

# Informatics Practices (Theory Code : 065)

## Marking Scheme

### Sample Question Paper - I

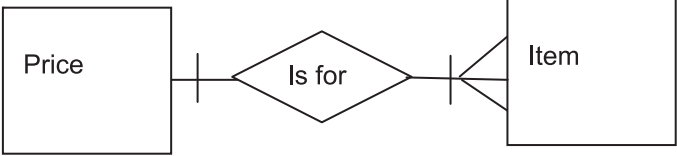
**TIME : 3 Hrs**

**MM : 100**

#### Note

1. This question paper is divided into 3 sections.
2. Section - A consists of 30 marks.
3. Section - B and Section - C are of 20 marks each.
4. Answer the questions after carefully reading the text.

No.	Answers	Marks
	<b>SECTION - A</b>	
Q 1.		
(a)	i) Open Source Software ii) GNU's Not Unix iii) World Wide Web Consortium iv) Free Open Source Solutions <i>( ½ Mark each for full form of each term)</i>	2
(b)	The front-end is the part of a software system that interacts directly with the user, and the back-end comprises the components that process the output from the front-end. <i>( 2 Marks for correct comparison)</i>	2
(c)	Data mining can be used by financial companies to rank customers on the basis of past payment behavior. This helps the companies to decide whether or not to approve further loans , credit cards etc. <i>( 2 Marks for correct answer)</i>	2
(d)	The purpose of feasibility study is to assess the viability of the proposed system. During the feasibility study, projects are typically evaluated in areas of economical, operational, and technical feasibility. The outcome of the feasibility study indicates whether or not to proceed with the proposed system. <i>( 2 Marks for correct answer)</i>	2
(e)	One to one and many to one relationship	2

No.	Answers	Marks				
Q2	 <p>(1 Mark for identifying the type of relationship)</p> <p>(1 Mark for ER diagram)</p>					
(a)	<p>ORACLE</p> <p>Microsoft SQL or MySQL</p> <p>Microsoft Access</p> <p>Firebird</p> <p>DB2</p> <p>Sybase</p> <p>(1mark each for naming any two correct RDBMS)</p>	2				
(b)	<p>SQL is the Structured Query Language used to interact with the RDBMS.</p> <p>The SQL Subcategories are:</p> <p>DML (INSERT, UPDATE, DELETE)</p> <p>DDL (CREATE TABLE, DROP TABLE, ALTER TABLE)</p> <p>DCL (GRANT, REVOKE)</p> <p>TCL (COMMIT, ROLLBACK)</p> <p>(1mark for the correct definition of SQL)</p> <p>(½ mark each for correctly naming any 3 subcategories)</p> <p>(½ mark each for the correct command in each category (Only one command in each category))</p>	4				
(c)	<table border="1" data-bbox="197 1659 1390 2020"> <thead> <tr> <th data-bbox="197 1659 788 1711">Decision Control</th> <th data-bbox="788 1659 1390 1711">Looping Control</th> </tr> </thead> <tbody> <tr> <td data-bbox="197 1711 788 2020"> <p>1. Decision control structures are used to implement decisions based on conditions. We check the condition and allow to execute code if the condition is found true else some other set of code is executed. This can be achieved using If then Else</p> </td> <td data-bbox="788 1711 1390 2020"> <p>1. Looping means iterations or repetition of some set of code. The same set of code can be executed more than once on entering into the Looping Construct.</p> </td> </tr> </tbody> </table>	Decision Control	Looping Control	<p>1. Decision control structures are used to implement decisions based on conditions. We check the condition and allow to execute code if the condition is found true else some other set of code is executed. This can be achieved using If then Else</p>	<p>1. Looping means iterations or repetition of some set of code. The same set of code can be executed more than once on entering into the Looping Construct.</p>	
Decision Control	Looping Control					
<p>1. Decision control structures are used to implement decisions based on conditions. We check the condition and allow to execute code if the condition is found true else some other set of code is executed. This can be achieved using If then Else</p>	<p>1. Looping means iterations or repetition of some set of code. The same set of code can be executed more than once on entering into the Looping Construct.</p>					

