D.Y. Patil Academic Education Excellence Federation's
Dr.D.Y. Patil Technical Campus
(Engineering and MCA)
(Approved by AICTE, DTE-Govt of Maharashtra, Affiliated to Savitribai Phule Pune University, Pune)
Sr.No.32/1/A/7, Near Talegaon Railway Station, A/P Varale, Talegaon Dabhade
Tal-Maval, Dist-Pune 410507
Ph. No.9920141406,9309228311,7666829653,9307909501
Website: www.dypatiltcs.com

Lab Manual – Unnamed PL/SQL Code Block using Control Structures and Exception Handling

Experiment No. 4

Title:

Unnamed PL/SQL Code Block using Control Structures and Exception Handling

---

Objectives:

- To understand the concept of unnamed PL/SQL blocks.
- To demonstrate the use of control structures (IF-ELSE, LOOP).
- To implement exception handling (predefined and user-defined).
- To calculate and insert fine details for book borrowers in a library system.

Problem Statement:

Design and implement a PL/SQL unnamed block for a library management system with the following specifications:

- Accept Roll_no and NameofBook from the user.
- Calculate the number of days between the issue date and return date.
- Fine calculation rules:
  - If days between **15–30** → Fine = Rs. 5/day.
  - If days > 30 → Fine = Rs. 50/day.
  - If days < 15 → No fine.
- Update the Borrower table by changing status from 'I' (Issued) to 'R' (Returned).
- If fine is applicable, insert details into the Fine table.
- Handle exceptions using **named/user-defined exception handling**.

Software and Hardware Requirements:

- **Software:** Oracle Database 10g/11g/12c or higher, SQL*Plus / Oracle SQL Developer
- **Hardware:** Intel i3/i5 Processor, 4GB+ RAM, 500GB HDD, Windows/Linux OS

Theory – Concept in Brief:

- **PL/SQL Unnamed Block:** A block that does not have a name, written directly in SQL*Plus.
- **Control Structures:**
  - Conditional: IF…ELSIF…ELSE
  - Iterative: LOOP, WHILE, FOR
- **Exception Handling:** Used to handle runtime errors gracefully using EXCEPTION block.
- **Tables Used:**
  - Borrower(Roll_no, Name, DateofIssue, NameofBook, Status)
  - Fine(Roll_no, Date, Amt)

Algorithm:

1. Start.
2. Accept Roll_no and NameofBook.
3. Retrieve DateofIssue from Borrower table.
4. Calculate No_of_days = sysdate - DateofIssue.
5. IF days < 15 → Fine = 0.
6. ELSE IF days BETWEEN 15 AND 30 → Fine = days * 5.
7. ELSE IF days > 30 → Fine = days * 50.
8. Update Borrower status from 'I' to 'R'.
9. If fine > 0 → Insert record into Fine table.
10. Handle exceptions (e.g., invalid Roll_no, no matching book).

11. End.

Flowchart:

PL/SQL Code (Unnamed Block):

```
DECLARE
  v_roll   Borrower.Roll_no%TYPE;
  v_book   Borrower.NameofBook%TYPE;
  v_issue  Borrower.DateofIssue%TYPE;
  v_days   NUMBER;
  v_fine   NUMBER := 0;

  e_no_record EXCEPTION;

BEGIN
  -- Accept input
  v_roll := &Roll_no;
  v_book := '&BookName';

  -- Fetch Issue Date
  SELECT DateofIssue
  INTO v_issue
  FROM Borrower
  WHERE Roll_no = v_roll AND NameofBook = v_book AND Status = 'I';

  v_days := TRUNC(SYSDATE - v_issue);

  -- Fine Calculation
  IF v_days < 15 THEN
    v_fine := 0;
  ELSIF v_days BETWEEN 15 AND 30 THEN
    v_fine := v_days * 5;
  ELSE
    v_fine := v_days * 50;
  END IF;
```

```
  -- Update Status
  UPDATE Borrower
  SET Status = 'R'
  WHERE Roll_no = v_roll AND NameofBook = v_book;

  -- Insert Fine if applicable
  IF v_fine > 0 THEN
    INSERT INTO Fine(Roll_no, Date, Amt)
    VALUES (v_roll, SYSDATE, v_fine);
  END IF;

  DBMS_OUTPUT.PUT_LINE('Book Returned. Fine = ' || v_fine);

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Error: Record not found or already returned.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Unexpected Error: ' || SQLERRM);
END;
/
```

Test Cases:

Roll_no	BookName	DateofIssue	Current Date	Days	Expected Fine	Expected Status
101	DBMS	01-Aug-25	10-Aug-25	9	0	R
102	SQL	01-Jul-25	20-Jul-25	19	95	R
103	OS	01-Jun-25	10-Jul-25	39	1950	R

Test Data Set:

```
INSERT INTO Borrower VALUES (101, 'Amit', '01-AUG-2025', 'DBMS', 'I');

INSERT INTO Borrower VALUES (102, 'Priya', '01-JUL-2025', 'SQL', 'I');

INSERT INTO Borrower VALUES (103, 'Rahul', '01-JUN-2025', 'OS', 'I');
```

Conclusion / Analysis:

- Successfully implemented unnamed PL/SQL block with **control structures** and **exception handling**.
- Fine calculation works as per given rules.
- Borrower status updates correctly from 'I' to 'R'.
- Exception handling ensures robustness against missing/invalid data.
