DECLARE

v\_num1 NUMBER := 5;

v\_num2 NUMBER := 3;

v\_temp NUMBER;

BEGIN

-- if v\_num1 is greater than v\_num2 rearrange their values

IF v\_num1 > v\_num2 THEN

v\_temp := v\_num1;

v\_num1 := v\_num2;

v\_num2 := v\_temp;

END IF;

-- display the values of v\_num1 and v\_num2

DBMS\_OUTPUT.PUT\_LINE ('v\_num1 = '||v\_num1);

DBMS\_OUTPUT.PUT\_LINE ('v\_num2 = '||v\_num2);

END;

**v\_num1 = 3**

**v\_num2 = 5**

**PL/SQL procedure successfully completed.**

DECLARE

v\_num NUMBER := &sv\_user\_num;

BEGIN

-- test if the number provided by the user is even

IF MOD(v\_num,2) = 0 THEN

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is even number');

ELSE

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is odd number');

END IF;

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

END;

**Enter value for v\_user\_num: 24**

**old 2: v\_num NUMBER := &v\_user\_num;**

**new 2: v\_num NUMBER := 24;**

**24 is even number**

**Done**

**PL/SQL procedure successfully completed.**

DECLARE

v\_num1 NUMBER := 0;

v\_num2 NUMBER;

BEGIN

IF v\_num1 = v\_num2 THEN

DBMS\_OUTPUT.PUT\_LINE ('v\_num1 = v\_num2');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('v\_num1 != v\_num2');

END IF;

END;

**v\_num1 != v\_num2**

**PL/SQL procedure successfully completed.**

-- ch04\_1a.sql, version 1.0

SET SERVEROUTPUT ON

DECLARE

v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');

v\_day VARCHAR2(15);

BEGIN

v\_day := RTRIM(TO\_CHAR(v\_date, 'DAY'));

IF v\_day IN ('SATURDAY', 'SUNDAY') THEN

DBMS\_OUTPUT.PUT\_LINE (v\_date||' falls on weekend');

END IF;

-- control resumes here

DBMS\_OUTPUT.PUT\_LINE ('Done…');

END;

**Enter value for sv\_user\_date: 09-JAN-2007**

**old 2: v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');**

**new 2: v\_date DATE := TO\_DATE('09-JAN-2007', 'DD-MON-YYYY');**

**Done...**

**PL/SQL procedure successfully completed.**

**Enter value for sv\_user\_date: 13-JAN-2007**

**old 2: v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');**

**new 2: v\_date DATE := TO\_DATE('13-JAN-2007', 'DD-MON-YYYY');**

**13-JAN-07 falls on weekend**

**Done...**

**PL/SQL procedure successfully completed.**

-- ch04\_1b.sql, version 2.0

SET SERVEROUTPUT ON

DECLARE

v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');

v\_day VARCHAR2(15);

BEGIN

**v\_day := TO\_CHAR(v\_date, 'DAY');**

IF v\_day IN ('SATURDAY', 'SUNDAY') THEN

DBMS\_OUTPUT.PUT\_LINE (v\_date||' falls on weekend');

END IF;

--- control resumes here

DBMS\_OUTPUT.PUT\_LINE ('Done…');

END;

**Enter value for sv\_user\_date: 14-JAN-2007**

**old 2: v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');**

**new 2: v\_date DATE := TO\_DATE('14-JAN-2007', 'DD-MON-YYYY');**

**Done...**

**PL/SQL procedure successfully completed.**

-- ch04\_1c.sql, version 3.0

SET SERVEROUTPUT ON

DECLARE

v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');

v\_day VARCHAR2(15);

BEGIN

v\_day := RTRIM(TO\_CHAR(v\_date, 'DAY'));

**IF v\_day LIKE 'S%' THEN**

DBMS\_OUTPUT.PUT\_LINE (v\_date||' falls on weekend');

END IF;

-- control resumes here

DBMS\_OUTPUT.PUT\_LINE ('Done…');

END;

-- ch04\_1d.sql, version 4.0

SET SERVEROUTPUT ON

DECLARE

v\_date DATE := TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');

v\_day VARCHAR2(15);

BEGIN

v\_day := RTRIM(TO\_CHAR(v\_date, 'DAY'));

IF v\_day IN ('SATURDAY', 'SUNDAY') THEN

DBMS\_OUTPUT.PUT\_LINE (v\_date||' falls on weekend');

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE (v\_date||' does not fall on the weekend');**

END IF;

-- control resumes here

DBMS\_OUTPUT.PUT\_LINE('Done…');

END;

