

Informatics Practices (Theory Code : 065)

Marking Scheme

Sample Question Paper - II

TIME : 3 Hrs

MM : 100

Note

1. This question paper is divided into 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
SECTION - A		
Q 1.		
(a)	<p>i. Unicode is an industry standard allowing computers to consistently represent and manipulate text expressed in most of the world's writing systems. Unicode's success at unifying character sets has led to its widespread and predominant use in the internationalization and localization of computer software.</p> <p>ii. GNU is a computer operating system composed entirely of free software. It stands for GNU's Not Unix; it was chosen because its design is Unix-like, but differs from Unix by being free software and containing no Unix code.</p> <p><i>(1 mark for each explanation)</i></p>	2
(b)	<p>Open source software allows a user or organization to tailor make the software according to its needs. In the case of proprietary software, customization is not possible at the user end as access to the source code is denied and solely controlled by the developer of the software.</p> <p><i>(2 marks for correct difference)</i></p>	2
(c)	<p>There are two key elements in Client / Server computing. The CLIENT and the SERVER. The CLIENT is one who requests the services and the SERVER is the one who is able to provide the requested services. It is not always necessary that the Client and the server are two separate machines. The same machine could be a server as well as a client. Such a computing model in which one process requests another process to do a task for it is called client - server computing.</p> <p>For example, a web browser is a client program at the user computer that may access information at any web server in the world.</p>	2

No.	Answers	Marks
	<p>Another example, in a network within a computer lab, the printer may be attached to a single computer and all other computers will be using that printer. In that case the computer to which the printer is attached becomes the printer server and all other machines using the printer become clients.</p> <p><i>(2 marks for explanation with example)</i></p> <p><i>OR</i></p> <p><i>(1 mark for only example without explanation)</i></p>	
(d)	<p>The tasks done during maintenance are :</p> <ul style="list-style-type: none"> • debugging the programs. • updating the system to accommodate changes in business conditions. • adding new functionality to the system. <p><i>(1 mark each for any 2 tasks)</i></p>	2
(e)	<p>UML stands for Unified Modeling Language.</p> <p>The Unified Modeling Language (UML) is a general-purpose modeling language used for visualizing, specifying, constructing and documenting the artifacts of software-intensive systems.</p> <p><i>(1 mark each for both parts)</i></p>	2
Q2		
(a)	<p>The DISTINCT keyword is used to restrict duplicate rows. It can easily remove multiple occurrences of an item so that it occurs only once.</p> <p>The DISTINCT clause removes the duplicate rows.</p> <p>SELECT DISTINCT Activity FROM Student</p> <p><i>(1 mark for stating the usage of DISTINCT keyword</i></p> <p><i>1 mark for example)</i></p> <p><i>Or</i></p> <p><i>(2 marks for explanation of usage of DISTINCT keyword with the help of example)</i></p>	2
(b)	<p>SQL functions are predefined stored software that manipulates submitted elements within SQL statements to carry out certain tasks and returns some value.</p> <p>Single-row functions return a single result row for every row of a queried table or view. The subcategories of single row functions are as follows</p>	4

No.	Answers	Marks
	<p>1. Numeric functions - They accept numeric input and return numeric values. COS, POWER, EXP</p> <p>2. Character functions - These are functions that work on character data types. They may either return character values or any other data type. The character functions that return character values are: CHR, CONCAT, INITCAP. The character functions that return number values are: ASCII, INSTR, LENGTH</p> <p>3. Datetime functions These are functions that are designed to accept any of the three types of data (date, timestamp, and interval) and to return a value of one of these types. The datetime functions are: ADD_MONTHS, CURRENT_DATE, SYSDATE</p> <p>4. Conversion functions These functions convert a value from one datatype to another. For Ex.: ASCIISTR, TO DATE, TO NUMBER</p> <p>5. Miscellaneous function : These are functions which are used to perform general tasks. For Ex.: NVL, DECODE</p> <p><i>(1 mark for definition of a function)</i></p> <p><i>(1 mark for Single row function)</i></p> <p><i>(1 mark for naming categories)</i></p> <p><i>(1 mark for writing one example in each category)</i></p> <p>(c) A row-level trigger fires once for each row that is affected by a triggering event. For example, if deletion is defined as a triggering event on a table and a single DELETE command is issued that deletes five rows from the table, then the trigger will fire five times, once for each row.</p> <p>A statement-level trigger fires once per triggering statement regardless of the number of rows affected by the triggering event. In the prior example of a single DELETE command deleting five rows, a statement-level trigger would fire only once.</p> <p>The following example is a before row-level trigger that calculates the commission of every new employee belonging to department 30 that is inserted into the emp table.</p> <pre>CREATE OR REPLACE TRIGGER emp_comm_trig BEFORE INSERT ON emp FOR EACH ROW BEGIN IF :NEW.deptno = 30 THEN DBMS_OUTPUT.PUT_LINE(:NEW.sal * .4); END IF;</pre>	4

