

# Simplst User Guide

[View on GitHub](#)

## Simplst User Guide

### Introduction

*Simplst* is a Warehouse Management System (WMS) for warehouse workers and managers to manage your warehouse inventory and orders. Simplst is optimised for use via a Command Line Interface (CLI).

Simplst aims to improve the efficiency of standard warehouse procedures such as inventory and order management by having a CLI app which is simple to set up and quick to use. Simplst is mainly used to add, remove, list and view goods and orders for the warehouse.

### Purpose

This User Guide is meant for both warehouse workers and managers to learn how to use the Simplst's features which will help you to improve efficiency in managing your warehouse inventory and orders. Our guide will show you how to get started with Simplst, and introduce the Simple commands to interact with the application.

As a new user of Simplst, you can go through the guide in detail starting from the [quick start](#) section. The guide also has [explanations](#) of certain key terms which are present throughout the guide.

As a seasoned user of Simplst, the [command summary](#) section would be useful to show you an overview of the commands available and how to use them.

Lastly, the [FAQ](#) section will provide you with answers to the questions that you might have while using Simplst. Otherwise, the [content's page](#) below displays all the features that Simplst has to offer. You can click on any section, and you will be brought to that section directly.


### Contents page

- [Explanation of Key Terms or Symbols](#)
- [Quick start](#)
- [Features](#)

- [Help Command](#)
- [Unit Good Commands](#)
  - [Adding a Unit Good](#)
  - [Removing a Unit Good](#)
  - [Listing Unit Goods](#)
- [Good Commands](#)
  - [Adding quantity of a Good](#)
  - [Removing quantity of a Good](#)
  - [Listing available Goods](#)
  - [Viewing a Good](#)
  - [Finding a Good](#)
- [Order Commands](#)
  - [Adding an Order](#)
  - [Removing an Order](#)
  - [Listing current Orders](#)
  - [View Order](#)
  - [Fulfill Order](#)
- [Orderline Commands](#)
  - [Adding an Orderline](#)
  - [Removing an Orderline by Quantity](#)
  - [Listing Orderlines in an Order](#)
- [Exiting the Program](#)
  - [Saving the Program State](#)
- [FAQ](#)
- [Command Summary](#)

## Explanation of key terms or symbols

Terms or Symbols used	Further Explanation
<a href="#">Unit Good</a>	A template containing details of a good.
SKU	Stands for Stock-Keeping Unit. It is the unique unit number for a specific warehouse item. It can contain characters and numbers (e.g WC01).
Capacity	Size of the object defined as Small, Medium and Large. Small Medium Large

Terms or Symbols used	Further Explanation
Good	A Good is the actual good that is currently in the warehouse inventory.
Order	An order in Simplst is used to know who made the order and what goods to deliver to that shipping address.
Orderline	An orderline is the goods required by the order.
Fulfill	Used to check and indicate if an order is completed.
Words in Markup	Used to highlight keywords used for commands and commands themselves.
	Used to denote features for warehouse managers that have higher sensitivity.
*Optional*	Fields in Markup bounded by asterisks (*) are optional to be filled in, but should be replaced with a space ' ' instead.

## Quick Start

1. Ensure that you have Java 11 or above installed. If you do not, you can download it from [here](#).

### Java SE Development Kit 11.0.14







Java SE subscribers will receive JDK 11 updates until at least **September of 2026**.

These downloads can be used for development, personal use, or to run Oracle licensed products. Use for other purposes, including production or commercial use, requires a Java SE subscription or another Oracle license.

JDK 11 software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#).

