Tshwane University of Technology

We empower people

YEAR: 2014
EXAMINATION: B

NOVEMBER MAIN

SUBJECT NAME:	Development Software IIB / Software Skills IIB
SUBJECT CODE:	DSO23BT /SFW20BT
QUALIFICATION(S):	Software Development/Business Informatics/Technical Applications/Financial Information Systems

PAPER DESCRIPTION: Computer-based **DURATION:** 4 Hrs **PAPER:** Only

SPECIAL REQUIREMENTS

- ☐ **NONE**
- ☐ **NON-PROGRAMMABLE POCKET CALCULATOR**
- ☐ **SCIENTIFIC CALCULATOR**
- ☐ **COMPUTER ANSWER SHEET**
- ☐ **GRAPH PAPER**
- ☐ **DRAWING INSTRUMENTS**

OTHER: The question paper comes along with the database structure booklet

INSTRUCTIONS TO CANDIDATES: Answer all questions

The instructions are on the cover page. Please do not remove the cover page of all DSO23BT/SFW20BT exam papers

TOTAL NUMBER OF PAGES INCLUDING COVER PAGE: 6

TOTAL NUMBER OF ANNEXURES: 0

EXAMINER: Ms TM Dlamin **FULL MARKS:** 100

MODERATOR: SK Mogapi **TOTAL MARKS:** 103

Class List No:

Tshwane University of Technology

We empower people

EXAM

INSTRUCTIONS TO CANDIDATES

1. Type your answer on the document and saved as ***StudNumCT1.sql*** in the relevant folder as instructed by your lecturer
2. **NO CALCULATORS** or **ELECTRONIC DEVICE** may be used
3.
4. **Use Doctor Database** answer all questions
5. Use **%type** for all questions that refers to a table or database
6. **Sign** the declaration before proceeding

November 2014

TOTAL MARKS: 100

FULL MARKS: 103

TIME: 240 MINUTES

PAGES: 4 (Exl. cover)

1ST EXAMINER: T Dlamini

MODERATOR: SK Mogapi

I declare that I am familiar with, and will abide to the Examination rules of Tshwane University of Technology – ***Annexure A***

SIGNATURE

FACULTY: INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT: SOFTWARE ENGINEERING

DSO23BT \SFW20BT

Exam

CH1 - CH 10

LECTURER's NAME:

VENUE :

STUDENT NUMBER:

SURNAME	**INITIALS**	**%**

Question 1 (8)

Write a PL\SQL Block for a subtraction calculator. The first number must be bigger than the second number. If the difference is even add 5 to it. If it is odd multiply it by 10. See the output below.

```
SQL> /
Enter value for number1: 5
Enter value for number2: 3
The final answer is 7

PL/SQL procedure successfully completed.

SQL> /
Enter value for number1: 2
Enter value for number2: 5
The first number must be bigger than the second number

PL/SQL procedure successfully completed.
```

Question 2 (8)

Write a PL\SQL block that will display the hour of the current time and the number of minutes that have gone since mid-night. DO NOT USE SUBSTRING.

```
  12  /
12 hours and 729 minutes have expired since midnight

PL/SQL procedure successfully completed.

SQL> _
```

Question 3 (7)

The hospital decided to amend patient's follow up visits to be only on Fridays. Write a PL\SQL block that displays patient's next follow-up date. The program must prompt the user to input the last visit date.

```
SQL> /
Enter value for last_visit: 3 September 2014
The next follow-up date is Friday    05TH September 2014

PL/SQL procedure successfully completed.

SQL> /
Enter value for last_visit: 15 September 2014
The next follow-up date is Friday    19TH September 2014

PL/SQL procedure successfully completed.

SQL>
```

Question 4 (13)

Create a PL/SQL program that declares an INDEX_BY_TABLE to temporarily store patient history. The program must take all the rows of the patient history table into the INDEX_BY table. Assign values of your own to the second row of the INDEX_BY table, the date of the event must be today. Finally use a loop to display the rows of the INDEX_BY_TABLE. See the sample output below.

```
7911020534084 consulted on 12TH MARCH     1994 for TONSI
7800001100340 consulted on 06TH SEPTEMBER 2014 for ABCD

PL/SQL procedure successfully completed.
```

Question 5 (21)

Create a PL\SQL block that has 2 cursors. The first cursor should contain treatment information from the treatment table. The second cursor should contain patient treatment information it receives treatid from the first cursor. Read the two cursor using BASIC LOOPS and display their information as shown below.

```
 35  /
FM001: FULL MEDICAL
7911020534084 visited on: 05-JAN-00
CSTIN: CONSULTATION
7503305057803 visited on: 13-JAN-00
7609097812034 visited on: 13-JAN-00
8106070534083 visited on: 13-JAN-00
IVN40: INJECTION - VOLTAREN 40
7911020534084 visited on: 05-JAN-00
IVN20: INJECTION - VOLTAREN 20
STRMV: STITCHES REMOVE
7911020534084 visited on: 05-JAN-00
STTHS: STITCHES

PL/SQL procedure successfully completed.
```

Question 6 (46)

The hospital needs a report that details Doctor's appointment with patients. The report must be in a format recommended by the hospital which contains consultation dates between doctors and patients, see output below:

```
The patient id: 81060***34083 Johan De Waardt Consulted last on: 05 of January
2000
Appointment Type: Full Medical
------------------------------------------------------------------------------
The patient id: 81101***67081 Liaan Lewis Consulted last on: 17 of January
2000
Appointment Type: Full Medical
**********************************************
Number of Appointments: 2

PL/SQL procedure successfully completed.
```

6.1 Create a procedure that contains 1 parameter to accept the doctor's ID number including extra 5 parameters that can be used to hold output. The procedure must retrieve all patients that has an appointment with that doctor by a means of a cursor. The output must show patientid, patient's first and last names must be one and only 25 characters of both must be displayed, appointment type and date. Ensure that the date caters for different centuries. Show the first 4 digits and the last 4 digits of patientid and replace the middle 5 with asterisks, do not use hard code asterisks. (21)

6.2 Create a function that contains 1 parameter to accept the doctor's ID number. The function must calculate the total number of appointment(s) that a doctor has with a patient as detailed in the above output. (8)

6.3 Create a procedure that will call or invoke a procedure and a function in question 6.1 And 6.2. The procedure must have 1 input parameter to accept Doctor's Id and 5 output parameters. After the compilation of this procedure, declare bind variables to will hold the output parameter's values. See the above output. (17)

Doctor Database