No.	Answers	Marks
	<p>END;</p> <p>The following is an example of a simple before statement-level trigger that displays a message prior to an insert operation on the emp table.</p> <pre>CREATE OR REPLACE TRIGGER emp_alert_trig BEFORE INSERT ON emp BEGIN DBMS_OUTPUT.PUT_LINE('New employees are about to be added'); END;</pre> <p><i>(2 marks for correct difference)</i> <i>(1 mark each for example of both)</i></p>	
Q3.		
(a)	<p>SDI: A function that allows an application to display only one document at a time.</p> <p>Example: Notepad OR MS-Paint</p> <p>MDI: A function that allows an application to display and lets the user work with more than one document at the same time.</p> <p>Example: MS Word OR MS Excel</p> <p><i>(½ Mark each for correct definition of MDI and SDI)</i> <i>(½ Mark each for any correct example of MDI and SDI)</i></p>	2
(b)	<p>System Defined Functions</p> <p>System defined functions are the pre-defined or in-built or library functions in the VB interpreter which can be used directly in any program to perform common tasks.</p> <p>There are 3 types of system defined functions : String function (left(), space() etc.), arithmetic function (val(), mid(), etc.) and calendar function (day(), month(), date(), etc..</p> <p>User Defined Functions</p> <p>User defined functions are the functions created by the programmer to perform specialized tasks.</p> <pre>Public Function funCheckNumeric(num as Variant) As String If IsNumeric(num) then funCheckNumeric = 'true' else</pre>	4

No.	Answers	Marks
	<pre> funCheckNumeric = 'false' EndIf End Function OR Public Function funCheckNumeric(num as Variant) As Boolean If IsNumeric(num) then funCheckNumeric = True else funCheckNumeric = False EndIf End Function </pre> <p><i>(2 marks for difference between System Defined and User Defined functions)</i> <i>(2 marks for defining a function)</i></p> <p>(c) Common Dialog Box is used to perform common application tasks, such as opening files, choosing color values, and printing documents. The common dialog boxes allow the user to implement a consistent approach to the application's user interface. This reduces the amount of effort that users spend in learning user interface behavior for the application.</p> <p>Although the Common Dialog Box control is known as a standard control, the control doesn't appear on toolbox. These steps have to be done to add the Common Dialog Box control to toolbox:</p> <ol style="list-style-type: none"> 1. Select Project -> Components from the Visual Basic Menu to display the Components dialog box. 2. Scroll to the control named Microsoft Common Dialog Control 6.0 and select it. 3. Click OK. The Common Dialog Box control will now appear at the end of the Toolbox window. 4. Double-click the Common Dialog Box control to add the control to Form window. <p>To display a File Open dialog box</p> <pre> cdbDialog.DialogTitle = "File Open" cdbDialog.ShowOpen 'Trigger the dialog box </pre> <p><i>(1 mark for definition of Common Dialog Box)</i></p>	4

