

# INFORMATICS PRACTICES

*Time allowed : 3 hours*

*Maximum Marks : 70*

## NOTE :

- (i) All questions are compulsory.
- (ii) **Section A** consists of 30 marks.
- (iii) **Section B** and **Section C** are of 20 marks each.
- (iv) Answer the questions after carefully reading the text.

## QUESTION PAPER CODE 90/1

### SECTION A

1. Answer the following questions :
  - (a) Define Data Mining using suitable example. 2
  - (b) Name any four application areas where business computing can be used. 2
  - (c) Differentiate between ER Modelling and Object Modelling Techniques. 2
  - (d) What do you understand by a One-to-Many Relationship ? Explain with the help of an example. 2
  - (e) Expand the term SDLC. Name its essential components. 2
2. Answer the following questions :
  - (a) What is a syntax error in the context of a program ? Give an example. 2
  - (b) Differentiate between Do While loop and Do Until loop of Visual Basic giving a suitable example of each. 2
  - (c) What do you understand by the term Record Source of an ADO Data Control ? 2
  - (d) Define the term Library Functions in Visual Basic. Name the different categories of library functions available in Visual Basic. Give the usage and syntax of any two library functions. 4
3.
  - (a) Define SQL. Name the different SQL subcategories (give full form). 2
  - (b) Explain the IN operator of SQL, specifying its syntax and usage. 2
  - (c) Differentiate between Single Row Functions and Multiple Row Functions of SQL. Give examples for both ? 2

- (d) What is a Cursor in PL/SQL ? List any two commands that are associated with cursor control. 2
- (e) Write a PL/SQL procedure called NEXTMONTH that takes a date as parameter and adds 30 days to that date and displays it. 2

### SECTION B

4. Read the following case study and answer the questions that follow :

Mr. Vidyarthi works in Blossoms Public School as a programmer. He is required to develop a student record. The school offers two different streams, medical and non-medical, with different grading criteria. The school also offers incentive to the NCC cadets in the form of a 3% increment in percentage for all the NCC cadets. The grading criteria for the two streams is given below :

Stream	Percentage	Grade
Medical	$\geq 80$	A
	60-80	B
	$< 60$	C
Non-Medical	$\geq 75$	A
	50-75	B
	$< 50$	C

Object Type	Object Name	Description
Form	FrmStudRec	The Main Form Object
Text Box	txtFirstTerm	To enter first term marks
	txtSecondTerm	To enter second term marks
	txtPercentage	To display the percentage of the student
	txtGrade	To display the grade of the student
Check Box	chkCadet	To be Checked if student is an NCC Cadet
Option Button	optMedical	To provide Stream Information
	optNonmedical	
Command Button	cmdCalcPer	To calculate the percentage
	cmdCalcGrade	To calculate the grade
	cmdClear	To clear the entered values
	cmdExit	To close the application

- (a) Write the commands to disable the textboxes txtPercentage and txtGrade. 1
- (b) Write the code for cmdClear Command Button to clear all the textboxes and the checkbox. 1
- (c) Write the code for cmdCalcPer to calculate the percentage after finding the total marks of first term and second term (assuming that both marks are out of 100). Also ensure that NCC cadets get an increment of 3% in their percentages. 4
- (d) Write the code for cmdCalcGrade to calculate the grade depending on the stream selected according to the criteria given above. 4

5. Answer the following questions :

- (a) Find the errors from the following code segment and rewrite, the corrected code underlining the correction made: 2

```

Private Sub Command1_Click()
Dim p = 1 As Integer, i As Integer
For i = 1 UpTo 50
    p = p + 1
    If p = 5 Then
        p = 1
        Display "P is equal to 5"
    Else
        Display "P is not equal to 5"
    End If
Loop i
End Sub

```

- (b) Find the Output of the following code segment: 2

```
Dim work As Integer, I As Integer
work = 10
I = 0
Do While work < 20
    Print work * I
    I = I + 2
    If I = 8 then
        work = 20
    Else
        work = work - 3
    End If
Loop
```

- (c) Rewrite the following code segment using For...Next instead of Do...While: 2

```
Private Sub cmdFactorial_Click ()
    Dim factorial As Long
    Dim counter As Integer
    Dim strAnswer As String
    counter = Val (txtFactorial.Text)
    factorial = counter
    Do While counter > 1
        counter = counter - 1
        factorial = factorial * counter
    Loop
    Print "Factorial of " & txtFactorial.Text & " is " & factorial
End Sub
```

- (d) Write a Visual, Basic procedure which takes a string as argument and displays the following: 4

- The string in uppercase
- The length of the string
- The string with its first and last characters *in* uppercase and all the other characters in lowercase.

## SECTION C

6. Read the questions given below and answer accordingly :

- (a) Write the output produced by the following part of code in PL/SQL : 2

```
DECLARE
    X NUMBER;
    SUM NUMBER := 0;
BEGIN
```

```

FOR X IN 1..5 LOOP
    IF MOD (X, 2) = 1 THEN
        SUM := SUM+X;
    ELSE
        SUM:=SUM-X;
    END IF;
    DBMS_OUTPUT.PUT_LINE (TO_CHAR (I*10));
END LOOP;
END;
```

(b)

<u>ENAME</u>	<u>SAL</u>	<u>COMM</u>
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400
TURNER	1500	0

Look at the Emp table given above and give the output produced by the following PL/SQL code on execution :

2

DECLARE

```

V_EName      Emp.EName%TYPE;
V_Sal        Emp.Sal%Type:=1500;
V_Counter    NUMBER (2) :=1;
BEGIN
LOOP
    SELECT EName INTO V_EName FROM Emp WHERE Sal < V_Sal ;
    DBMS_OUTPUT.PUT_LINE (V_EName) ;
    V_Sal : = 1800;
    V_Counter = V_Counter + 1
    EXIT WHEN V_Counter >2;
END LOOP;
END;
```

(c) Explain the usage of the %TYPE attributes in variable declaration in a PL/SQL block with the help of an example.

