

prog1

February 28, 2025

```
[1]: class Recipient:
    """
    The Recipient class stores customer details such as name, contact, and
    ↪address.
    It provides methods to retrieve and update recipient information.
    """
    def __init__(self, name, contact, address):
        self.__name = name # Private attribute for recipient name
        self.__contact = contact # Private attribute for recipient contact
        self.__address = address # Private attribute for recipient address
    def get_details(self):
        return f"Name: {self.__name}, Contact: {self.__contact}, Address: {self.
    ↪__address}"
    def update_info(self, name, contact, address):
        self.__name = name
        self.__contact = contact
        self.__address = address
    # Setter and getters
    def get_name(self):
        return self.__name
    def set_name(self, name):
        self.__name = name
    def get_contact(self):
        return self.__contact
    def set_contact(self, contact):
        self.__contact = contact
    def get_address(self):
        return self.__address
    def set_address(self, address):
        self.__address = address
class Package:
    """
    The Package class stores package details such as package ID, dimensions, and
    ↪weight.
    It provides methods to retrieve package details.
    """
    def __init__(self, package_id, dimensions, weight):
```

```

        self.__package_id = package_id
        self.__dimensions = dimensions
        self.__weight = weight
    def get_package_details(self):
        return f"Package ID: {self.__package_id}, Dimensions: {self.
↪__dimensions}, Weight: {self.__weight}kg"
class DeliveryOrder:
    """
    The DeliveryOrder class represents a delivery order, linking a recipient to
↪a package.
    It includes order ID, delivery status, and methods for creating and updating
↪an order.
    """
    def __init__(self, order_id, recipient, package, status="Pending"):
        self.__order_id = order_id # Private order ID
        self.__recipient = recipient # Private Recipient object
        self.__package = package # Private Package object
        self.__status = status # Private status of delivery (e.g., "Pending",
↪"Delivered")
    def create_order(self):
        return f"Order {self.__order_id} Created for {self.__recipient.
↪get_details()}"
    def update_status(self, status):
        self.__status = status
class DeliveryNote:
    """
    The DeliveryNote class generates a delivery note for a specific order.
    It includes the note ID, associated order, and the date the note was issued.
    """
    def __init__(self, note_id, order, date_issued):
        self.__note_id = note_id # Private note ID
        self.__order = order # Private DeliveryOrder object linked to the note
        self.__date_issued = date_issued # Private date of issuance
    def generate_note(self):
        return f"Delivery Note {self.__note_id} generated for Order {self.
↪__order.create_order()}"
# Example Usage (Creating and Displaying Order Information)
recipient = Recipient("Sultan Allanjawi", "sultan.allanjawi@gmail.com", "Al
↪khwaneej 1, Dubai")
package = Package("PKG123", "30x20x10 cm", 7)
order = DeliveryOrder("DEL123456789", recipient, package)
delivery_note = DeliveryNote("DN-2025-001", order, "2025-03-01")
print(order.create_order()) # Creates a new order
print(delivery_note.generate_note()) # Generates a delivery note for the order

```

Order DEL123456789 Created for Name: Sultan Allanjawi, Contact:
sultan.allanjawi@gmail.com, Address: Al khwaneej 1, Dubai

Delivery Note DN-2025-001 generated for Order Order DEL123456789 Created for
Name: Sultan Allanjawi, Contact: sultan.allanjawi@gmail.com, Address: Al
khwaneej 1, Dubai