Class List No:

Tshwane University of Technology

We empower people

INSTRUCTIONS TO CANDIDATES

1. Write all your answers **on the test paper** OR do them on the computer and save them as ***StudNumST.sql*** in the relevant folder as instructed by your lecturer
2. All answers must be in ink, **no pencil** answers will be marked
3. **No additional paper** may be used or handed in, Extra paper is added at the back of the test and must be handed in with the test
4. **NO CALCULATORS** or **ELECTRONIC DEVICE** may be used
5. **Use STUDENT Database** to answer all questions
6. **Sign** the declaration before proceeding

TOTAL: 100

TIME: 240 MINUTES
PAGES: 6 incl. cover

1ST EXAMINER: SK Mogapi

2nd EXAMINER: H Oberholzer

MODERATOR: S Ndlovu

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: SYSTEMS DEVELOPMENT

DSO23BT / SFW20BT

November Main

Chapter 1-10

LECTURER's NAME:

VENUE :

STUDENT NUMBER:

SURNAME	INITIALS	%

QUESTION 1 **[10]**

Write a PL/SQL block that will ask the user to enter a temperature. The block also requests the user to enter the option. Option 1 is to convert a temperature to degree Celsius and option 2 is to convert a temperature to degree Fahrenheit. Once the user has entered his option print a message indicating the new temperature based on the option as indicated below. If the user enters the wrong option a message must be printed that the user has entered a wrong option number and that the program will be terminated. Use the following formulas:

- Celcius = (temperature – 32) * 5/ 9
- Fahrenheit = (temperature * 9/ 5) + 32

```
Enter value for p_temp: 100
Enter value for p_option: 1
Your new temperature is 38 Degrees Celsius
PL/SQL procedure successfully completed.

Enter value for p_temp: 100
Enter value for p_option: 2
Your new temperature is 212 Degrees Fahrenheit
PL/SQL procedure successfully completed.

Enter value for p_temp: 100
Enter value for p_option: 3
Your have entered an invalid option - program will terminate!!!
PL/SQL procedure successfully completed.
```

QUESTION 2 **[15]**

Write a block of code that receives a user's savings amount and calculates the total savings, making use of **compound** interest (interest on interest), at the end of 12 months. Make use of a BASIC loop to calculate the savings. The interest rate is 15%. Display the new savings using an appropriate message.

```
Enter value for p_savings: 100

The interest earn on R100 at the end of 1 month is : R15
The interest earn on R115 at the end of 2 month is : R17.25
The interest earn on R132.25 at the end of 3 month is : R19.84
The interest earn on R152.09 at the end of 4 month is : R22.81
The interest earn on R174.9 at the end of 5 month is : R26.24
The interest earn on R201.14 at the end of 6 month is : R30.17
The interest earn on R231.31 at the end of 7 month is : R34.7
The interest earn on R266.01 at the end of 8 month is : R39.9
The interest earn on R305.91 at the end of 9 month is : R45.89
The interest earn on R351.8 at the end of 10 month is : R52.77
The interest earn on R404.57 at the end of 11 month is : R60.69
The interest earn on R465.26 at the end of 12 month is : R69.79

Your total savings at 12 months will be : R535.05
```

Question 3 **[10]**

Write a PL/SQL anonymous block that will prompt the user to enter a particular staff number and then print the staff information as shown below:

Enter value for staffno: 513700
PRETORIUS, BB (LECTURER) earns R96000 as salary (Bonus inclusive)

Enter value for staffno: 513500
OLIVIER, G (LECTURER) earns R78000 as salary (Bonus inclusive)

Enter value for staffno: 111111
LAZYBOY, VERY (SPECIALIST) earns R84000 as salary (Bonus inclusive)

QUESTION 4 **[22]**

Write a PL/SQL Block to display the employee number, name, pin#, job, salary, commission, total salary and annual salary of any employee prompted by the user using the following format:

- Employee number and name format: 7934-Miller
- The first 2 characters of the employee name(The first letter being capital letter), the 1st and 3rd number of the employee is used as a pin#
- Label the columns as listed in the output below
- The first letter of the employee name must be in capital letter
- Employees not earning a bonus now earns R500 as a bonus allowance
- Total Salary is given as salary plus bonus as applicable

Your output must resemble the one below.

***Enter the Value for Empnumber*: 513500**
Employee 513500-Olivier with pin Ol53
Works as a LECTURER and earns 78000 has a bonus of 500
A total salary of 78500 with an annual salary of 936000

***Enter the Value for Empnumber*: 500121**
Employee 500121-Mhlanga with pin Mh50
Works as a ASSISTANT and earns 24000 has a bonus of 1800
A total salary of 25800 with an annual salary of 288000

QUESTION 5 **[32]**

Create **one** PL/SQL block that uses a substitution variable called P_STNR that is prompted for in SQL*PLUS.

Use a SQL*Plus **predefined** exception to inform the user when the student does not exist. Display the message listed below and stop all processing.

```
SQL> Enter value for p_stnr: 12345678
```

NO STUDENT NUMBER LIKE 12345678

When the student does exist and is registered for some subjects, display the names of the subjects that the student is registered for in the following format:

```
SQL> Enter value for p_stnr: 94000001

Student: SMITH, VM (94000001)
REGISTERED FOR THE FOLLOWING 3 SUBJECTS:
Cost And Management Acc 1
Financial Accounting 1
Internal Auditing 1
```

You must retrieve the subject information as PL/SQL records.

Assume that a student is not allowed to be registered for more than five subjects. Use a **user-defined** exception named E_TOO_MANY to inform the user that the student is registered for more than five subjects in the following format:

```
SQL> Enter value for p_stnr: 96445566

Student: SLEEPY, AL (96445566)
TOO MANY SUBJECTS REGISTERED: 6
```

Use a **user-defined** exception E_NO_SUBJECTS to inform the user when a student is not registered for any subjects.

```
SQL> Enter value for p_stnr: 94535353

Student: SMITH, TF (94535353)
NO SUBJECTS REGISTERED
```

QUESTION 6 **[8]**

Write a stored function named CALC_AVG. This function must receive a student number and then calculate the average of the final marks of this student. This average must then be sent back to the calling environment. For example: The final marks of the subjects of student 97003455 are as listed below:

```
SQL> SELECT studnr, subj_code, finalmark
  2  FROM s_registration
  3  WHERE studnr = 97003455;

    STUDNR SUB  FINALMARK
---------- --- ----------
  97003455 IS1          56
  97003455 DS1          84
  97003455 TP1          51
  97003455 QT1          51
  97003455 FS1          59
  97003455 SP1          84

6 rows selected.
```

The average must be **calculated and sent back** by this function as 64.2

QUESTION 7 **[10]**

Declare one host variable named G_RESULT that can be used to print the following result, for example:

The average of JH Smuts is 64.2%

Write a stored procedure named FINAL_AVG_PROC. This procedure must recieve a student number. The procedure must then call the function that was written in the previous question to calculate the average of the student's final marks and print the result as listed above.

Run the procedure and print the result.

Remember to copy the call to the procedure and the printing of the results.

STUDENT DATABASE