JDK 11.0.14 [checksum](#)

**Linux**   **macOS**   **Solaris**   **Windows**

Product/file description	File size	Download
ARM 64 Debian Package	134.61 MB	 <a href="#">jdk-11.0.14_linux-aarch64_bin.deb</a>
ARM 64 RPM Package	140.63 MB	 <a href="#">jdk-11.0.14_linux-aarch64_bin.rpm</a>
ARM 64 Compressed Archive	156.99 MB	 <a href="#">jdk-11.0.14_linux-aarch64_bin.tar.gz</a>
x64 Debian Package	138.3 MB	 <a href="#">jdk-11.0.14_linux-x64_bin.deb</a>
x64 RPM Package	144.48 MB	 <a href="#">jdk-11.0.14_linux-x64_bin.rpm</a>
x64 Compressed Archive	160.87 MB	 <a href="#">jdk-11.0.14_linux-x64_bin.tar.gz</a>

2. Down the latest version of **Simplst** from [here](#).
3. Copy the file to the folder you want to use as the home folder for Simplst.
4. Open the terminal in the folder which contains Simplst.jar and run `java -jar Simplst.jar`.

The Startup should look like below:

```
New login.  
Hello from  
Simplst  
What would you like to do?
```

5. Type the commands in the terminal and press **Enter** to execute them. For example: typing `help` and pressing **Enter** will show you the features available in Simplst and how to type the command into the command line.

## Features

### Help Command `help`

All the features can be viewed on the Command-Line Interface using the `help` command if you want to view the commands there instead. This is done by using the `help` command. You can also use flags to indicate which specific set of commands to view.

Help Commands:

- `help` - View all commands
- `help ug/` - View commands related to Unit Goods
- `help g/` - View commands related to Goods
- `help o/` - View commands related to Orders and Orderlines

### Unit Good Commands

A Unit Good is a template of a good. A unit good should be added to the warehouse to allow Simplst to know what kind goods will be added to the warehouse later. Unit Goods can help Simplst estimate storage capacity in the future and other predictive features for future versions.

A Unit Good contains:

- Unique SKU
- Name

- Description of unit good
- Capacity

## Adding a Unit Good `add ug/`

Add a new unit good to the warehouse, creating a Good with quantity 0 in the process.

Format: `add ug/ sku/[SKU] n/[NAME] d/*[DESCRIPTION]* size/[CAPACITY]`

Note:

- Optional description field. If there are no descriptions, key in `d/` followed by a space before keying in the next parameter.
- If the size input is not either [SMALL / MEDIUM / LARGE], the capacity would be set to the default MEDIUM

Example of adding a unit good:

Adding a Unit Good with SKU being WC1, with name Wooden Chair, description as Chair made of oak from Europe, and a size of Medium

```
add ug/ sku/WC1 n/Wooden Chair d/Chair made of oak from Europe size/Medium
```

Expected Output

```
Unit Good of SKU: WC1 added to warehouse
Another command?
```

## Removing a Unit Good `remove ug/`

Removing a unit good from the warehouse.

Tips:

- You can use this command when there was a mistake adding a Unit Good, or if you know that the warehouse will no longer take in this Unit Good anymore.

Format: `remove ug/ sku/[SKU]`

Example of removing a unit good:

Removing a Unit Good of SKU WC1

```
remove ug/ sku/WC1
```

Expected Output

```
Unit Good of SKU: WC1 has been removed from warehouse  
Another command?
```

## Listing Unit Goods `list ug/`

Listing all Unit Goods already added to the warehouse.

Format: `list ug/`

Example of listing Unit Goods:

```
list ug/
```

Expected Output

```
List of unit goods (in no order):  
    WC1 - Wooden Chair (chair made of oak from europe)  
    WC2 - Wooden Table (Table made of oak from Italy)  
Another command?
```

## Good Commands

A Good is the actual good that is currently in the warehouse inventory. It will also have the same information as the Unit Good previously added into Simplst with the same SKU. In addition to the details already present in the Unit Good, a Good will also contain:

- Quantity of Good available in the warehouse

## Adding quantity of a Good `add g/`

Adding a quantity of a [Unit Good](#) to be available at the warehouse.

