

## Transcript of Our Conversation

**User:** Why am I getting errors?

```
/* This is a stub for the House class */
import java.util.ArrayList;

public class House extends Building {
    //Attributes
    private ArrayList<Student> residents; // The <Student> tells Java what kind of data we plan to
    store IN the ArrayList
    private boolean hasDiningRoom;

    public House(ArrayList<Student>residents, boolean hasDiningRoom) {
        System.out.println("You have built a house: 🏠");
        this.residents = residents;
        this.hasDiningRoom = hasDiningRoom;
    }
}
```

**Assistant:** It looks like your constructor is missing a call to `super(...)`, which is required if `House` extends `Building`. Also, you're directly assigning the `residents` list, which might lead to unintended side effects.

---

**User:** Why is this throwing an exception at line 25?

```
public void moveIn(Student s){
    if(!residents.contains(s)){
        throw new RuntimeException("Student already in this House");
    } else{
        residents.add(s);
    }
}
```

**Assistant:** There's a **logical mistake** in your condition. You should check **if the student is already in the list** before throwing an exception. The corrected version should be:

```
public void moveIn(Student s){
    if (residents.contains(s)) {
        throw new RuntimeException("Student already in this House");
    } else {
        residents.add(s);
    }
}
```

```
}
```

---

**User:** How would I change this comment to code?

```
public void checkOut(String title){
    if(collection.contains(title)){
        if(collection.contains(true)){
            //change boolean to false
        } else {
            throw new RuntimeException("This book is already checked out");
        }
    } else{
        throw new RuntimeException("This book is not at this library");
    }
}
```

**Assistant:** The main issue is that `collection.contains(true)` does not check the value of a **specific** book. Instead, you need to use `collection.get(title)`. Here's the correct version:

```
public void checkOut(String title){
    if(collection.containsKey(title)){
        if(collection.get(title)){ // Check if the book is available
            collection.put(title, false); // Mark it as checked out
        } else {
            throw new RuntimeException("This book is already checked out");
        }
    } else{
        throw new RuntimeException("This book is not at this library");
    }
}
```

---

**User:** How can I fix this?

```
public void printCollection(){
    System.out.println("The following books are in the library: ");
    for (int i = 0; i < collection.size(); i++) {
        System.out.println(collection.get(i) + "and isn't checked out: " + collection.get(i));
    }
}
```

**Assistant:** The main issue is that `Hashtable` does **not** support index-based access. Instead, you need to iterate over its **keys**. Here's the corrected version:

```
public void printCollection() {
    System.out.println("The following books are in the library:");
    for (String title : collection.keySet()) { // Loop through book titles
        System.out.println(title + " and isn't checked out: " + collection.get(title));
    }
}
```

---

**User:** How do I get a key at an index?

**Assistant:** Since `Hashtable` doesn't support index-based access, you can **convert its keys into a list** and access the key by index:

```
public String getKeyAtIndex(int index) {
    if (index < 0 || index >= collection.size()) {
        throw new IndexOutOfBoundsException("Invalid index: " + index);
    }
    ArrayList<String> keys = new ArrayList<>(collection.keySet()); // Convert keys to a list
    return keys.get(index);
}
```

---

**User:** What is `pass` in Java?

**Assistant:** Java does **not** have a `pass` statement like Python. However, you can simulate it by using:

1. **An empty block:**

```
public void myMethod() {}
```

2. **A `return;` statement in a void method:**

```
public void myMethod() {
    return; // Explicitly doing nothing
}
```

3. **Using `continue;` to skip an iteration in loops:**

```
for (int i = 0; i < 5; i++) {
    if (i == 2) {
        continue; // Skip when i == 2
    }
    System.out.println(i);
}
```

---

This transcript captures our discussion, including the errors and solutions. Let me know if you need any modifications! 🚀