2

(d) Write a PL/SQL Function CheckDiv that takes two numbers as arguments and returns the value 1 if the first argument passed to it is divisible by the second argument, else will return the value 0.

4

7. Answer the questions based on the table Flight given below :

**Table : Flight**

Column Name	Data Type	Size	Constraint	Description
Flight_No	NUMBER	4	PRIMARY KEY	Flight number
Origin	VARCHAR2	30	NOT NULL	Place of origin of flight
Destination	VARCHAR2	30	Not Null	Destination of the flight
Seats	NUMBER	3		Number of seats available
Flt_Date	DATE			Date of flight
Rate	NUMBER	7, 2		Rate of a ticket on the flight

- Write the SQL command to create the table **Flight** including the constraints. 2
- Write the SQL command to display the details of all the flights whose **Destination** is the same as the destination of Flight\_no **9001**. 2
- Write a PL/SQL code to increase the **Rate** of all the flights by 5%. 2
- Write PL/SQL code to create two Statement level triggers **B\_D\_Flight** and **A\_D\_Flight** before and after DELETE statement respectively on the table **Flight** which displays the message 'Ready for Deletion' and 'Records Deleted' respectively. 4

### QUESTION PAPER CODE 90

#### SECTION A

- Answer the following questions :
  - Define Data Warehousing and explain its usage. 2
  - What do you understand by the term SDLC ? Name its essential components. 2
  - Explain the term Front End Tool and Back End Tool in the context of software development process. 2
  - What is the Entity Relationship Model ? Explain the concept of ER Model using the case study of a School that has three entities : Student, Teacher and Subject. Assume that each student can study more than one subject while each teacher can teach only one subject. 4
- Answer the following questions :
  - What are conditional or decision control statements ? Name the two conditional statements used in Visual Basic. 2
  - Differentiate between public and private variables and discuss their scope of visibility. 2

- (c) Explain the terms ODBC and OLE-DB. 2
- (d) Differentiate between SDI Form and MDI Form in Visual Basic. In the similar context, explain the concept of parent and child forms. 4
3. Answer the following questions :
- (a) Differentiate between DML and DCL giving suitable example in each case. 2
- (b) What is NULL value ? What is the result of an arithmetic operation containing NULL value ? 2
- (c) What is the use of a Sub-Query ? Which query gets executed first, the parent query or the sub query ? 2
- (d) Which PL/SQL command is used to display an output on the screen ? Also name the command used to enable the server output option before using the above command. 2
- (e) Differentiate between Row-level and Statement-level triggers in PL/SQL. 2

## SECTION B

4. Read the following case study and answer the questions that follow :

Mr. Rouble of Passawasool Financiers frequently needs to calculate the interest and amount due from his clients. He asks his software programmer to design an interest calculator which will calculate the compound interest and amount due if a person takes a loan for 5, 10 or 15 years. The programmer opts for Visual Basic Language to develop this and creates the following form :

The screenshot shows a Windows-style application window titled "Interest Calculator". Inside the window, there is a form with the following elements:

- Principal:** A text input field.
- Rate:** A text input field.
- Time:** A group box containing three radio buttons:
  - ☒ 5 Years
  - ☐ 10 Years
  - ☐ 15 Years
- Buttons:** Three buttons labeled "Calculate", "Clear", and "Exit" are arranged horizontally.
- Output Fields:** At the bottom, there are two text input fields labeled "Interest" and "Amount".

The List for the above form is as follows :

Object Type	Object Name	Description
<b>Form</b>	frmInterestCalc	The Main Form Object
<b>Text Box</b>	txtPrincipal	To enter the principal
	txtRate	To enter the interest rate
	txtInterest	To display the compound interest
	txtAmount	To display the amount due
<b>Option Buttons</b>	opt5years	To select time as 5 years
	opt10years	To select time as 10 years
	opt15years	To select time as 15 years
<b>Command Button</b>	cmdCalculate	To calculate the interest and amount
	cmdClear	To clear all the entered data
	cmdExit	To end the application

- (a) Write the code to disable the textboxes txtInterest and txtAmount in the form load event of frmInterestCalc. 2
- (b) Write the code for cmdClear Command Button to clear all the textboxes and set default choice in the Option Buttons as 5 years. Also set the focus to txtPrincipal. 2
- (c) Write the code for the Change event of the textboxes txtPrincipal and txtRate to ensure that the user enters only numeric values in them. 2
- (d) Write the code for the Click event of the command button cmdCalculate to calculate the compound interest and amount depending on the principal, rate and time. The amount is calculated as  $P \times (1 + R/100)^T$  and Compound Interest as Amount – Principal. 4

5. Answer the following questions :

- (a) A student wants to write a procedure to find if a given year is a leap year or not. He had written the following code segment and he is unable to correct some of the errors in the code. Find the Errors and rewrite the corrected code underlining the correction made. 2

Note : A leap year is a year which is divisible by 4 or divisible by 400.

'Program code to check if a year is a leap year or not

Private Sub cmdLeapyear\_ONClicking ()

Y=Val (txtyear)

IF Y Mod 100 = 0 and Y Mod 400 = 0



```

        MsgBox "It is a Leap Year"
ELSEIF Y Mod 4 = 0
        MsgBox "It is a Leap Year"
ELSE
        MsgBox "It is not a Leap Year"
Sub End

```

- (b) Find the output of the following code segment :

2

```

Private Sub cmdClickMe_Click()
    Dim N1, N2, Counter
    Counter = 1
    N1 = 0
    N2 = 1
    Do while Counter <= 4
        Print N1
        Print N2
        N1 = N1 + N2
        N2 = N2 + N1
        Counter = Counter + 2
    Loop
End Sub

```

- (c) Rewrite the following code segment using If...Elseif...EndIf instead of Select...Case.