No.	Answers		Marks	
Q.3	End if command in PL/SQL.		4	
	<p>2. Decision Control Structure in PL/SQL are</p> <p>IF .. THEN..ENDIF statement</p> <p>IF .. THEN ..ELSE..ENDIF statement</p> <p>IF .. THEN .. ELSIF..ENDIF statement</p> <p>If &lt;Condition&gt; Then</p> <p>&lt;Executable Statements&gt;</p> <p>Else</p> <p>&lt;Executable Statements&gt;</p> <p>End If</p>	<p>2. Looping Control in PL/SQL are</p> <p>Basic LOOP</p> <p>FOR Loop</p> <p>WHILE Loop</p>		
	<p>3. In decision control, statement(s) is/are executed maximum one time only whereas in loops statement(s) is/are executed more than one time</p>			
	<p><i>(2 marks for correct difference)</i></p> <p><i>(1 mark each for command of control structure)</i></p>			
(a)	<p>An event represents the state of the keyboard keys, the location of the mouse, and the state of the mouse buttons, which occurs as the result of user interaction with an element.</p> <p>For example, when a user clicks a button, an event is generated.</p> <p>Visual Basic is one example of an event driven programming language.</p> <p><i>(1 mark for the correct definition)</i></p> <p><i>(1 mark for naming the language)</i></p>		2	
(b)	MsgBox ()	InputBox ()	4	
<p>1. As the name explains it's a mechanism to show a small dialog box, which pops up over the existing application for displaying custom messages.</p>	<p>1. The InputBox is for displaying a message along with collecting User Inputs.</p>			
<p>2. Message Box can also be used to gather user responses but they are limited to the user actions such as whether the user has Clicked OK</p>	<p>The Input Box gathers user data on string form.</p>			

No.	Answers		Marks
	button, or Cancel Button or Yes button or No button etc.		
	<b>Example</b> MsgBox "Hello User"	<b>Example</b> Dim Age Age=InputBox ("Enter Age")	
(c)	<i>(2 marks for any two correct differences)</i> <i>(1 mark for each example)</i>		
	<b>ADODB</b>	<b>ADO DC</b>	<b>4</b>
	It is an object library available in visual basic and a reference can be added by going to the menu Project > References.	It is a data control like other controls available in the toolbox. This control can be added by going to the Menu Project > Components.	
	ADODC is class library.	ADODC is just a data control,	
	Ex: Programmer Controlled Navigation of Data from table	Ex: This control does pre-defined navigations Move First, Move Last, Move Next, Move Previous	
	<i>(2 marks for any two correct differences)</i> <i>(1 mark for each example)</i>		
	<b>Section - B</b>		
<b>Q4.</b>			
(a)	&Check Status <i>(½ mark for '&amp;' symbol)</i> <i>(½ mark for the text 'Check Status')</i>		1
(b)	On Click event of the cmdCloseMe button write the following: END <i>(½ mark for mentioning the event or writing the equivalent code for it)</i> <i>(½ mark for the code)</i>		1
(c)	On KeyPress Event write the following code fragment: If Not(IsNumeric(txtDD.Text)) Then		4

No.	Answers	Marks
	<p>Msgbox ("Please enter numbers only")</p> <p>End If</p> <p><i>(2½ marks for the correct validation using the IF statement)</i></p> <p><i>(1 mark for displaying a message using the MsgBox)</i></p> <p><i>(½ mark for End If)</i></p>	
(d)	<p>Procedure Coding:</p> <p>Dim strDate as String</p> <p>strDate = trim(txtDD.Text) &amp; "-" &amp; trim (txtMM.text) &amp; "-"&amp; trim(txtYYYY.text)</p> <p>If Not IsDate(strDate) Then</p> <p>MsgBox "Please Enter a Valid Date"</p> <p>txtDD.Text = ""</p> <p>txtMM.Text=""</p> <p>txtYYYY.text=""</p> <p>txtDD.SetFocus</p> <p>End If</p> <p><i>(1 mark for extracting the value from the textbox)</i></p> <p><i>(1 mark for the correct validation using the IF statement)</i></p> <p><i>(1 mark for displaying a message using the MsgBox)</i></p> <p><i>(1 mark for clearing the textboxes)</i></p>	4
Q5.	<p>(a) Hello</p> <p>10</p> <p>450</p> <p>Bye</p> <p><i>(½ mark for each correct line of output)</i></p>	2
(b)	<p>SELECT CASE Total</p> <p>CASE IS&gt;=75</p> <p>Grade = "A"</p>	2

