- **Scenario 1**

Package Specification

```
CREATE OR REPLACE PACKAGE CustomerManagement IS
   PROCEDURE AddCustomer(p_id NUMBER, p_name VARCHAR2, p_dob DATE, p_balance NUMBER);
   PROCEDURE UpdateCustomer(p_id NUMBER, p_name VARCHAR2, p_balance NUMBER);
   FUNCTION GetCustomerBalance(p_id NUMBER) RETURN NUMBER;
END CustomerManagement;
/
```

Package Body

```
CREATE OR REPLACE PACKAGE BODY CustomerManagement IS

   PROCEDURE AddCustomer(p_id NUMBER, p_name VARCHAR2, p_dob DATE, p_balance NUMBER) IS
   BEGIN
      INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
      VALUES (p_id, p_name, p_dob, p_balance, SYSDATE);
   EXCEPTION
      WHEN DUP_VAL_ON_INDEX THEN
         DBMS_OUTPUT.PUT_LINE('Error: Customer ID already exists.');
   END;

   PROCEDURE UpdateCustomer(p_id NUMBER, p_name VARCHAR2, p_balance NUMBER) IS
   BEGIN
      UPDATE Customers
      SET Name = p_name,
         Balance = p_balance,
         LastModified = SYSDATE
      WHERE CustomerID = p_id;
```

```
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Update failed: ' || SQLERRM);
    END;

    FUNCTION GetCustomerBalance(p_id NUMBER) RETURN NUMBER IS
        v_balance NUMBER;
    BEGIN
        SELECT Balance INTO v_balance FROM Customers WHERE CustomerID = p_id;
        RETURN v_balance;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RETURN -1;
    END;

END CustomerManagement;
/
```

- **Scenario 2**

#Package Specification

```
CREATE OR REPLACE PACKAGE EmployeeManagement IS
    PROCEDURE HireEmployee(p_id NUMBER, p_name VARCHAR2, p_position VARCHAR2, p_salary
NUMBER, p_department VARCHAR2, p_hiredate DATE);
    PROCEDURE UpdateEmployee(p_id NUMBER, p_position VARCHAR2, p_salary NUMBER);
    FUNCTION GetAnnualSalary(p_id NUMBER) RETURN NUMBER;
END EmployeeManagement;
/
```

#Package Body

```
CREATE OR REPLACE PACKAGE BODY EmployeeManagement IS

    PROCEDURE HireEmployee(p_id NUMBER, p_name VARCHAR2, p_position VARCHAR2, p_salary
NUMBER, p_department VARCHAR2, p_hiredate DATE) IS
    BEGIN
        INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
        VALUES (p_id, p_name, p_position, p_salary, p_department, p_hiredate);
    END;

    PROCEDURE UpdateEmployee(p_id NUMBER, p_position VARCHAR2, p_salary NUMBER) IS
    BEGIN
        UPDATE Employees
        SET Position = p_position,
            Salary = p_salary
        WHERE EmployeeID = p_id;
    END;

    FUNCTION GetAnnualSalary(p_id NUMBER) RETURN NUMBER IS
        v_salary NUMBER;
    BEGIN
        SELECT Salary INTO v_salary FROM Employees WHERE EmployeeID = p_id;
        RETURN v_salary * 12;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RETURN -1;
    END;

END EmployeeManagement;
/
```

- **Scenario 3**

#Package Specification

```
CREATE OR REPLACE PACKAGE AccountOperations IS
   PROCEDURE OpenAccount(p_account_id NUMBER, p_customer_id NUMBER, p_type VARCHAR2, p_balance NUMBER);
   PROCEDURE CloseAccount(p_account_id NUMBER);
   FUNCTION GetTotalBalance(p_customer_id NUMBER) RETURN NUMBER;
END AccountOperations;
/
```

#Package Body

```
CREATE OR REPLACE PACKAGE BODY AccountOperations IS

   PROCEDURE OpenAccount(p_account_id NUMBER, p_customer_id NUMBER, p_type VARCHAR2, p_balance NUMBER) IS
   BEGIN
      INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)
      VALUES (p_account_id, p_customer_id, p_type, p_balance, SYSDATE);
   EXCEPTION
      WHEN DUP_VAL_ON_INDEX THEN
         DBMS_OUTPUT.PUT_LINE('Account already exists.');
   END;

   PROCEDURE CloseAccount(p_account_id NUMBER) IS
   BEGIN
      DELETE FROM Accounts WHERE AccountID = p_account_id;
   EXCEPTION
      WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Error closing account: ' || SQLERRM);
    END;

    FUNCTION GetTotalBalance(p_customer_id NUMBER) RETURN NUMBER IS
        v_total NUMBER := 0;
    BEGIN
        SELECT SUM(Balance) INTO v_total FROM Accounts WHERE CustomerID = p_customer_id;
        RETURN NVL(v_total, 0);
    END;

END AccountOperations;
/
```