-- ch04\_2a.sql, version 1.0

SET SERVEROUTPUT ON

DECLARE

v\_total NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_total

FROM enrollment e

JOIN section s USING (section\_id)

WHERE s.course\_no = 25

AND s.section\_no = 1;

-- check if section 1 of course 25 is full

IF v\_total >= 15 THEN

DBMS\_OUTPUT.PUT\_LINE ('Section 1 of course 25 is full');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('Section 1 of course 25 is not full');

END IF;

-- control resumes here

END;

-- ch04\_2b.sql, version 2.0

SET SERVEROUTPUT ON

DECLARE

v\_total NUMBER;

**v\_course\_no CHAR(6) := '&sv\_course\_no';**

**v\_section\_no NUMBER := &sv\_section\_no;**

BEGIN

**SELECT COUNT(\*)**

**INTO v\_total**

**FROM enrollment e**

**JOIN section s USING (section\_id)**

**WHERE s.course\_no = v\_course\_no**

**AND s.section\_no = v\_section\_no;**

-- check if a specific section of a course is full

IF v\_total >= 15 THEN

DBMS\_OUTPUT.PUT\_LINE ('Section 1 of course 25 is full');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('Section 1 of course 25 is not full');

END IF;

-- control resumes here

END;

-- ch04\_2c.sql, version 3.0

SET SERVEROUTPUT ON

DECLARE

v\_total NUMBER;

v\_students NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_total

FROM enrollment e

JOIN section s USING (section\_id)

WHERE s.course\_no = 25

AND s.section\_no = 1;

-- check if section 1 of course 25 is full

IF v\_total >= 15 THEN

DBMS\_OUTPUT.PUT\_LINE ('Section 1 of course 25 is full');

ELSE

**v\_students := 15 - v\_total;**

**DBMS\_OUTPUT.PUT\_LINE (v\_students||' students can still enroll'||**

**' into section 1 of course 25');**

END IF;

-- control resumes here

END;

DECLARE

v\_num NUMBER := &sv\_num;

BEGIN

IF v\_num < 0 THEN

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is a negative number');

ELSIF v\_num = 0 THEN

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is equal to zero');

ELSE

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is a positive number');

END IF;

END;

**Enter value for sv\_num: 5**

**old 2: v\_num NUMBER := &sv\_num;**

**new 2: v\_num NUMBER := 5;**

**5 is a positive number**

**PL/SQL procedure successfully completed.**

DECLARE

v\_num NUMBER := &sv\_num;

BEGIN

IF v\_num < 0 THEN

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is a negative number');

ELSIF v\_num > 0 THEN

DBMS\_OUTPUT.PUT\_LINE (v\_num||' is a positive number');

END IF;

DBMS\_OUTPUT.PUT\_LINE ('Done…');

END;

**Enter value for sv\_num: 0**

**old 2: v\_num NUMBER := &sv\_num;**

**new 2: v\_num NUMBER := 0;**

**Done…**

**PL/SQL procedure successfully completed.**

-- ch04\_3a.sql, version 1.0

SET SERVEROUTPUT ON

DECLARE

v\_student\_id NUMBER := 102;

v\_section\_id NUMBER := 89;

v\_final\_grade NUMBER;

v\_letter\_grade CHAR(1);

BEGIN

SELECT final\_grade

INTO v\_final\_grade

FROM enrollment

WHERE student\_id = v\_student\_id

AND section\_id = v\_section\_id;

IF v\_final\_grade BETWEEN 90 AND 100 THEN

v\_letter\_grade := 'A';

ELSIF v\_final\_grade BETWEEN 80 AND 89 THEN

v\_letter\_grade := 'B';

ELSIF v\_final\_grade BETWEEN 70 AND 79 THEN

v\_letter\_grade := 'C';

ELSIF v\_final\_grade BETWEEN 60 AND 69 THEN

v\_letter\_grade := 'D';

ELSE

v\_letter\_grade := 'F';

END IF;

-- control resumes here

DBMS\_OUTPUT.PUT\_LINE ('Letter grade is: '||  
 v\_letter\_grade);

END;

-- ch04\_3b.sql, version 2.0

SET SERVEROUTPUT ON

DECLARE

v\_student\_id NUMBER := 102;

v\_section\_id NUMBER := 89;

v\_final\_grade NUMBER;

v\_letter\_grade CHAR(1);

BEGIN

SELECT final\_grade

INTO v\_final\_grade

FROM enrollment

WHERE student\_id = v\_student\_id

AND section\_id = v\_section\_id;

**IF v\_final\_grade IS NULL THEN**

**DBMS\_OUTPUT.PUT\_LINE('v\_final\_grade is null');**

**ELSIF v\_final\_grade BETWEEN 90 AND 100 THEN**

v\_letter\_grade := 'A';