No.	Answers	Marks
(b)	<p>End Sub</p> <p><i>(1 mark for rectifying each error)</i></p> <p>Correct Code:</p> <p>'Program code to display a message on Clicking of cmdClickMe command button</p> <pre>Private Sub cmdClickMe_Click() For i = 1 To Val(txtNumber.Text) MsgBox "I will say Hello at least: " & txtNumber.Text & " times" NEXT i End Sub</pre> <p><i>(1 for identifying the error)</i></p> <p><i>(1 mark for writing the correct code)</i></p>	2
(c)	<p>The error "Division by zero" occurs here as division by zero is not defined ;Here the initial value of i is zero and j when divided by i would give the error.</p> <p>To debug the overflow error, For i = 0 To 5 can be changed to For i = 1 To 5</p> <p><i>(2 marks for correct answer)</i></p>	2
(d)	<p>The Message will be printed one time.</p> <p><i>(2 marks for the correct answer)</i></p>	
(e)	<p>The option Components of Project menu displays the Components dialog box from which we can add controls, designs, insertable objects. The option References refers to Visual ActiveX controls that can be added to the project. It allows us to select another application's objects that we want in our code by setting a reference to the application's object library.</p> <p><i>(2 marks for the correct difference)</i></p>	2
Section - C		
(a)	<p>A variable is a symbol or a name that stands for a value that can change .Variable scope means the part of the program where the variable is accessible or can be manipulated.</p> <pre><<OUTER>> DECLARE A NUMBER(2); BEGIN</pre>	2

No.	Answers	Marks
	<pre> A:=2; DBMS_OUTPUT.PUT_LINE (A); <<INNER>> DECLARE B VARCHAR2(10); BEGIN B := 'Class'; DBMS_OUTPUT.PUT_LINE (A); DBMS_OUTPUT.PUT_LINE (B); END; DBMS_OUTPUT.PUT_LINE (A); DBMS_OUTPUT.PUT_LINE (B); END; </pre> <p><i>(½ mark for definition of variable)</i> <i>(½ mark for definition of Scope of variable)</i> <i>(1 mark for example illustrating both)</i></p>	
(b)	<p>Output:</p> <p>6 8 11 15</p> <p><i>(½ mark for each line of correct output)</i></p>	2
(c)	<pre> IF num = 5 K := 2000 ELSIF num > 5 K := 3000; END IF; </pre> <p><i>(½ Mark each for identifying and correcting four errors)</i></p>	2

A and B are both variables here

A and B are both accessible here

A is accessible but B is not accessible here as the scope of B has ended

SEMICOLON REMOVED

No.	Answers	Marks
(d)	<pre>CREATE OR REPLACE Function ODDEVEN (X IN NUMBER) RETURN BOOLEAN IS BEGIN IF MOD (X,2)=0 THEN RETURN TRUE; ELSE RETURN FALSE; END IF; END;</pre> <p><i>(2 marks for correct Function header 1 mark for IF statement 1 mark for Return statement)</i></p>	4
Q7.		
(a)	<p>Cursors are the work areas for SQL SELECT statements to fetch records from a database table and work with each record on record-by-record basis.</p> <p>Triggers supports event based execution of code statements in database. A trigger gets initiated on the events such as Updating of records, Deletion of records etc. and can be triggered on Statement Level or Row Level.</p> <p><i>(2 marks for the correct difference)</i></p>	2
(b)	<pre>CREATE OR REPLACE PROCEDURE EDSAL AS msalary Employee.Salary%TYPE; BEGIN SELECT Salary into mSalary FROM Employee WHERE ID = 1234; IF mSalary < 5000 THEN DBMS_OUTPUT.PUT_LINE ('Salary less than 5000-granting Increment'); UPDATE Employee SET Salary = Salary + salary * 0.15</pre>	4

No.	Answers	Marks
	<pre> WHERE ID = 1234; ELSE DBMS_OUTPUT.PUT_LINE ('Salary more than 5000-NO Increment'); END IF; END; </pre> <p>(1 mark for procedure header)</p> <p>(1 mark for SELECT)</p> <p>(1 mark for UPDATE- SET command)</p> <p>(1 mark for IF statement)</p> <p>(b) CREATE OR REPLACE TRIGGER BefInsert BEFORE INSERT ON Employee FOR EACH ROW BEGIN DBMS_OUTPUT.PUT_LINE('Hello'); END;</p> <p><i>(1 mark for create Trigger)</i></p> <p><i>(1 mark for BEFORE)</i></p> <p><i>(1 mark for FOR EACH ROW)</i></p> <p><i>(½ mark for BEGDU END)</i></p>	