Flag not set, both offset and length must be 0.
O_Flag set, at least length must be non-zero positive value.
O_Flag set, then offset + length should be <= length of string.
S_Flag set, then string must contain leading zeroes.

2. Main Processing Logic:

Depending upon the FLAG, corresponding operation should be performed on the string.

3. Output Format:

If there is an error during validations, suitable error message should be displayed.
Or If the validations are successful, then the resultant string should be displayed after the operation is performed.

4. Test Conditions

Testing should be performed for all values of flag.

Error checking: Check out all scenarios where error will occur and check whether appropriate error message is displayed or not (Define text elements for message text).

Hints:

For this ABAP make use of following:
RADIOBUTTON GROUP, TRANSLATE, STRLEN, SHIFT.

2. Create an ABAP 'Z_XX_ASSG_2' where XX denotes Group Id of your assignment group.

ABAP will have following parameters.

Define P_STRING – input string default value .
'000000000075001234, Material No: 000000000014634566, 53'.

Screen-validations:

Check if atleast 1 comma exists in the input string.

Main Processing Logic:

Split the input string at comma into 3 different variables.

Search 2nd of these variables for string 'Material No:'.

Move the Material no. into another variable.

Output Format:

Write the extracted material no. to the screen.

Hints:

Use SPLIT, SEARCH, STRLEN, commands

3. Create an ABAP 'Z_XX_ASSG_3' program where XX denotes Group Id of your assignment group.

ABAP will have following parameters.

Provide String_1 and String_2 value as a default in the program and user can change the values at runtime with own input values.

Screen-validations:

Check atleast one space should be exists in the input value of string_1 and it should be an obligatory and check String_2 value also an obligatory.

Main Processing Logic:

Accept two input strings (String_1 and String_2) from the user at rutime by using parameters and display these two string values at the list output as a heading.

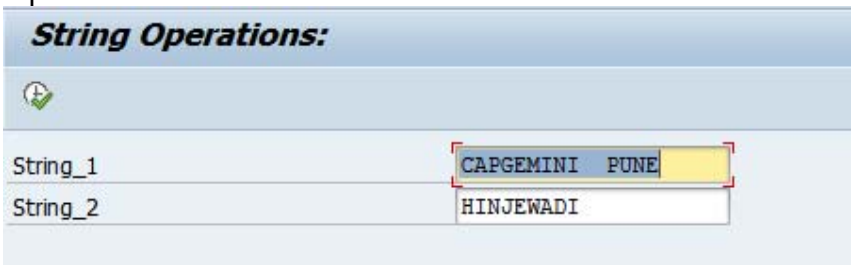
Hints:

 # 1. Use CONCATENATE, TRANSLATE, CONDENSE, NO-GAPS, STRING LENGTH, SHIFT AND SHIFT BY PLACES commands to display the main logic.

 # 2. Use split command by using Internal tables .

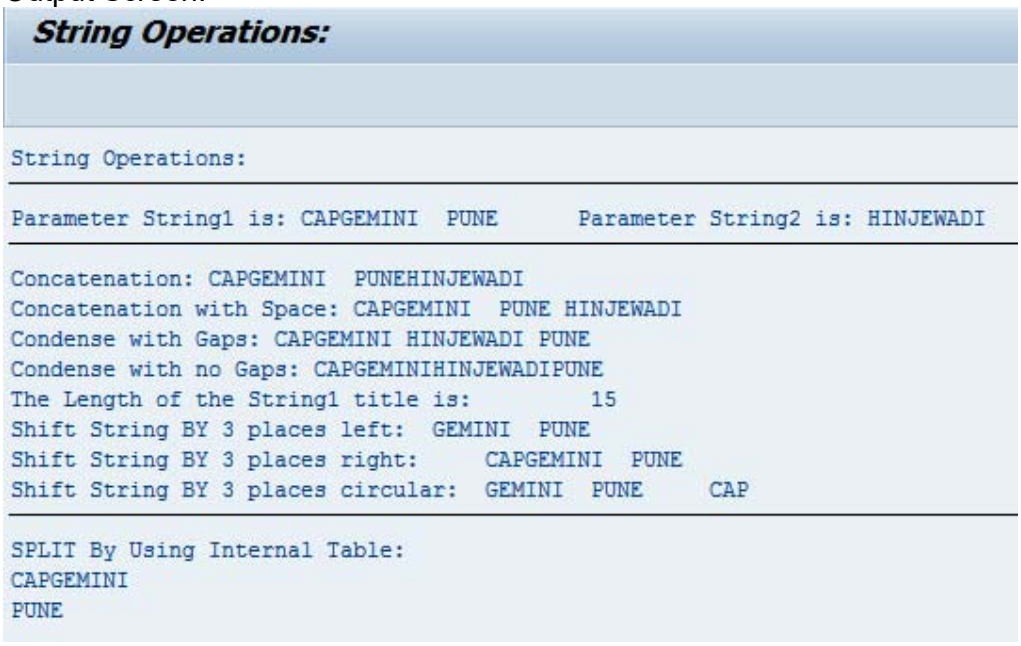
Input Format: Format of the input screen like below with proper text and title by using the text elements and list headings.

Input Screen:



Output Format: the output format should be like as shown below.

Output Screen:



Lab 4-1 Internal tables

Goals	• Insert data into an Internal table and process it
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

1) Create an ABAP 'Z_XX_ASSG_2' where XX denotes Group Id of your assignment group.

ABAP will have following parameters.

P_ITEM – input string of 20 characters (select options)
P_QTY – input number (select options)
P_RATE – input number with 2 decimal points (select options)

1. Screen Validations

- All the inputs are mandatory.
- Qty and Rate cannot be zero.