Format: `add g/ sku/[SKU] qty/[QUANTITY]`

Example of adding quantity to a specific Unit Good:  
Adding 30 of a good with its SKU being WC1

```
add g/ sku/WC1 qty/30
```

Expected Output

```
30 of Good with SKU: WC1 added to warehouse
```

```
Another command?
```

## Removing quantity of a Good `remove g/`

Removing a quantity of a [Unit Good](#) available in the warehouse.

Format: `remove g/ sku/[SKU] qty/[QUANTITY]`

Note:

- Used when certain units of the good has been shipped out or damaged and thus quantity has to be decreased.

Example of removing quantity of a specific Unit Good:

Removing 10 of a good with its SKU being WC1

```
remove g/ sku/WC1 qty/10
```

Expected Output

```
10 of Good with SKU: WC1 has been removed from warehouse  
Another command?
```

## Listing available Goods `list g/`

Listing all available goods in the warehouse.

Format: `list g/`

Example of listing all Goods (This will continue from the above example [here](#)):

Note:

- If a Unit Good is added, but the Good has yet to be added into the warehouse (There are currently 0 quantity of the Good in the warehouse). It will not show up in the list of Goods.

```
list g/
```

Expected Output

```
List of available goods with its quantity:  
WC1 - Wooden Chairs (chair made of oak from Europe) [Qty: 30]
```

```
Another command?
```

## Viewing a Good `view g/`

View a specific Good in the warehouse using its SKU. This will show more details of the specified Good.

Format: `view g/ sku/[SKU]`

Note:

- Differs from listing available Goods as Viewing a Good looks into a single specific Good, while listing displays all Goods present in the warehouse

Example of finding a Good:

Viewing details of the Wooden Chair with SKU: WC1

```
view g/ sku/WC1
```

Expected Output

```
WC1 - Wooden Chair (chair made of oak from Europe) [Qty: 30]
Unit size of good: MEDIUM
Another command?
```

If there are no Unit Good in the warehouse with such SKU, Simplst will inform you accordingly. Example of trying to view an SKU not in the warehouse.

```
view g/ sku/fakeSKU
```

Expected Output

```
Could not find good with given SKU! Please check input SKU!
Another command?
```

## Finding a Good `find`

Finding Goods in the warehouse through the name of the [Unit Good](#). After finding the goods, it will show the details of all the goods found.

Tips:



- This command is good for finding the SKU of a Good or Unit Good which you only know the name for.

Format: `find n/[NAME]`

Note:

- Differs from Viewing a Good as Finding a Good works through searching a keyword and does not require one to know the exact SKU of the Good that you are looking for

Example of finding a Good:

Finding if any Unit Good with “Wooden” in its name exists in the warehouse.

```
find n/Wooden
```

Expected Output

```
WC1 - Wooden Chair (chair made of oak from Europe) [Qty: 30]
Unit size of good: MEDIUM
WC2 - Wooden Table (Table made of oak from Italy) [Qty: 0]
Unit size of good: LARGE
Another command?
```

Note:

- If there are no Unit Goods in the warehouse with its name containing any part of the input, Simplst will inform you accordingly. Example of trying to find spoon in the warehouse.

```
find n/spoon
```

Expected Output

```
Could not find any item contining 'spoon' in the inventory!
Another command?
```

## Order Commands

An order in Simplst is used to know who made the order and what [Goods](#) to deliver to that shipping address.

An Order contains:

- Order ID which will be a unique positive number

- Receiver name
- Shipping address

## Adding an Order `add o/`



Adding a new order to be tracked in the warehouse.

Format: `add o/ oid/[ORDER_ID] r/[RECEIVER_NAME] addr/[SHIPPING_ADDRESS]`

Notes:

- Order ID (oid) must be a unique non-negative number (i.e 0, 1, 2... etc)

Example of adding an order for Danny Phantom who lives at Amity Park:

```
add o/ oid/1 r/Danny Phantom addr/Amity Park
```

Expected Output

```
Order 1 added to the warehouse
Another command?
```

## Removing an Order `remove o/`



Removing an existing order in the Warehouse.

