NOTE: All the programs are executed on
apex.oracle.com.

1.

```
DECLARE
  first NUMBER(5);
  second NUMBER(5);
  result NUMBER(5);


  PROCEDURE addition (first IN NUMBER, second IN NUMBER, result OUT
  NUMBER) IS BEGIN
     result:=first+second;
  END;

BEGIN
first:=:first; second:=:second; addition(first,second,result);
dbms_output.put_line('The addition of the two numbers is ' || result);

END; /
```

Output

```
The addition of the two numbers is 11
```

2.

```
DECLARE
  ID NUMBER(5);
  NAME
  VARCHAR(30);

  PROCEDURE member_name_search(ID IN
  NUMBER) IS BEGIN
SELECT MEMBER_NAME INTO NAME FROM MEMBER WHERE MEMBER_ID=ID;
dbms_output.put_line('The Member Name corresponding to Member Id. ' || ID || ' is ' ||
NAME); END;

BEGIN
ID:=:ID;
```

```
member_name_search(ID);
END; /
```

Output

The Member Name corresponding to Member Id. 2 is Abhirup Sarkar

3.

```
DECLARE
  details BOOKS%ROWTYPE; CURSOR
  books_cursor IS (SELECT * FROM BOOKS);

  PROCEDURE
  display_book IS BEGIN
    OPEN books_cursor; dbms_output.put_line('BID BNAME AUTHOR NAME
    COST CATEGORY');

dbms_output.put_line('_____
____') ;
    LOOP
      FETCH books_cursor INTO details;
      EXIT WHEN
      books_cursor%NOTFOUND; IF
      details.Cost>500 THEN
          dbms_output.put_line(details.Book_No || ' ' || Rpad(details.Book_Name,20,' ') || ' '
|| Rpad(details.Author_Name,15,' ') || ' ' || Rpad(details.Cost,10,' ') || ' ' ||
Rpad(details.Category,20,' '));
      END IF; END
      LOOP; END;

BEGIN
    display_book(

); END; /
```

OUTPUT

BID BNAME AUTHOR NAME COST CATEGORY

PL SQL-Ref Scott Urman 750 Database 106 UNIX Sumitava Das 850 System 107
Optics Ghatak 600 Science

4.

```
DECLARE
  BID NUMBER(5);

  PROCEDURE update_cost(BID IN
  NUMBER) IS BEGIN
  UPDATE BOOKS_COPY SET Cost=Cost+50 WHERE
  Book_No=BID; dbms_output.put_line('Row updated'); END;

BEGIN
BID:=:BID;
update_cost(BID);
END; /
```

OUTPUT

Row update

5.

```
DECLARE
  first NUMBER(5);
  second NUMBER(5);
  result NUMBER(10);


  PROCEDURE multiply(first IN NUMBER, second IN NUMBER, result OUT
  NUMBER) IS BEGIN
      result:=first*second;
  END;

BEGIN
  first:=:first;
  second:=:second;
  multiply(first,second,result);

      dbms_output.put_line('The result after multiplication of ' || first || ' with ' || second || ' is '
```

|| result || '.'); END; / OUTPUT

The result after multiplication of 5 with 7 is 35.

6. CREATE TRIGGER

member_error BEFORE
DELETE ON MEMBER_COPY
FOR EACH ROW BEGIN
RAISE_APPLICATION_ERROR(-20000,'Applicati

on Error'); END; / 7. CREATE TRIGGER books_error
BEFORE DELETE ON BOOKS_COPY FOR EACH
ROW WHEN (OLD.Cost<200) BEGIN
RAISE_APPLICATION_ERROR(-20000,'Applicati

on Error'); END; / 8. DECLARE id NUMBER(5); name
VARCHAR(30); BEGIN
id:=:id; SELECT MEMBER_NAME INTO name FROM MEMBER_COPY WHERE
MEMBER_ID=id; dbms_output.put_line('The name of the member corresponding to ID: ' ||
id || ' is ' || name); EXCEPTION
WHEN NO_DATA_FOUND
THEN
RAISE_APPLICATION_ERROR(-20403,'No such

data found'); END; / OUTPUT

ORA-20403: No such data found

9. CREATE TABLE

CHANGE(BOOK_ID
NUMBER(5),
CHANGE_DATE DATE,
CHANGE_TYPE
VARCHAR(10));

-- CREATE TRIGGER

```
update_changes BEFORE
UPDATE OR DELETE ON
BOOKS_COPY FOR EACH
ROW BEGIN
  IF UPDATING
  THEN
     INSERT INTO CHANGE VALUES
  (:OLD.BOOK_NO,SYSDATE,'UPDATE'); ELSE
     INSERT INTO CHANGE VALUES
(:OLD.BOOK_NO,SYSDATE,'DELETE'); END IF; END; /
```

10.

```
CREATE TRIGGER
issue_exception BEFORE
INSERT ON ISSUE_COPY FOR
EACH ROW DECLARE
Day VARCHAR(10); MyException EXCEPTION;
PRAGMA
EXCEPTION_INIT(MyException,-20001); BEGIN
    DAY:=TO_CHAR(:NEW.ISSUE_DATE,'DA
  Y'); IF (DAY='WEDNESDAY' OR
  DAY='SUNDAY') THEN
     RAISE
MyException; END IF;
END; /
```

----------X----------