2. Main Processing Logic:

1. Accept 5 items each with qty and rate.
2. Store in internal table.
3. Calculate their price.
4. Sort on price in ascending order.
5. Display output as per section 6.

6. Output Format:

ITEM	QTY	RATE	PRICE
TOTAL			

7. Test Conditions

Testing should be performed for all options.
Error checking: Check out all scenarios where error will occur and check whether appropriate error message is displayed or not (Define Text elements for message text).

Hints

For this ABAP make use of following,
APPEND, SUM, MODIFY

Lab 5-1 Advanced Internal tables

Goals	• Use control break logic to display data from Internal table and other commands. To work with an Internal table commands INSERT, INSERT Multiple, UPDATE, MODIFY and DELETE.
Time	60 Minutes
Lab Setup	• Connectivity to SAP server • Login details for connecting to SAP server

List Processing with control-break statements

1) Create an ABAP 'Z_XX_ASSG_11' where XX denotes Group Id of your assignment group.

1. ABAP Inputs :

ABAP will have following parameters.

1) Select option S_DEPTNO which is non-mandatory
2) parameter P_SALARY Default this with some salary amount.

2. Main Processing Logic:

The report will fetch data from employee table which you created in previous assignment.

If select-option S_DEPTNO has some values then data for only those dept.s should be fetched. Otherwise all records from table should be fetched. If P_SALARY is entered, records should be filtered on that basis. If both S_DEPTNO and P_SALARY are entered, records should be filtered based on both selection-criteria.

You need to output data for each dept. No.

The report output should be like.

```
DEPT No : Dept_1
-------------------------------------------
Employee No.   Salary
-------------------------------------------
Emp1           20000
EMP2            40000
-------------------------------------------
```


Total salary for Dept :

DEPT No. : Dept_2

```
--------------------------------------------
Emp3            50000
Emp4            30000
--------------------------------------------
Total salary for Dept :
--------------------------------------------
```

3. Test Conditions

Testing should be carried out for different values of S_DEPTNO .
Test if proper records are selected when records should be fetched only where salary > 30000. (You should enter p_salary > 30000 at selection screen.)

Hints

Use control-break statements.

2) Write an ABAP Program to create an internal table with the Following Specifications.

Field Name	Data Type	Length
Mat No	Char	18
Mat Description	Char	40
Mat Type	Char	4
Quantity	Quan	10
Price	Quan	10
Base UOM	Unit	3

Perform the below an operations on an Internal table and display the output in a list format:

A. Insert records into an internal table.
B. Display all the records from the table.
C. Display the records whose price is > 20,000.
D. Change the base unit of measure (UOM) from 'PC' to 'EA'.
E. Delete the records where the quantity is < 10.
F. Display the total amount.
G. Delete the all the records.

3) Write an ABAP program to calculate the Grade of the students, based on the input marks.

A. If the input Marks are > 90 - Grade A.
B. If the input Marks are between 75 and 90 – grade B.
C. If the input Marks are between 60 and 75 – grade C.
D. If the input Marks are < 60 grade D.

4) Write an ABAP program for INSERT Records, INSERT Multiple Records, Modify and Delete the records from the Internal Table.

4.1 Create a field string with the following fields and data types for the Insert records of an internal table.

Field Name	Data Type	Length
Name	C	30
Age	I	NA
Weight	P	Decimals 3
Land	C	20

Hints:

1. Assign some hardcoded values to the internal table fields as shown the below screen.
2. Insert 3 records into an internal table.
3. Use standard or sorted tables to identify the Insert order of the records.

Output Screen:

```
Internal Table [Insert] By Using Sorted Table
-------------------------------------------------
NAME                AGE             WEIGHT COUNTRY
-------------------------------------------------
DAVID               40              95.00  DE
JOHAN               35              80.00  IND
PETER               35              45.00  USA
```

4.2 Create a field string (with DATA) with the following fields and data types, to Insert the multiple records of an internal table.

Field Name	Data Type
COL_1	I
COL_2	I

Hints:

1. Insert multiple lines with **Do Times** statements by using the **Sy-Index.**
2. Take the index values into COL_1 = (sy-index) and COL_2 = (2 * sy-index).
3. Loop the internal table values into Field string and multiply the COL_1 (3 * sy-tabix) and COL_2 (5 * sy-tabix) with the sy-tabix.

Output Screen:

Internal Table [Insert Multiple Lines]

TABIX	COL_1	COL_2
1	3	5
2	1	2
3	9	15
4	2	4

4.3 Create a field string (with DATA) with the following fields and data types to modify the records of an Internal table.

Field Name	Data Type
COL_1	I
COL_2	I

Hints:

1. Insert multiple lines with **Do 3 Times** statements by using the **Sy-Index.**
2. Take the index values into COL_1 (**Sy-Index**) and COL_2 (sy-index ** 2.).
3. Loop the internal table values into Field string and multiply the COL_1 (sy-tabix * 10) and COL_2 ((sy-tabix * 10) ** 2).
4. Modify the itab if sy-tab index eq 2.

Output Screen:

Internal Table [Modify]

TABIX	COL_1	COL_2
1	1	1
2	20	400
3	3	9

4.4 Create a field string (with DATA) with the following fields and data types to delete the records of an internal table.

Field Name	Data Type
COL_1	I
COL_2	I

Hints:

1. Insert multiple lines with **Do 4 Times** statements by using the **Sy-Index.**
2. Take the index values into COL_1 = (Sy-Index) and COL_2 = (sy-index ** 2.) Insert from field string to itab.
3. Delete the internal table records, where the col_1 = 3rd row.

Output Screen:

```
Internal Table [Delete]
-----------------------

      COL_1        COL_2
-----------------------

         1            1
         2            4
         4           16
```