2

```

Select Case marks
Case Is >= 85
    comment.Caption = "Excellent"
Case Is >= 60
    comment.Caption = "Above Average"
Case Is >= 50
    comment.Caption = "Average"
Case Else
    comment.Caption = "Need to work harder"
End Select

```

- (d) Write a Visual Basic function that takes an integer argument N and returns the sum of all even numbers less than the given number N.

4

## SECTION C

6. Read the questions given below and answer accordingly :

(a) Write the output produced by the following part of code in PL/SQL :

2

```
DECLARE
    COUNT  NUMBER:=10;
    SUM  NUMBER  :=  0;
BEGIN
    LOOP
        SUM  =  SUM  +  COUNT;
        DBMS_OUTPUT.PUT_LINE  (TO_CHAR  (SUM)  )  ;
        COUNT  :=  COUNT  -  2;
        IF  COUNT  =  2  THEN
            EXIT;
        END  IF;
    END  LOOP;
END;
```

(b)	<u>EMPNO</u>	<u>ENAME</u>	<u>DEPTNO</u>	<u>SAL</u>
	7839	KING	10	5000
	7782	CLARK	10	2450
	7934	MILLER	10	1300
	7788	SCOTT	20	3000
	7902	FORD	20	3000
	7566	JONES	20	2975
	7876	ADAMS	20	1100
	7369	SMITH	20	800
	7698	BLAKE	30	2850
	7499	ALLEN	30	1600
	7844	TURNER	30	1500

Look at the Emp table given above and give the output produced by the following PL/SQL code on execution :

2

```
DECLARE
    Sum_Sal      Emp.Sal%TYPE  ;
    s_Sal        Emp.Sal%TYPE  :=2000  ;
    s_Dept       Emp.Deptno%Type  :=20  ;
```

```

BEGIN
    SELECT SUM (Sal) INTO Sum_Sal FROM Emp WHERE Deptno=s_Dept AND
    Sal>s_Sal ;
    DBMS_OUTPUT.PUT_LINE (TO_CHAR (Sum_Sal) ) ;
END;

```

2

(c) Differentiate between the IN and OUT parameters of a PL/SQL procedure.

2

(d) Write a PL/SQL Function Power that takes two numbers as arguments and returns the value of the first number raised to the power of the Second.

4

7. Answer the questions based on the table Employee given below :

**Table : Employee**

Column Name	Data Type	Size	Constraint	Description
Emp_ID	NUMBER	8	PRIMARY KEY	Employee's Identification number
First_Name	VARCHAR2	25	NOT NULL	First name of employee
Last_Name	VARCHAR2	25		Last name of employee
Date_Join	DATE			Date of joining
Basic_Sal	NUMBER	8, 2		Basic salary
Dept_ID	NUMBER	3		Department number

(a) Write the SQL command to create the table **Employee** including the constraints.

2

(b) Write the SQL command to display first name, date of joining and department id of employees from the Employee table who are hired between March 20, 1991, and December 31, 1991. Order the query in ascending order of date of joining.

2

(c) Write a PL/SQL command to create a view, which contains **Emp\_ID** and **Bonus** where Bonus is 12% of the **Basic\_Sal**.

2

(d) Write a PL/SQL block to delete all the rows of a particular department from the table **Employee** where the department id to be deleted is accepted from the user.

4

# Marking Scheme – Informatics Practices

Examiners are requested to consider all possible answers and program code/segments

Examiners are also requested to strictly follow the instructions (making scheme)

Questions having answers in the marking scheme are just indicative answers and bare minimum requirement, anything written extra should be ignored unless and until it is contradicting the actual answer.

## QUESTION PAPER CODE 90/1

### EXPECTED ANSWERS / VALUE POINTS

#### SECTION A

1. (a) An information extraction activity whose goal is to discover hidden facts contained in databases.

For example, data mining software can help retail companies find customers with common interests or predict the turnover for next year based on data of previous years.

*(1 mark for any correct definition)*

*(1 mark for any correct example)*

**OR**

*(Full 2 marks for correct example with explanation)*

- (b) Inventory control, Financial Accounting, Pay-Accounting System, Invoicing Management System, Personal Management System / HRD System, Fees Management system, Result Analysis System, Admission Management System, Income Tax Management System.

*(1/2 mark each for naming any four areas similar to those mentioned above)*

- (c) The entity-relationship modeling (or ER modeling) is a graphical representation of entities and their relationships to each other, used in organization of data within databases. An entity represents a discrete object and a relationship captures how two or more entities are related to one another.

Object Modeling technique is based on the concept of an “object” which is a data structure, which consists of data, and a set of routines, called methods/messages/functions, which operate on the data. It is used for figuring out what the objects of a system are, how they are related, and how they collaborate with each other.

*(2 marks for correct distinction)*

**OR**

*(1 mark for each correct definition i.e. for ER modeling and Object modeling)*

**OR**

*(½ mark if Only full form of ER as Entity Relationship is given)*

- (d) One-to-many relationship is where one instance of an entity (A) is associated with one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A.

For example, in a company with many employees working in one department, but each employee belongs to a single department. (Or any such example)

*(1 mark for any correct definition)*

*(1 mark for correct example/diagrammatic representation)*

**OR**

*(Full 2 marks for correct example/diagrammatic representation with explanation)*

- (e) **Systems Development Life Cycle (SDLC)**

Essential Components are :

- Evaluation / Analysis/ Preliminary Investigation/Feasibility Study
- Design
- Implementation / Coding
- Testing/Debugging
- Installation/ Parallel Run/ Final Execution
- Maintenance
- Review

*(1 mark for correct full form)*

*1 mark for any four correct components in any order)*

*(½ mark if any two components are correct)*

**OR**

*½ Mark for full form having three terms in the full form correctly identified)*

2. (a) A syntax error is an error of language resulting from code that does not conform to the syntax of the programming language. Syntax errors can be recognized at compilation time.

