# Exercise 1 (1%)

A screenshot of a computer

Description automatically generated

Code:

DECLARE

SILVER\_AMOUNT CONSTANT NUMBER(10,0) := 2000;

GOLD\_AMOUNT CONSTANT NUMBER(10,0) := 3500;

-- Declaring the variable total\_purchase

total\_purchase NUMBER(10,0);

customer\_rating VARCHAR2(10);

BEGIN

-- Prompting user to enter total purchase

total\_purchase := :ENTER\_TOTAL\_PURCHASE;

IF total\_purchase > GOLD\_AMOUNT THEN

customer\_rating := 'Gold';

ELSIF total\_purchase > SILVER\_AMOUNT THEN

customer\_rating := 'Silver';

ELSE

customer\_rating := 'Bronze';

END IF;

-- Displaying the determined customer rating according to the amount

DBMS\_OUTPUT.PUT\_LINE('The customer is a ' || customer\_rating || ' member.');

END;

# Exercise 2 (1.5%)

A screenshot of a computer

Description automatically generated

DECLARE

SILVER\_AMOUNT CONSTANT NUMBER(10,0) := 2000;

GOLD\_AMOUNT CONSTANT NUMBER(10,0) := 3500;

-- Declaring the variable total\_purchase

total\_purchase NUMBER(10,0);

customer\_rating VARCHAR2(10);

BEGIN

-- Prompting user to enter total purchase

total\_purchase := :ENTER\_TOTAL\_PURCHASE;

IF total\_purchase > GOLD\_AMOUNT THEN

customer\_rating := 'Gold';

ELSIF total\_purchase > SILVER\_AMOUNT THEN

customer\_rating := 'Silver';

ELSE

customer\_rating := 'Bronze';

END IF;

-- Displaying the determined customer rating according to the amount

DBMS\_OUTPUT.PUT\_LINE('The customer is a ' || customer\_rating || ' member.');

END;

DECLARE -- Declaring the variables and prompt to enter to the user

start\_nn NUMBER(2) := :ENTER\_START\_NN;

end\_nn NUMBER(2) := :ENTER\_END\_NN;

start\_mmm NUMBER(3) := :ENTER\_START\_MMM;

end\_mmm NUMBER(3) := :ENTER\_END\_MMM;

license\_plate\_number VARCHAR2(7);

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Available License Plates are :'); -- Printing and displaying the output

FOR nn IN start\_nn..end\_nn LOOP

FOR mmm IN start\_mmm..end\_mmm LOOP

license\_plate\_number := ( nn || '-' || LPAD(mmm, 3, '0') );

DBMS\_OUTPUT.PUT\_LINE(license\_plate\_number);

END LOOP;

END LOOP;

END;

# Exercise 3 (1.5%):

A screenshot of a computer

Description automatically generated

DECLARE --Declaring the variables

Loan\_amount NUMBER(7,2) := :ENTER\_LOAN\_AMOUNT;

Loan\_payment NUMBER(5,2) := :ENTER\_LOAN\_PAYMENT;

Equal\_payment NUMBER(3,0);

outstanding\_balance NUMBER(7,2);

--calculating equal payments

BEGIN

Equal\_payment := FLOOR(loan\_amount / loan\_payment);

-- Displaying loan amounts

DBMS\_OUTPUT.PUT\_LINE('Loan Amount: ' || TO\_CHAR(loan\_amount, '$99999.00'));

DBMS\_OUTPUT.PUT\_LINE('Loan Payment: ' || TO\_CHAR(loan\_payment, '$99999.00'));

DBMS\_OUTPUT.PUT\_LINE('Equal Payment: ' || equal\_payment);

DBMS\_OUTPUT.PUT\_LINE('Payment# Balance');

DBMS\_OUTPUT.PUT\_LINE('-------- --------');

-- Initializing outstanding balance with the loan amount

outstanding\_balance := Loan\_amount;

--Using loop through payments

FOR i IN 1..equal\_payment LOOP

outstanding\_balance := outstanding\_balance - Loan\_payment;

DBMS\_OUTPUT.PUT\_LINE(' ' || i || ' ' || TO\_CHAR(outstanding\_balance, '$99999.00'));

END LOOP;

-- Displaying final outstanding balance

DBMS\_OUTPUT.PUT\_LINE('Outstanding balance: ' || TO\_CHAR(outstanding\_balance, '$99999.00'));

END;