No.	Answers	Marks
	<p>CASE 50 TO 74 Grade = "B" CASE 33 TO 49 Grade = "C" CASE ELSE Grade = "D" END SELECT</p> <p><i>(½ mark for SELECT statement)</i> <i>(½ mark for case value)</i> <i>(½ mark for CASE ELSE statement)</i> <i>(½ mark for END SELECT statement)</i></p>	
(c)	<p>Corrected code:</p> <pre> DIM Total AS INTEGER Total=30.0 DO WHILE Total&gt;=20 Total=Total-5 LOOP PRINT Total </pre> <p><i>( ½ Mark each for identifying and correcting each of four errors)</i></p>	2
(d)	<pre> FOR Num = 1 TO 5 PRINT Num NEXT Num </pre> <p><i>(½ mark for using correct control variable)</i> <i>(½ mark for using correct initial value)</i> <i>(½ mark for using correct final value)</i> <i>(½ mark for NEXT statement)</i></p>	
(e)	<p>Output:</p> <p>Tech and Kids</p> <p>False</p>	2

No.	Answers	Marks
	110 7 <i>(½ mark for each correct line of output)</i>	
Q6.	<b>Section - C</b>	
	(a) Code: CREATE TABLE Employee ( EmpID        NUMBER (6)        PRIMARY KEY, EmpName      VARCHAR2 (20)      NOT NULL, EmpAddress   VARCHAR2 (30), EmpPhone     VARCHAR2 (10), EmpSal        NUMBER (9,2))  <i>(½ mark for CREATE TABLE )</i> <i>(½ mark for fields with data types)</i> <i>(½ mark for PRIMARY KEY constraint)</i> <i>(½ mark for NOT NULL constraint)</i>	2
	(b) Code: BEGIN UPDATE EMPLOYEE; SET EmpSal = EmpSal+EmpSal * &SAL_PER/100; END;  <i>(1 mark for UPDATE Employee statement)</i> <i>(1 mark for correct SET statement)</i> <i>(2 marks for using the correct expression to increase salary)</i>	4
(c) Code: DECLARE V_Sal EMP.SAL%TYPE; CURSOR C_Emp IS SELECT SAL FROM EMP;	4	

No.	Answers	Marks
	<pre> BEGIN OPEN C_Emp;   LOOP FETCH C_Emp INTO V_Sal; V_Sal := V_Sal * 1.1; EXIT WHEN C_Emp%NOTFOUND; DBMS_OUTPUT.PUT_LINE (' Raised Salary is: '    V_Sal); END LOOP; CLOSE C_Emp; END; </pre> <p><i>(1 mark for creating the Cursor)</i>  <i>(1 mark for the loop)</i>  <i>(1 mark for using DBMS_OUTPUT.PUT_LINE)</i>  <i>(1 mark for correct expression for increase of Salary)</i></p>	
Q7. (a)	<p>Code:</p> <pre> CREATE VIEW VU_EMP AS      SELECT EName, Dname, Job , Sal                            FROM Emp, Dept                            WHERE Emp.DeptNo = Dept.DeptNo                            AND Emp.DeptNo IN (10,20); </pre> <p><i>(½ Mark for CREATE VIEW command)</i>  <i>( ½ Mark for SELECT statement)</i>  <i>(1 Mark for WHERE clause)</i></p>	2
(b)	<p>Code:</p> <pre> CREATE OR REPLACE FUNCTION FindBonus (V_Amount IN NUMBER) RETURN NUMBER AS V_Bonus NUMBER; BEGIN </pre>	4

No.	Answers	Marks
(c)	<p>V_Bonus := V_Amount * 1.2;  RETURN V_Bonus;  END;  Example of use:  SQL&gt; SELECT Ename, FindBonus(Sal) FROM Emp;  <i>(1 mark for correct function header)</i>  <i>(½ mark for correct positions of BEGIN and END)</i>  <i>(1 mark for correct calculation expression)</i>  <i>(½ mark for RETURN)</i>  <i>(1 mark for Example)</i></p> <p>Code:  CREATE OR REPLACE PROCEDURE RAISE_SALARY  (V_EmpNo EMP.EMPNO%TYPE, SAL_PERCENTAGE NUMBER)  AS  V_SAL NUMBER (10,2);  BEGIN  UPDATE EMP  SET SAL= SAL+(SAL*(SAL_PER/100))  WHERE EMPNO = V_EMPNO;  SELECT SAL INTO V_SAL  FROM EMP  WHERE EMPNO = V_EMPNO;  DBMS_OUTPUT.PUT_LINE ('The New Salary is : '    V_SAL);  END;  <i>(1 mark for correct Procedure header with required parameters)</i>  <i>(1 mark for correct UPDATE statement)</i>  <i>(1 mark for correct SELECT statement)</i>  <i>(1 mark for displaying the changed value of salary)</i></p>	4