Any one example of a syntax error in Visual Basic Or PL/SQL.

*(1 mark for any correct definition)*

*(1 mark for correct example)*

**OR**

*(Full 2 marks for correct example with explanation)*

- (b) Do While loop repeats a block of statements until a specified condition is true.

Do Until repeats a block of statements until a specified condition is false.

**Any example** using one of the following syntax for do while loop :

Do While Condition	Do
Statement	Statement
Statement	Statement
...	...
Loop	Loop While Condition

**Any example** using one of the following syntax for do until loop:

Do Until Condition	Do
Statement	Statement
Statement	Statement
...	...
Loop	Loop Until Condition

*(1 mark each for any correct example/syntax of each loop)*

**OR**

*(½ mark each for any correct explanation of each loop without example)*

- (c) The table or logical set of records that will be accessible through bound control using ADODC in a form.

**OR**

The Record Source property of ADODC specifies the source of the records accessible through bound controls on the form. It can be set to a table Or an SQL statement, or stored procedure.

*(2 mark for any correct definition/explanation/self explanatory example)*

**OR**

*(1 mark for only defining ADODC)*

- (d) Visual Basic offers many functions that are pre-defined or built-in into the VB interpreter, which can be used directly in any program. These functions are highly reliable and can tremendously reduce the amount of coding required for a program.

Different categories are:

- Math functions
- String functions
- Date Functions
- Conversion Functions
- Type Checking functions

Example (usage) **OR** syntax of any two inbuilt functions to be specified.

*(1 mark for any correct definition)*

*(1/2 mark each for any two valid categories)*

*(1 mark each for correct example **OR** syntax of two built-in functions)*

3. (a) **SQL (Structured Query Language)** is a standard computer language for accessing and manipulating databases.

The different categories are:

DDL - Data Definition Language

DML - Data Manipulation Language

DCL - Data Control Language

TCL - Transaction Control Language

*(1/2 mark for correct definition **OR** fullform)*

*(1/2 mark each for full form of Any three subcategories)*

**OR**

*(Only 1/2 mark for Any three of the abbreviated subcategories)*

- (b) The IN operator implements comparison to a list of values, i.e. it tests whether a value matches any value in a list of values.

Syntax :

```
SELECT column_name FROM table_name WHERE column name IN  
(value1,value2,...);
```

Usage/Example :

```
SELECT name FROM travel WHERE city IN ('Rome', 'Paris') ;
```

*(1 mark for any correct definition)*

*(1 mark for correct syntax **OR** example)*

**OR**

*(Full 2 marks for any correct explanation through example/syntax)*

- (c) Single-row functions return a single result row for every row of a queried table or view processed. Example: Number functions, Character functions, Datetime functions, Conversion functions etc.

Multiple row functions work on a set of rows. Each function has one input argument and returns one result for each group of rows processed. Example: SUM, AVG, COUNT, MAX or MIN.

*(½ mark each for any correct definition)*

*½ mark each for any one example of both)*

**OR**

*(Full 2 mark for correct differentiation through example/syntax)*

- (d) Whenever a SQL statement is issued the Database server opens an area of memory in which the command is parsed and executed. This area is called a cursor. Cursors can be explicitly defined and manipulated allowing the processing of multiple rows.

DECLARE, OPEN, FETCH and CLOSE are the commands, which are associated with cursor control.

*(1 mark for any correct definition)*

*(½ mark each for any two cursor control commands)*

**OR**

*(Only ½ mark for any two SQL commands used in cursor)*

- (e) CREATE OR REPLACE Optional PROCEDURE NEXTMONTH (D1 DATE) AS /IS  
D2 DATE;  
BEGIN  
D2 := D1 + 30;  
DBMS\_OUTPUT.PUT\_LINE (TO\_CHAR(D2)); TO\_CHAR is optional  
END;

**OR**

CREATE OR REPLACE PROCEDURE NEXTMONTH (D1 DATE) AS /IS  
BEGIN  
DBMS\_OUTPUT.PUT\_LINE (TO\_CHAR (D1+30)); TO\_CHAR is optional  
END;

*(1 mark for writing correct procedure header)*

*(½ mark for correct calculation)*

*(½ mark for display)*

**NOTE: Deduct ½ mark if argument not specified in header**



## SECTION B

4. (a) `txtPercentage.Enabled = False`  
`txtGrade.Enabled = False`

**OR**

`txtPercentage.Locked = True`  
`txtGrade.Locked = True`

*(½ mark each valid command)*

- (b) **Method 1** **Method 2**

`txtFirstTerm.Text = " "` `txtFirstTerm = " "`  
`txtSecondTerm.Text = " "` `txtSecondTerm = " "`  
`txtPercentage.Text = " "` `txtPercentage = " "`  
`txtGrade.Text = " "` `txtGrade = " "`  
`chkCadet.Value = False` `chkCadet.Value = 0`

`/ vbUnchecked`

*(½ mark for clearing all text boxes)*

*(½ mark for clearing checkbox)*

**NOTE: Ignore the event header**

- (c) `Sub cmdCalcPer_Click ()`  
`Dim first, second, total, per`  
`first = Val(txtFirstTerm.Text)`  
`second = Val(txtSecondTerm.Text)`  
`total = first + second`  
`per = total / 2`  
`If chkCadet.value = 1 Then`  
`per = per + 3`  
`End If`  
`txtPercentage.Text=per`  
`End Sub`

