|  |
| --- |
| DECLARE  -- Declare the host variable G\_SHOPPER to hold the shopper ID  G\_SHOPPER NUMBER;  -- Declare other necessary variables here (e.g., customer information)  v\_customer\_name VARCHAR2(100);  v\_customer\_address VARCHAR2(100);  BEGIN  -- Assume that the employee is entering the shopper ID  G\_SHOPPER := :G\_SHOPPER; -- :G\_SHOPPER is the host variable binding (this would be set by your application)  -- Here, you would query the database to get the customer's information  SELECT customer\_name, customer\_address  INTO v\_customer\_name, v\_customer\_address  FROM customers  WHERE shopper\_id = G\_SHOPPER;    -- Output the customer information (example for a console-based application)  DBMS\_OUTPUT.PUT\_LINE('Customer Name: ' || v\_customer\_name);  DBMS\_OUTPUT.PUT\_LINE('Customer Address: ' || v\_customer\_address);  EXCEPTION  -- Exception handler to handle invalid shopper id (e.g., not found or invalid ID)  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('Invalid shopper id');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('An unexpected error occurred: ' || SQLERRM);  END;  / |

Assignment 4-5 Performing exception handling of predefined errors

A block of code has been created to retrieve basic customer information and can be found in file assignment04-05.txt in the Chapter.04 folder. The application screen was modified so that an employee can directly enter a customer number that could potentially cause an error. An appropriate exception handler needs to be added to the block. The handler should display a message to the screen that states 'Invalid shopper id'. Use a host variable named G\_SHOPPER to provide a shopper id. Test the block with a shopper id of 99.

Assignment 4-6 Performing exception handling of non-predefined errors

Brewbean's wants to add a check constraint on the quantity column of the BB\_BASKETITEM table. If a quantity value provided by a shopper on an item is greater than 20, Brewbean's wants to display a message saying "Check Quantity." The file named assignment04-06.txt can be found in the Chapter.04 folder. The first statement is an ALTER TABLE statement that needs to be executed to add the check constraint. The next item is a PL/SQL block containing an insert that tests this check constraint. Add the appropriate code to the block to trap the check constraint violation and display the desired message.

Assignment 4-7 Performing exception handling with user-defined errors

On occasion, some of Brewbean's customers mistakenly leave an item out of a basket already checked out, so they create a new basket containing the missing items. However, they request that the baskets be combined so that they are not charged extra shipping. A screen has been developed to allow an employee to modify the basket id of items in the BB\_BASKETITEM table to another existing basket to combine the baskets. A block has been constructed to support this screen and can be found as file assignment04-07.txt in the Chapter.04 folder. However, an exception handler needs to be added to trap the situation in which an invalid basket id is entered for the original basket. In this case, the UPDATE affects no rows but does not raise an Oracle error.

The handler should display a message stating "Invalid original basket id." Use a host variable named G\_OLD with a value of 30 and a host variable named G\_NEW with a value of 4 to provide the values to the block. First, verify that no item rows exist in the BB\_BASKETITEM table with a basket id of 30.