ELSIF v\_final\_grade BETWEEN 80 AND 89 THEN

v\_letter\_grade := 'B';

ELSIF v\_final\_grade BETWEEN 70 AND 79 THEN

v\_letter\_grade := 'C';

ELSIF v\_final\_grade BETWEEN 60 AND 69 THEN

v\_letter\_grade := 'D';

ELSE

v\_letter\_grade := 'F';

END IF;

-- control resumes here

‑DBMS\_OUTPUT.PUT\_LINE ('Letter grade is: '||  
 v\_letter\_grade);

END;

-- ch04\_3c.sql, version 3.0

SET SERVEROUTPUT ON

DECLARE

**v\_student\_id NUMBER := &sv\_student\_id;**

**v\_section\_id NUMBER := &sv\_section\_id;**

v\_final\_grade NUMBER;

v\_letter\_grade CHAR(1);

BEGIN

SELECT final\_grade

INTO v\_final\_grade

FROM enrollment

WHERE student\_id = v\_student\_id

AND section\_id = v\_section\_id;

IF v\_final\_grade BETWEEN 90 AND 100 THEN

v\_letter\_grade := 'A';

ELSIF v\_final\_grade BETWEEN 80 AND 89 THEN

v\_letter\_grade := 'B';

ELSIF v\_final\_grade BETWEEN 70 AND 79 THEN

v\_letter\_grade := 'C';

ELSIF v\_final\_grade BETWEEN 60 AND 69 THEN

v\_letter\_grade := 'D';

ELSE

v\_letter\_grade := 'F';

END IF;

-- control resumes here

DBMS\_OUTPUT.PUT\_LINE ('Letter grade is: '||  
 v\_letter\_grade);

END;

-- ch04\_3d.sql, version 4.0

SET SERVEROUTPUT ON

DECLARE

v\_student\_id NUMBER := 102;

v\_section\_id NUMBER := 89;

v\_final\_grade NUMBER;

v\_letter\_grade CHAR(1);

BEGIN

SELECT final\_grade

INTO v\_final\_grade

FROM enrollment

WHERE student\_id = v\_student\_id

AND section\_id = v\_section\_id;

**IF v\_final\_grade >= 90 THEN**

v\_letter\_grade := 'A';

**ELSIF v\_final\_grade >= 80 THEN**

v\_letter\_grade := 'B';

**ELSIF v\_final\_grade >= 70 THEN**

v\_letter\_grade := 'C';

**ELSIF v\_final\_grade >= 60 THEN**

v\_letter\_grade := 'D';

ELSE

v\_letter\_grade := 'F';

END IF;

--- control resumes here

‑DBMS\_OUTPUT.PUT\_LINE ('Letter grade is: '||  
 v\_letter\_grade);

END;

DECLARE

v\_num1 NUMBER := &sv\_num1;

v\_num2 NUMBER := &sv\_num2;

v\_total NUMBER;

BEGIN

IF v\_num1 > v\_num2 THEN

DBMS\_OUTPUT.PUT\_LINE ('IF part of the outer IF');

v\_total := v\_num1 - v\_num2;

ELSE

DBMS\_OUTPUT.PUT\_LINE ('ELSE part of the outer IF');

v\_total := v\_num1 + v\_num2;

**IF v\_total < 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE ('Inner IF');**

**v\_total := v\_total \* (-1);**

**END IF;**

END IF;

DBMS\_OUTPUT.PUT\_LINE ('v\_total = '||v\_total);

END;

**Enter value for sv\_num1: -4**

**old 2: v\_num1 NUMBER := &sv\_num1;**

**new 2: v\_num1 NUMBER := -4;**

**Enter value for sv\_num2: 3**

**old 3: v\_num2 NUMBER := &sv\_num2;**

**new 3: v\_num2 NUMBER := 3;**

**ELSE part of the outer IF**

**Inner IF**

**v\_total = 1**

**PL/SQL procedure successfully completed.**

DECLARE

v\_letter CHAR(1) := '&sv\_letter';

BEGIN

IF (v\_letter >= 'A' AND v\_letter <= 'Z') OR

(v\_letter >= 'a' AND v\_letter <= 'z')

THEN

DBMS\_OUTPUT.PUT\_LINE ('This is a letter');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('This is not a letter');

IF v\_letter BETWEEN '0' and '9' THEN

DBMS\_OUTPUT.PUT\_LINE ('This is a number');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('This is not a number');

END IF;

END IF;

END;