`Dim` is optional

`Val` is optional

OR

`per = total / 200*100`

OR

`True` or `vbChecked`

*(1 mark for calculation of percentage before considering the NCC criteria)*

*(1 mark for conditional statement)*

*(1 mark for incrementing the percentage)*

*(1 mark for refreshing textbox)*

```

(d) Sub cmdCalcGrade_Click()
    Dim per, grade
    per = Val(txtPercentage.Text)
    If optMedical.Value = True And per >= 80 Then
        grade = "A"
    ElseIf optMedical.Value = True And per >=60 Then
        grade = "B"
    ElseIf optMedical.Value = True Then
        grade = "C"
    End If
    If optNonMedical.Value = True And per >= 75 Then
        grade = "A"
    ElseIf optNonMedical.Value = True And per >= 50 Then
        grade = "B"
    Else
        grade = "C"
    End If
    txtGrade.Text=grade
End Sub

```

**OR**

```

Sub cmdCalcGrade_Click()
    Dim per, grade
    per = Val(txtPercentage.Text)
    If optMedical.Value = True
        If per >= 80 Then
            grade = "A"
        ElseIf per>=60 Then
            grade = "B"
        Else
            grade = "C"
        End If
    Else
        If per >= 75 Then
            grade = "A"
        ElseIf per >= 50 Then
            grade = "B"
        Else
            grade = "C"
        End If
    End If

```

```
End If
txtGrade.Text=grade
End Sub
```

**OR**

*Any other equivalent logic*

*(1/2 mark for each condition- 3½ marks if all the conditions are correct)*

*(1/2 mark for displaying grade)*

**NOTE:** Ignore overlapping range of marks i.e.  or

5. (a) Corrected Code with errors underlined:

```
Private Sub Command1_Click()
Dim p As Integer, i As Integer
p = 1 1
For i = 1 To 50 2
    p = p + 1
    If p = 5 Then
        p = 1 3
        Print "p is equal to 5"
    Else
        Print "p is not equal to 5"
    End If 4
Next 5
End Sub
```

*(½ mark each for identification and correction of any four errors)*

**OR**

*(Only 1 mark to be given for only identification of any four errors)*

**NOTE:** Ignore underlining

- (b) **Output :**

0  
14  
16  
6

*(½ mark each line of correct output)*

**NOTE:** No marks to be deducted if the entire output is displayed in single line.

(c) **Converted Code using For Next Loop :**

```
Private Sub cmdFactorial_Click()  
    Dim factorial As Long  
    Dim counter As Integer  
    Dim strAnswer As String  
    counter = Val(txtFactorial.Text)  
    factorial = 1  
    For a = counter to 1 Step -1  
        factorial = factorial * a  
    Next  
    Print "Factorial of " & txtFactorial.Text & " is " & factorial  
  
End Sub
```

**OR**

Any equivalent code using FOR loop for finding factorial

*(½ mark for correct syntax of FOR..NEXT loop)*

*(½ mark for initialization of factorial)*

*(½ mark for correct increment/decrement)*

*(½ mark for calculation of factorial i.e. factorial = factorial \* a)*

**NOTE: Ignore all other additional lines**

(d) Sub Procname(str)

```
Dim newstr, L  
Print UCASE (str)  
L = LEN(str)  
Print L  
newstr = UCASE (LEFT(str, 1)) OR (UCASE (MID (str,1,1))  
    + LCASE (MID(str, 2, L-2))  
    + UCASE(RIGHT(str, 1)) OR (UCASE (MID (str,L-1,1 ))  
Print newstr  
End Sub
```

**OR**

```
Sub Procname(str)  
Print UCASE(str)  
Print LEN (str)
```

```

Print UCASE(LEFT(str, 1)) OR UCASE (MID (str,1,1))
      & LCASE (MID (str, 2, LEN (str)-2))
      & UCASE (RIGHT (str, 1) ) OR UCASE (MID (str, LEN(str)-1,1))
End Sub

```

*(½ mark for correct procedure header with argument)*

*(1 mark for displaying string in uppercase) (1 mark for displaying length of string)*

*(½ mark for converting first and last character)*

*(½ mark for converting remaining characters in tower case)*

*(½ mark for correct concatenation)*

### SECTION C

6.	(a)	If i*10 is used	If X*10 is used	If SUM*10 is used	ERROR _____
		0	10	10	
		0	20	-10	
		0	30	20	If error is mentioned
		0	40	-20	
		0	50	30	

*(½ mark each for first 3 lines = 1½ marks)*

*(½ mark for last two lines)*

*Note: Full marks if error is mentioned*

(b) *(Full 2 marks if ERROR is mentioned)*

*(Full 2 marks if ENAME “WARD” or “ALLEN” is mentioned)*

**NOTE:** Full 2 marks to be given if the student has attempted all the other parts of Question 6 correctly [Other than 6(b) part] irrespective of attempt of 6 (b)

(c) **Example:**

```

DECLARE
var1 NUMBER (4);
var2 var1%TYPE;

```

OR

```

v_empno Employee.empno%TYPE;

```

**Explanation:**

Var2 is of the same data type as var1 or v\_empno is of the same data type as empno field of Employee table.