Format: `remove o/ oid/[ORDER_ID]`

Example of removing order 1:

```
remove o/ oid/1
```

Expected Output

```
Order 1 has been removed
Another command?
```

## Listing Current Orders `list o/`



Listing all orders in the warehouse. This would show the order details such as:

- Order id
- Receiver Name
- Shipping address
- Fulfilled/Completion status

Tips:

- Listing is a good way to find out the Order Id of any order in Simplst.

Format: `list o/`

Example of listing all Orders (This will continue the above example [here](#)):

```
list o/
```

Expected Output

```
List of orders:
      1 - Danny Phantom (Amity Park) : completed
Another command?
```

View Order `view o/`



View a specific order in the warehouse. This would show the order details such as:

- Order id
- Receiver Name
- Shipping address
- Details of the orderlines relating to this order

Format: `view o/ oid/[ORDER_ID]`

Example of viewing order ID 1: `view o/ oid/1`

Expected Output

```
Viewing order with order ID 1
Receiver: Danny Phantom
```

```
Shipping address:Amity Park
Items in the order:
8 of WC1 - Wooden Chair needed (Not Done)
Another command?
```

If there are no such order in the warehouse with the specified order ID, Simplst will inform you accordingly.

Example of trying to find order ID 4: `view o/ oid/4`

Expected Output

```
Could not find order with given id!
Another command?
```

## Fulfill Order `fulfill`



Fulfill a currently unfulfilled order in the warehouse. To fulfill an order, Simplst will check if all the [Orderlines](#) relating to the order have their required quantities met.

When all the orderlines quantities are fulfilled, the order will be considered as fulfilled.

Format: `fulfill oid/[ORDER_ID]`

Note:

- If the current quantity in the warehouse is lesser than the required quantity to fulfill an orderline, Simplst will not fulfill that orderline and consequently not fulfill that order

Example of fulfilling order 1:

```
fulfill oid/1
```

Expected Output

```
Order 1 completed
Another command?
```

If there are no such order in the warehouse with the specified order ID, Simplst will inform you accordingly. Example of trying to fulfill order ID 4. `view o/ oid/4`

Expected Output

```
No order with oid: 4 found in the warehouse  
Another command?
```

If the order in the warehouse with the specified order ID are already fulfilled, Simplst will inform you accordingly. Example of trying to fulfill order ID 1 which is already fulfilled.

```
fulfill oid/1
```

Expected Output

```
Order 1 already completed  
Another command?
```

## Orderline Commands

An orderline records the Goods required by an [order](#). It is equivalent to a [Good](#) with the same SKU, and contains the quantity of the Good required to fulfill the particular order.

In addition to the details in a Good. An orderline will also contain:

- Quantity of Good required to fulfill the order

## Adding an Orderline `add og/`

This will add an orderline to a specific order.

Format: `add og/ oid/[ORDER_ID] sku/[SKU] qty/[QUANTITY_NEEDED_FOR_ORDER]`

Example of adding an orderline to order ID 1 for 10 Wooden Chairs:

```
add og/ oid/1 sku/WC1 qty/10
```

Expected Output

```
Wooden Chair is added to order. 8 required to fulfill  
8 of WC1 is added to order number 1  
Another command?
```

## Removing a quantity of an Orderline `remove og/`

Removing the quantity required in an orderline from a specific order.

Format: `remove og/ oid/[ORDER_ID] sku/[SKU] qty/[QUANTITY_TO_REMOVE]`

Example of removing 1 Wooden Chair orderline from order ID 1:

```
remove og/ oid/1 sku/WC1 qty/1
```

Expected Output

```
1 Wooden Chair is removed.  
Another command?
```

## Listing Orderlines in an Order `list og/`

Listing the orderlines from a specific order.