**Enter value for sv\_letter: ?**

**old 2: v\_letter CHAR(1) := '&sv\_letter';**

**new 2: v\_letter CHAR(1) := '?';**

**This is not a letter**

**This is not a number**

**PL/SQL procedure successfully completed.**

-- ch04\_4a.sql, version 1.0

SET SERVEROUTPUT ON

DECLARE

v\_temp\_in NUMBER := &sv\_temp\_in;

v\_scale\_in CHAR := '&sv\_scale\_in';

v\_temp\_out NUMBER;

v\_scale\_out CHAR;

BEGIN

IF v\_scale\_in != 'C' AND v\_scale\_in != 'F' THEN

DBMS\_OUTPUT.PUT\_LINE ('This is not a valid scale');

ELSE

IF v\_scale\_in = 'C' THEN

v\_temp\_out := ( (9 \* v\_temp\_in) / 5 ) + 32;

v\_scale\_out := 'F';

ELSE

v\_temp\_out := ( (v\_temp\_in – 32) \* 5 ) / 9;

v\_scale\_out := 'C';

END IF;

DBMS\_OUTPUT.PUT\_LINE ('New scale is: '||v\_scale\_out);

DBMS\_OUTPUT.PUT\_LINE ('New temperature is: '||v\_temp\_out);

END IF;

END;

**Enter value for sv\_temp\_in: 100**

**old 2: v\_temp\_in NUMBER := &sv\_temp\_in;**

**new 2: v\_temp\_in NUMBER := 100;**

**Enter value for sv\_scale\_in: C**

**old 3: v\_scale\_in CHAR := '&sv\_scale\_in';**

**new 3: v\_scale\_in CHAR := 'C';**

**New scale is: F**

**New temperature is: 212**

**PL/SQL procedure successfully completed.**

**Enter value for sv\_temp\_in:**

**old 2: v\_temp\_in NUMBER := &sv\_temp\_in;**

**new 2: v\_temp\_in NUMBER := ;**

**Enter value for sv\_scale\_in: C**

**old 3: v\_scale\_in CHAR := '&sv\_scale\_in';**

**new 3: v\_scale\_in CHAR := 'C';**

**v\_temp\_in NUMBER := ;**

**\***

**ERROR at line 2:**

**ORA-06550: line 2, column 27:**

**‑PLS-00103: Encountered the symbol ";" when expecting one of the following:**

**( - + mod not null <an identifier>**

**<a double-quoted delimited-identifier> <a bind variable> avg**

**count current exists max min prior sql stddev sum variance**

**cast <a string literal with character set specification>**

**<a number> <a single-quoted SQL string>**

**The symbol "null" was substituted for ";" to continue.**

**Enter value for sv\_temp\_in: 45**

**old 2: v\_temp\_in NUMBER := &sv\_temp\_in;**

**new 2: v\_temp\_in NUMBER := 45;**

**Enter value for sv\_scale\_in: V**

**old 3: v\_scale\_in CHAR := '&sv\_scale\_in';**

**new 3: v\_scale\_in CHAR := 'V';**

**This is not a valid scale**

**PL/SQL procedure successfully completed.**

-- ch04\_4b.sql, version 2.0

DECLARE

v\_temp\_in NUMBER := &sv\_temp\_in;

v\_scale\_in CHAR := '&sv\_scale\_in';

v\_temp\_out NUMBER;

v\_scale\_out CHAR;

BEGIN

IF v\_scale\_in != 'C' AND v\_scale\_in != 'F' THEN

DBMS\_OUTPUT.PUT\_LINE ('This is not a valid scale');

**v\_temp\_out := 0;**

**v\_scale\_out := 'C';**

ELSE

IF v\_scale\_in = 'C' THEN

v\_temp\_out := ( (9 \* v\_temp\_in) / 5 ) + 32;

v\_scale\_out := 'F';

ELSE

v\_temp\_out := ( (v\_temp\_in - 32) \* 5 ) / 9;

v\_scale\_out := 'C';

END IF;

END IF;

**DBMS\_OUTPUT.PUT\_LINE ('New scale is: '||v\_scale\_out);**

**DBMS\_OUTPUT.PUT\_LINE ('New temperature is: '||v\_temp\_out);**

END;

**Enter value for sv\_temp\_in: 100**

**old 2: v\_temp\_in NUMBER := &sv\_temp\_in;**

**new 2: v\_temp\_in NUMBER := 100;**

**Enter value for sv\_scale\_in: V**

**old 3: v\_scale\_in CHAR := '&sv\_scale\_in';**

**new 3: v\_scale\_in CHAR := 'V';**

**This is not a valid scale.**

**New scale is: C**

**New temperature is: 0**

**PL/SQL procedure successfully completed.**