*(1 mark for any one example)*

*(1 mark for explanation of example or definition)*

Optional

(d) CREATE OR REPLACE FUNCTION CheckDiv(a NUMBER, b NUMBER)  
RETURN NUMBER AS OR IS  
K NUMBER;  
BEGIN  
IF MOD (a, b) = 0 THEN  
K := 1;  
ELSE  
K := 0;  
END IF;  
RETURN K;  
END;

**OR**

Any equivalent code

*(1 mark for function name and arguments)*

*(½ mark for RETURN in function header)*

*(1 mark for conditional statement)*

*(½ mark for using MOD function correctly - OR equivalent function/ expression)*

*(½ mark for BEGIN and END)*

*(½ mark for returning the value)*

7. (a) CREATE TABLE FLIGHT  
(Flight\_No NUMBER(4),  
Origin VARCHAR2(30) NOT NULL,  
Destination VARCHAR2(30) NOT NULL,  
Seats NUMBER(3),  
Flt\_Date DATE,  
Rate NUMBER(7,2),  
CONSTRAINT PK PRIMARY KEY(Flight\_No)  
) ;

**OR**

```
CREATE TABLE FLIGHT
    (Flight_No NUMBER(4) CONSTRAINT PK PRIMARY KEY,
    Origin VARCHAR2(30) NOT NULL,
    Destination VARCHAR2(30) NOT NULL,
    Seats NUMBER(3),
    Flt_Date DATE,
    Rate NUMBER(7,2) ,
);
```

NOTE: Constraint name need not be mentioned

*(½ mark CREATE TABLE flight)*

*(½ mark for fields with data types)*

*(½ mark for constraint primary key)*

*(½ mark for NOT NULL)*

(b) 

```
SELECT * FROM Flight WHERE Destination = Selcet
Destination FROM Flight WHERE Flight_No=9001);
```

**OR**

```
SELECT * FROM Flight WHERE Destination IN (SELECT
Destination FROM Flight WHERE Flight_No=9001);
```

*(1 mark for parent query)*

*(1 mark for sub query)*

(c) 

```
BEGIN
    UPDATE Flight SET Rate = Rate + .05 * Rate;
END;
```

*(½ mark for BEGIN and END)*

*(½ mark focusing keywords UPDATE and SET)*

*(1 mark for correct updation)*

**Note: Explicit Cursor is also acceptable**

Optional

(d) 

```
CREATE OR REPLACE TRIGGER B_D_Flight
BEFORE DELETE ON Flight
BEGIN
    DBMS_OUTPUT.PUT_LINE("Ready for Deletion");
END;
```

Optional

```
CREATE OR REPLACE TRIGGER A_D_Flight  
AFTER DELETE ON Flight  
BEGIN  
DBMS_OUTPUT.PUT_LINE ("Records Deleted");  
END;
```

*(1/2 mark for each line except BEGIN and END - Total 3 marks) (1/2 mark for each BEGIN and END -Total 1 mark)*

**Important Note:**

All answers provided in the marking scheme are SUGGESTIVE. Examiners are requested to accept all possible equivalent alternative answers.

**QUESTION PAPER CODE 90**

**EXPECTED ANSWERS / VALUE POINTS**

**SECTION A**

1. (a) Data warehousing refers to the collection of data gathered and organized so that data can be easily analyzed, extracted or otherwise be used for further understanding of data.

Used to meet various short-term and long-term business objectives.

*(1 mark for any correct definition)*

*(1 mark for any correct usage/example)*

- (b) The **systems development life cycle** (SDLC) is a conceptual model used in project management that describes the stages involved in a software system development project.

Essential components are:

- Evaluation / Preliminary Investigation/ Feasibility Study
- Analysis
- Design
- Implementation / Coding
- Testing/Debugging
- Installation/ Parallel Run/ Final Execution
- Maintenance
- Review

*(1 mark for any correct definition/ full form)*

*(1 mark for any four components)*

*(1/2 mark if any two components are correct)*



- (c) A front-end tool is an application that users interact with directly i.e. the user-interface. For example Visual Basic OR Visual C++ OR Developer 2000 OR Delphi

A back-end tool is the program or application to which the front-end forwards requests to get requested data or perform a requested service.

For example Oracle OR MS-SQL Server OR Sybase OR MySQL OR Delphi

**(1 mark for any correct definition of front-end)**

**(1 mark for any correct definition of back-end)**

**(½ mark each if only example is correct)**

- (d) The entity-relationship modeling (or ER modeling) is a graphical representation of entities and their relationships to each other, used in organization of data within databases. An entity represents a discrete object and a relationship captures how two or more entities are related to one another.



**(2 mark for any equivalent correct definition)**

**(1 mark for equivalent ER diagram with required interpretation for student:subject -1:m)**

**(1 mark for equivalent ER diagram with required interpretation for subject:teacher-1:1 OR m:1)**

2. (a) Conditional statements control the flow of execution of a program. Conditional statements perform comparisons and take appropriate actions depending on the outcome of such comparisons. VB provides two conditional statements:

- if... statement
- Select Case statement

**(1 mark for each correct definition/explanation using suitable example)**

**(½ mark for naming each conditional statement)**

- (b) A Private variable declared in the general declarations section of a Form or code module is called a Module-level variable. The value of the module level variable is available to every procedure in that module.

A Public variable declared in the general declarations section of a Form or code module is called a Global variable. The value of a Global variable is available to any procedure, in any Form or code module.

**OR**

Variables declared using Public and Private keywords. The variables declared as public is accessible in all the procedures (throughout the project), whereas the private variables are specific to the procedure/module in which they are declared.

**OR**

Any equivalent definition

*(1 mark for each correct definition/explanation)*

- (c) **ODBC (Open Database Connectivity)** is used to access any data from any application, regardless of which DBMS is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS.

**OLE-DB (Object Linking Embedding-Database)** is an open specification system-level programming interface designed to build on the success of ODBC by providing an open standard for accessing all kinds of data.

*(1 mark each for any correct/equivalent definition)*

**OR**

*(½ mark each for only correct full forms)*

- (d) **SDI (Single Document Interface)** forms are the forms, which allow us to work with a single document at a time. Example: WordPad

**MDI (Multiple Document Interface)** form is one in which we can view and work with several documents at once. Example: Microsoft Excel

An MDI Form acts as a container or Parent for other Child Forms. A project can have only one MDI Form or Parent form, but any number of Child Forms. The MDI Form has to be specifically added to a project by using Project/Add MDI Form from the menu. Child Forms can be created from any regular form by setting the MDIChild-property to True.