Format: `list og/ oid/[ORDER_ID]`

Example of listing orderlines linked to order ID 1:

```
list og/ oid/1
```

Expected Output

```
9 of WC1 - Metal Chair needed (Not Done)  
Another command?
```

## Exiting the Program `bye`

In order to properly exit the program and ensure the session is saved onto a file, we use the `bye` command.

Example of using the `bye` to exit the program and a successful save of the session:

```
bye
```

Expected Output

```
Warehouse information succesfully stored in output/WAREHOUSE.json  
State saved!  
Bye
```

## Saving Program State

The program uses a save-file to save all information before the program exits, and loads back everything when the program first starts up.

All you have to do is ensure that the jar file and the save-file are in a specific file directory structure as shown below:

```
tp.main.jar output/ WAREHOUSE.json
```

You are strongly advised against editing the save-file structure. We did not design this system to handle manual edits to the save-file.

## FAQ

Q: How to use the program effectively as a new user?

A: Start by [adding Unit Goods](#) to your Warehouse. Once satisfied with the content, begin [populating the available Goods](#) so that there are Goods available in the warehouse. As orders for Goods come in, you can begin [adding Orders](#) and [goods to those Orders](#) to simplify workflow for your Warehouse needs. Other features can be done as well including [listing available Goods](#) to enhance usage of this program.

Q: Can I edit the savefile directly?

A: As mentioned in the section [Saving Program State](#), it is not recommended, but you are able to do so by opening the file and editing its contents. However, and manual edits that are corrupt/incorrect/incomplete and the problems that follow will not fall under the responsibility of the team. Do contact the team through email at [queriesforsimplst@gmail.com](mailto:queriesforsimplst@gmail.com) if there are any issues faced or if the file has accidentally been manually edited.

Q: What happens when I call fulfill for an order but the warehouse does not have enough goods to fulfill all orderlines in that order?

A: [Fulfill](#) will not consider an order fulfilled or completed if any single orderline is not fulfilled. You can call fulfill once the warehouse has enough quantity of the corresponding [Good](#) and Simplst will attempt to complete the unfulfilled orderlines in that order again. The order will only be considered fulfilled when all orderlines in that order is fulfilled.

## Command Summary

Action	Command to be keyed into the terminal
--------	---------------------------------------

Action	Command to be keyed into the terminal
Help	Display commands for ... General: <code>help</code> Unit Good: <code>help ug/</code> Good: <code>help g/</code> Order: <code>help o/</code>
Add Unit Good	<code>add ug/ sku/[SKU] n/[NAME] d/*[DESCRIPTION]* size/[CAPACITY]</code>
Remove Unit Good	<code>remove ug/ sku/[SKU]</code>
List Unit Good	<code>list ug/</code>
Add Good quantity	<code>add g/ sku/[SKU] qty/[QUANTITY]</code>
Remove Good quantity	<code>remove g/ sku/[SKU] qty/[QUANTITY]</code>
List Goods (with quantity displayed)	<code>list g/</code>
Viewing a Good	<code>view g/ sku/[SKU]</code>
Find Good	<code>find n/[NAME]</code>
Add Order	<code>add o/ oid/[ORDER_ID] r/[RECEIVER_NAME] addr/[SHIPPING_ADDRESS]</code>
Remove Order	<code>remove o/ oid/[ORDER_ID]</code>
List Orders	<code>list o/</code>
View Order	<code>view o/ oid/[ORDER_ID]</code>
Fulfill Order	<code>fulfill oid/[ORDER_ID]</code>
Add Orderline	<code>add og/ oid/[ORDER_ID] sku/[SKU] qty/[QUANTITY_NEEDED]</code>
Remove Orderline Quantity	<code>remove og/ oid/[ORDER_ID] sku/[SKU] qty/[QUANTITY_TO_REMOVE]</code>
List Orderlines	<code>list og/ oid/[ORDER_ID]</code>



---

**tp** is maintained by **AY2122S2-CS2113T-T09-4**.

This page was generated by [GitHub Pages](#).