Restaurant Discount Calculator Documentation

Overview
The **Restaurant Discount Calculator** program calculates a discount percentage for restaurant customers based on their age. The discount logic is as follows:
- **100% discount** for customers aged **10 or below**.
- **20% discount** for customers aged **60 or above**.
- **0% discount** for customers aged between **11 and 59**.

The program uses a class structure to encapsulate the discount logic and provides an interactive prompt as well as a test suite to validate various age cases.

Class: `RestaurantDiscount`

This class contains methods for calculating and retrieving the discount information based on the customer's age.

Attributes
- `discount_percent` (`int`): Stores the calculated discount percentage.

Methods

1. **`calculate_discount(user_age)`**
   - **Description**: Calculates the discount percentage based on the `user_age`.
   - **Args**:
     - `user_age` (`int`): Customer's age.
   - **Returns**:
     - `int`: The discount percentage (0, 20, or 100).
   - **Logic**:
     - If `user_age` is 10 or below, it sets `discount_percent` to 100.
     - If `user_age` is 60 or above, it sets `discount_percent` to 20.
     - Otherwise, it sets `discount_percent` to 0.

2. **`get_discount_info(user_age)`**
   - **Description**: Provides a formatted message with the discount percentage for the given age.
   - **Args**:
     - `user_age` (`int`): Customer's age.
   - **Returns**:

  - `str`: A message with the calculated discount, e.g., `"Discount is 20%"`.
- **Logic**: Calls `calculate_discount` to determine the discount percentage and returns a formatted message.

Function: `run_tests()`

- **Description**: Runs a set of predefined test cases to validate the discount calculations. It verifies the function's accuracy by comparing actual output with expected output.
- **Test Cases**:
  - Ages tested: 9, 10, 30, 59, 60, and 61.
  - Expected discounts: 100% for ages 9 and 10, 0% for ages 30 and 59, 20% for ages 60 and 61.
- **Output**:
  - Prints a table displaying each test case's age, expected discount, actual discount, and whether the result matched the expectation (`Pass` or `Fail`).

Main Execution (`if __name__ == "__main__":`)

1. **Interactive Mode**:
   - Prompts the user to enter a customer's age.
   - If the age is valid, displays the discount message.
   - If the input is invalid (non-integer), shows an error message (`"Please enter a valid age (whole number)."`).

2. **Automatic Tests**:
   - Calls `run_tests()` to execute predefined test cases, displaying results in a structured table format.

Example Usage

Running in Interactive Mode

When running the code, the user will be prompted to enter the customer's age. The program will display the discount based on the entered age. Example:

```
Enter customer age: 60
Discount is 20%
```

```

#### Test Results

After entering an age, the program automatically runs predefined test cases, showing output similar to:

```
Test Results:

Case | Age | Expected | Output | Evaluation
----
1 | 9 | 100 | 100 | Pass
2 | 10 | 100 | 100 | Pass
3 | 30 | 0 | 0 | Pass
4 | 59 | 0 | 0 | Pass
5 | 60 | 20 | 20 | Pass
6 | 61 | 20 | 20 | Pass
```

---

### Summary of Code Flow
1. **User Input**: Prompts for age, validates input, and displays discount.
2. **Discount Calculation**: Determines the discount based on age using conditional statements.
3. **Test Suite**: Runs a set of test cases to validate the discount logic.

---
```