*(1 mark for any correct definition/explanation of SDI)*

*(1 mark for any correct definition/explanation of MDI)*

*(1 mark for explaining child form)*

*(1 mark for explaining parent form)*

**OR**

*(½ mark each for only mentioning full forms of SDI and MDI)*

3. (a) DML (Data Manipulation Language): The category of SQL statements used to manipulate the data already stored in the database.

Example: SELECT, INSERT, DELETE and UPDATE.

DCL (Data Control Language): The category of SQL statements that control access to the data and to the database.

Examples: GRANT and REVOKE

*(½ mark for each correct definition)*

*(½ mark each for any one correct example of both categories)*

**OR**

*(½ mark for only mentioning the full forms of DML and DCL)*

**Note: Deduct ½ Mark if the DCL is taken as TCL**

- (b) NULL represents an unknown value

**OR**

NULL is absence of any value.

**OR**

NULL is No Value

**OR**

NULL is empty value

**OR**

NULL is not entered value

**OR**

NULL is an unassigned value

**OR**

NULL is not meaningful value

An arithmetic operation involving a NULL returns NULL.

*(1 mark for any correct definition of NULL)*

*(1 mark for “an arithmetic operation involving a NULL returns NULL”)*

- (c) A sub-query is used when an SQL query specification is nested in another query specification. The sub-query gets executed first and its result is used to execute the parent query.

*(1 mark for definition of sub-query)*

*(1 mark for “sub-query gets executed first”)*

- (d) DBMS\_OUTPUT.PUT\_LINE() **OR** DBMS\_OUTPUT.PUT()  
SET SERVEROUTPUT ON

*(1 mark for each command)*

- (e) Row level trigger is fired once for each record being updated/inserted/deleted while. Statement level trigger is fired just once per statement even if more than one record is affected.

Row level trigger contains the statement FOR EACH ROW in the trigger definition while Statement level trigger does not.

*(1 mark each for any correct definition)*

**OR**

*(Full 2 marks for correct example with explanation)*

**OR**

*(Only 1 mark if only example without explanation is given)*

## SECTION B

4. (a) `txtInterest.Enabled=False` OR 

<code>txtInterest.Locked=True</code>
--------------------------------------

  
`txtAmount.Enabled=False` OR 

<code>txtAmount.Locked=True</code>
------------------------------------

*(1 mark for each line of the code)*

**NOTE: Ignore any additional line(s)**

- (b) `txtPrincipal.Text= " "`  
`txtRate.Text= " "`  
`txtInterest.Text= " "`  
`txtAmount.Text= " "`  
`opt5years.value=True` OR 

<code>opt5years.value=1</code>
--------------------------------

  
`txtPrincipal.setFocus`

*(1 mark for clearing the textboxes)*

*(1/2 mark for setting the default choice of the checkbox)*

*(1/2 mark for setFocus)*

- (c) `Sub txtPrincipal_Change( )`  
`If NOT(RIGHT(txtPrincipal.Text, 1)>="0" AND RIGHT(txtPrincipal.Text,1)<="9") Then`  
`txtPrincipal.Text= " "` OR 

<code>MsgBox "Enter Only Numeric value"</code>
------------------------------------------------

  
`End If`  
`End Sub`  
  
`Sub txtRate_Change()`  
`If NOT (RIGHT(txtRate.Text,1)>="0" AND RIGHT(txtRate.Text,1)<="9") Then`  
`txtRate.Text= " "` OR 

<code>MsgBox "Enter Only Numeric value"</code>
------------------------------------------------

  
`End If`  
`End Sub`

**OR**

```
Sub txtPrincipal_Change()  
If Not(IsNumeric(txtPrincipal.Text)) Then  
txtPrincipal.Text= " " OR MsgBox "Enter Only Numeric value"  
End If  
End Sub  
  
Sub txtRate_Change()  
If Not(IsNumeric(txtRate.Text)) Then  
txtRate. Text= " " OR MsgBox "Enter Only Numeric value"  
End If  
End Sub
```

**OR**

Any other equivalent code

***(1 mark for each)***

**(d)** Dim P,R,T,A,S  
P=Val(txtPrincipal.Text)  
R=Val(txtRate.Text)  
If opt5years.value=True Then  
T=5  
Elseif opt10years. valuesTrue Then  
T=10  
Elseif opt15years.value=True Then  
T=15  
End If  
S= P\*(1+R/100)^T  
txtInterest.Text=S  
txtAmount.Text = P + 3

**OR**

Any other equivalent code considering the correct formula of compound interest.

***(1 mark for determining T)***

***(1 mark for calculating Amount)***

***(1 mark for calculating Compound Interest)***

***(1 mark for refreshing the text boxes - ½ mark each)***

5. (a) **Corrected Code with errors underlined:**

```
Private Sub cmdLeapyear Click() 1
    Y=Val (txtYear)
    IF Y Mod 100 = 0 AND Y Mod 400 = 0 THEN 2
        MsgBox "It is a Leap Year"
    ELSEIF Y Mod 4=0 THEN
        MsgBox "It is a Leap Year"
    ELSE
        MsgBox "It is not a Leap Year" 3
    END IF
    End Sub 4
```

