**HashSet with Custom object**

**Bachelor Party Gift Management System Using HashSet**

You are building a system to manage the gift inventory for a bachelor party. Each gift has a unique ID, a name, and a status to indicate whether it has been delivered or not. The system needs to:

1. Add new gifts to the inventory.
2. Mark gifts as delivered by their ID.
3. Remove gifts from the inventory by their ID.
4. Track how many gifts are still pending delivery after each operation.

**Task:**

Implement a BachelorPartyGiftSystem class using a HashSet to manage the list of gifts. Each gift should be represented by a Gift class with the following attributes:

* id (int): A unique identifier for the gift, starting from 501 and increasing sequentially for each new gift added.
* name (String): The name of the gift.
* delivered (boolean): Status indicating whether the gift has been delivered or not. (default = false)

The BachelorPartyGiftSystem class should include the following methods:

**Functionalities**

**Add Gift**

* Implement the method public void addGift(String name) to add a new gift to the inventory list.
* The gift's ID should be assigned automatically, starting from 501 and incrementing for each new gift.
* This method should take the gift name as input and create a new Gift object with a unique ID and the given name, then add it to the list of gifts.

**Deliver Gift**

* Implement the method public void deliverGift(int id) to mark a gift as delivered by its ID.
* If the gift is found, mark it as delivered and print a message indicating that the gift with the given ID has been delivered.
* If the gift is not found, print a message indicating that the gift with the given ID is not found.

**Remove Gift**

* Implement the method public void removeGift(int id) to remove a gift from the inventory by its ID.
* Print whether the gift was successfully removed or if it was not found.

**Track Pending Gifts**

* Implement the method public int countPendingGifts() to count and return the number of gifts that are still pending delivery.
* This method will be used to display how many gifts are still pending after performing the deliver and remove operations.

**Exception Handling**

* In case a gift is not found during the Deliver or Remove operations, a custom exception GiftNotFoundException should be thrown.
* This exception should be caught in the methods where the operation is performed and should print a message indicating that the gift with the given ID could not be found.

**Input Format**

**Adding Gifts:**

* The first line contains an integer n, representing the number of gifts to add to the inventory.
  1. For each gift, input: The name of the gift (String).

**Perform Operations:**

* A line containing the ID of the gift to mark as delivered.
* A line containing the ID of the gift to remove from the inventory.

**Output Format**

**Current Inventory:**

* Print "Gifts in the Inventory:"
* For each gift in the inventory, print the gift details in the format: Gift{id=<id>, name='<name>', delivered=<delivered>}

**Gift Delivery Status:**

* If the gift is delivered, print: Gift with ID <id> has been delivered.
* If the gift is not found, throw and catch the GiftNotFoundException and print: Gift with ID <id> not found.

**Gift Removal Status:**

* If the gift is removed, print: Gift with ID <id> removed successfully.
* If the gift is not found, throw and catch the GiftNotFoundException and print: Gift with ID <id> not found.

**Updated Inventory:**

* Print "Updated Inventory:"
* For each gift in the updated inventory, print the gift details in the format: Gift{id=<id>, name='<name>', delivered=<delivered>}

**Total Pending Gifts:**

* Print "Total pending gifts: <number>", where <number> is the total count of gifts that are still pending delivery.

**Sample Input 1**

3

Wine Bottle

Groom’s Hat

Custom Mug

502

503

**Sample Output 1**

Gifts in the Inventory:

Gift{id=501, name='Wine Bottle', delivered=false}

Gift{id=502, name='Groom’s Hat', delivered=false}

Gift{id=503, name='Custom Mug', delivered=false}

Gift with ID 502 has been delivered.

Gift with ID 503 removed successfully.

Updated Inventory:

Gift{id=501, name='Wine Bottle', delivered=false}

Gift{id=502, name='Groom’s Hat', delivered=true}

Total pending gifts: 1

**Sample Input 2**

4

Party Popper

Dance Shoes

Sunglasses

Cufflinks

510

508

**Sample Output 2**

Gifts in the Inventory:

Gift{id=501, name='Party Popper', delivered=false}

Gift{id=502, name='Dance Shoes', delivered=false}

Gift{id=503, name='Sunglasses', delivered=false}

Gift{id=504, name='Cufflinks', delivered=false}

Gift with ID 510 not found.

Gift with ID 508 not found.

Updated Inventory:

Gift{id=501, name='Party Popper', delivered=false}

Gift{id=502, name='Dance Shoes', delivered=false}

Gift{id=503, name='Sunglasses', delivered=false}

Gift{id=504, name='Cufflinks', delivered=false}

Total pending gifts: 4

**HashSet with Wrapper class( this can be tried with both Integer and String)**

**Factory Machine Management System**

You are developing a **Factory Machine Management System** to manage unique machine IDs in a factory. The system uses a **'HashSet'** to store machine IDs and ensure that all IDs are unique. Your task is to implement the following operations:

**Operations to be Performed:**

1. **Add Machine IDs:** Input a number of unique machine IDs and store them in the HashSet.
2. **Check for a Machine ID:** After storing the IDs, check if a specific machine ID exists in the HashSet and report whether it is present or not.
3. **Delete a Machine ID:** If the machine ID exists, it should be removed from the HashSet. If the ID is not found, indicate that it is not present.
4. **Display the Updated List of Machine IDs:** After attempting the deletion, display the updated list of machine IDs.
5. **Count Odd Machine IDs:** Count how many machine IDs are odd numbers and display the count.

**Input Format:**

* The first line contains an integer n, representing the number of machine IDs.
* The next n lines each contain a unique integer representing a machine ID.
* The following line contains an integer representing the machine ID to be checked.
* The final line contains an integer representing the machine ID to be removed.

**Output Format:**

**Check Machine ID:**

* Print: "The machine ID [ID] is present in the HashSet." if the ID is found.
* Print: "The machine ID [ID] is not present in the HashSet." if the ID is not found.

**Delete Machine ID:**

* Print: "The machine ID [ID] was removed from the HashSet." if the ID was found and removed.
* Print: "The machine ID [ID] was not found in the HashSet." if the ID was not found.

**Display Updated List:**

* Print: "Updated list of machine IDs:" followed by each ID in the HashSet.

**Count Odd Machine IDs:**

* Print: "Number of odd machine IDs: [count]" where [count] is the number of odd machine IDs.

**Sample 1 Input:**

4

150

301

502

703

301

502

**Sample 1 Output:**

The machine ID 301 is present in the HashSet.

The machine ID 502 was removed from the HashSet.

Updated list of machine IDs:

Machine ID: 150

Machine ID: 703

Machine ID: 301

Number of odd machine IDs: 2

**Sample 2 Input:**

3

22

44

66

30

100

**Sample 2 Output:**

The machine ID 30 is not present in the HashSet.

The machine ID 100 was not found in the HashSet.

Updated list of machine IDs:

Machine ID: 22

Machine ID: 44

Machine ID: 66

Number of odd machine IDs: 0