Solution

Scenario: 1

```
CREATE OR REPLACE FUNCTION CalculateAge( cust_dob IN DATE)
Return number
IS
cust_age NUMBER;
Begin
cust_age:=(SYSDATE-cust_dob)/365;
RETURN cust_age;
END;
/

SET SERVEROUT ON;
DECLARE
        age NUMBER;
    BEGIN
      age := CalculateAge(TO_DATE('2002-08-26', 'YYYY-MM-DD'));
      DBMS_OUTPUT.PUT_LINE('Age of Customer:  ' || TRUNC(age));
   END;
  /
```

Scenario: 2

```
CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment (
  loan_amount NUMBER,
  interest_rate NUMBER,
  loan_duration NUMBER
)
RETURN NUMBER
IS
  monthly_interest NUMBER;
  monthly_install NUMBER;
BEGIN
  monthly_interest := interest_rate / 1200;
  monthly_install :=
(loan_amount*monthly_interest*POWER(1+monthly_interest,loan_duration
*12)) / (POWER(1+monthly_interest,loan_duration * 12) - 1);
  RETURN monthly_install;
END;
/
SET SERVEROUTPUT ON;
DECLARE
       res NUMBER;
  BEGIN
         res := CalculateMonthlyInstallment(100,4,2);
         DBMS_OUTPUT.PUT_LINE(TRUNC(res));
     END;
     /
```

Scenario: 3

```
CREATE OR REPLACE FUNCTION HasSufficientBalance (
   account_id IN NUMBER,
   amount IN NUMBER
)
RETURN BOOLEAN
IS
    f_Balance NUMBER;
BEGIN
     SELECT balance INTO f_balance FROM ACCOUNTS where
accountid=account_id;
      IF f_balance >= amount THEN
     RETURN TRUE;
       ELSE
     RETURN FALSE;
       END IF;
END;
/
SET SERVEROUTPUT ON;
DECLARE
        ans BOOLEAN;
    BEGIN
     ans := HasSufficientBalance(1, 1000);
        IF ans THEN
            DBMS_OUTPUT.PUT_LINE('TRUE');
        ELSE
            DBMS_OUTPUT.PUT_LINE('FALSE');
       END IF;
   END;
   /
```