*(½ mark for each error)*

**OR**

*(Only 1 mark to be given for only identification of any four errors)*

**NOTE: Ignore underlining**

**Don't deduct marks if only one THEN is identified**

(b) **Output :**

0  
1  
1  
2

*(½ mark for each line of output)*

**NOTE: No marks to be deducted if the entire output is displayed in single line.**

(c) **Method 1**

```
If marks >= 85 Then
    comment.Caption = "Excellent"
Elseif marks >= 60 Then
    comment.Caption = "Above Average".
Elseif marks>= 50 Then
    comment.Caption = "Average"
Else
    comment.Caption = "Work harder"
End If
```

**Method 2**

```
If marks >= 85 Then
    comment.Caption = "Excellent"
Else
    If marks>= 60 Then
        comment.Caption = "Above Average"
    Else
        If marks>= 50 Then
            comment.Caption = "Average"
        Else
            comment.Caption = "Work harder"
        End If
    End If
End If
```

*(½ marks for each condition - Total 2 marks)*

<p>(d) Function SumEven(N)  SumEven=0  For k = 2 To N Step 2  SumEven = SumEven + k  Next  End Function</p>	<p>Function SumEven(N)  SumEven=0  For k = 1 To N  If k mod 2 =0 Then  SumEven = SumEven + k  End If  Next  End Function</p>
---------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------

**OR**

Any other equivalent logic

*(1 mark for function header and correct function return -  $\frac{1}{2} + \frac{1}{2}$ )*

*(1 mark for the correct loop logic)*

*(1 mark for correct summation)*

*(1 mark for increment)*

### SECTION C

6. (a) **Output :**

10

18

24

28

*( $\frac{1}{2}$  mark for each line)*

**OR**

*(Full 2 marks if ERROR is mentioned)*

**NOTE: No marks to be deducted if the entire output is displayed in single line.**

(b) **Output :**

8975

*(2 marks for the correct output)*

(c) The IN mode means that the parameter is read only i.e. the procedure can use or reference the value of the parameter but cannot change it.

The OUT mode is write only i.e. the procedure can set, but not reference, the value of the parameter. This mode is used to return values from the procedure back to the calling program.

**OR**

Any other equivalent definition or explanation through example.

*(1 mark each)*

Optional

(d) CREATE  FUNCTION Power (a NUMBER(5) , b NUMBER (5))  
RETURN NUMBER(8) AS / IS  
P NUMBER(8):=1;  
K NUMBER(5):=1;  
BEGIN  
FOR K IN 1 .. b LOOP  
P = P \* a;  
END LOOP;  
RETURN P;  
END;

**Marks to be given if any of *the* following loops is used instead of FOR LOOP:**

WHILE K >= b	LOOP	LOOP
LOOP	P = P * a;	P = P * a;
P = P * a;	K = K + 1;	K = K + 1;
K = K + 1;	EXIT WHEN K > b;	IF K > b THEN
END LOOP;	END LOOP;	EXIT;
		END IF;
		END LOOP;

**OR**

```
CREATE  FUNCTION Power (a NUMBER, b NUMBER)
RETURN NUMBER AS OR 
BEGIN
RETURN a**b;
END;
```

**OR**

Any equivalent code

*(1 mark for function name and arguments)*

*(½ mark for RETURN in function header)*

*(1½ mark for computing power)*

*(½ mark for BEGIN and END)*

*(½ mark for returning the value)*

**NOTE: No marks to be deducted if function name has been changed.**



7. (a) CREATE TABLE EMPLOYEE  
 (Emp\_ID NUMBER(8),  
 First\_Name VARCHAR2(25) NOT NULL,  
 Last\_Name VARCHAR2(25) ,  
 Date\_Join DATE,  
 Basic\_Sal NUMBER(8,2),  
 Dept\_ID NUMBER(3),  
 CONSTRAINT PK PRIMARY KEY(Emp\_ID)  
 ) ;

**OR**

CREATE TABLE EMPLOYEE  
 (Emp\_ID NUMBER(8) CONSTRAINT PK PRIMARY KEY,  
 First\_Name VARCHAR2(25) NOT NULL,  
 Last\_Name VARCHAR2(25),  
 Date\_Join DATE,  
 Basic\_Sal NUMBER(8,2) ,  
 Dept\_ID NUMBER(3)  
 ) ;

**NOTE: Constraint name need not be mentioned**

*(½ mark CREATE TABLE Employee)*

*(½ mark for fields with data types)*

*(½ mark for constraint primary key)*

*(½ mark for NOT NULL)*

(b) SELECT First\_Name, Date\_Join, Dept\_ID FROM Employee  
 WHERE Date Join >= '20-Mar-91' AND Date\_Join <= '31-Dec-91'  
 ORDER BY Date\_Join;

*(½ mark for correct SELECT)*

*(1 mark for WHERE)*

*(½ mark for ORDER BY)*

**NOTE: 1 mark to be given if student has written no Hire\_Date is available in the table.**

(c) CREATE VIEW V1 AS SELECT Emp\_ID, 0.12 \* Basic\_Sal "Bonus" FROM Employee;

**OR**

CREATE VIEW V1 AS SELECT Emp\_ID, 0.12 \* Basic\_Sal As Bonus FROM Employee;

*(½ mark for CREATE VIEW)*

*(1 mark for SELECT)*

*(½ mark for naming the column as Bonus)*

(d) DECLARE  
dept Employee.Dept\_ID%TYPE;  
BEGIN  
dept := & d;  
DELETE FROM Employee WHERE Dept\_ID = dept;  
END;

**OR**

BEGIN  
DELETE FROM Employee WHERE Dept\_ID = & d;  
END;

**OR**

Any other equivalent code

*(1 mark for accepting value from the user)*

*(1 mark for BEGIN and END)*

*(1 mark for DELETE)*

*(1 mark for correct WHERE)*

**NOTE: No marks to be given if only BEGIN and END are written.**

**Important Note:**

All answers provided in the marking scheme are SUGGESTIVE. Examiners are requested to accept all possible equivalent alternative answers.