HONG KONG INSTITUTE OF VOCATIONAL EDUCATION (SHA TIN)
DEPARTMENT OF INFORMATION TECHNOLOGY

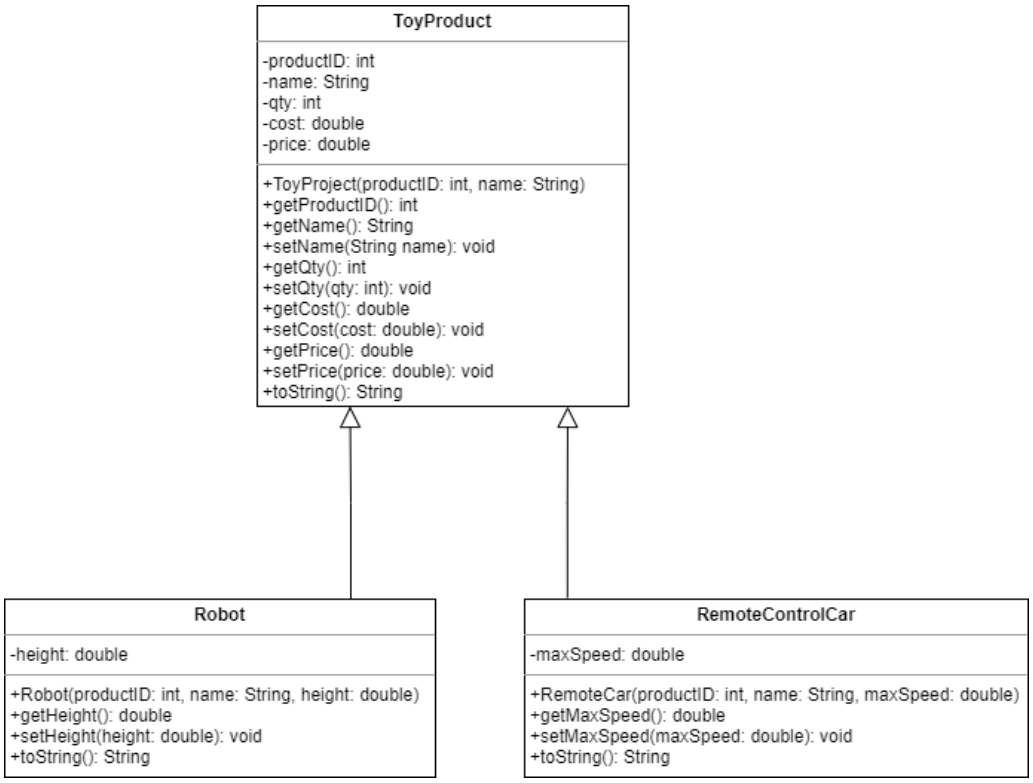
HIGHER DIPLOMA IN SOFTWARE ENGINEERING (IT114105)

Module Name:	Contemporary Topics in Software Engineering
Module Code:	ITP4507
Assignment Number:	One
Hand-in:	**12 November, 2021** (On or before 4:30 PM to Moodle)
Weighting of This Assignment:	50% of the End of Module Assessment

This assignment must be done by individual only. Plagiarism will be treated seriously. Any assignments that are found involved wholly or partly in plagiarism (no matter the assignments are from the original authors or from the plagiarists) will score Zero mark. **Late submission will NOT be accepted**.

Task Specification

Imaginary Toys Corporation is a toy trading company in Hong Kong. The company purchases various toy products from overseas and selling the products in Hong Kong. The company plans to develop a toy inventory management system (TIMS) for maintaining inventory records. The following is the simplified class diagram of existing data maintained by the company.



As a system analyst of the Company, you are required to design and develop TIMS. You are reminded that the design of the existing classes: ToyProduct, Robot and RemoteControlCar must be kept unchanged.

TIMS should provide the following functions:

1. Create a ToyProduct record with zero quantity (Robot or RemoteControlCar or any new kind of ToyProduct in the coming future).
2. Display ToyProduct details (such as productID, name, qty, height/maximum speed) by a given productID (input code=* to display all records)
3. Purchase ToyProduct and update the qty and the cost of the inventory records
4. Sell ToyProduct and update the qty and the price of the inventory records
5. Undo last command
6. Redo the last undone command
7. Show undo/redo list

Your system design should conform to the Open Closed Principle so that your design should easily be extended to support new functions (e.g. change of name of ToyProduct) and toy products, for example, creation of Doll record which is a subclass of ToyProduct.

You **MUST** apply the following design patterns for your new system

- **Command pattern** to provide the "create toy", "display toy", "purchase toy", "sell toy", "undo", "redo" and "display undo/redo list" functions
- **Factory pattern *or* Abstract Factory Pattern** to create different Command objects and ToyProduct objects (e.g. Robot object, RemoteControlCar object, etc.)
- **Memento pattern** to provide "Undo" and "Redo" functions

Assignment Report

In addition to the system development, you are required to write up a **Short Report** covers the following sections:

1. Assumptions regarding the problem context
2. Application design with class diagram
3. Discussion and explanation on each of the design patterns applied to the application
4. User Guide
5. Test Plan and Test Cases
6. Well documented Source Code

Mark Allocation

Your assignment work will be marked according to the following criteria.

Work	Mark Allocated
System Coding and Implementation	
a) Implementation of the system and coding style **(Hard-coded output will result in zero mark.)**	30%
b) Correctness of system functions * (Hard-coded output will result in zero mark.)	15%
c) User Guide	5%
d) Test Plan and Test Cases (Will be used in testing your own application.)	5%
System Analysis and Design, and Discussion	
e) Design of your system and correct use of design patterns	20%
f) Application design with class diagram	10%
g) Discussion and explanation on each of the design patterns applied to the application	15%
Total	**100%**

Note: * Please note that your source code will be recompiled and tested for the correctness of the system functions. Your implementation is required to support the 'Copy and Paste' method for

testing which is described in page 12.

Submission of Assignment Work

1. The front page of your submission should include the programme title, module title, student identity number(s), student name(s), and group number.

2. Submit a zip file of all your work to the module's Moodle website (https://moodle2122.vtc.edu.hk/course/view.php?id=2473):
   - Well documented Source Code of your program. Store the Source Code files in Folder "source code\" of your zip file. Store compiled class files and the "run.bat" file for executing the program in Folder "bin\" of your zip file.
   - Report for analysis, design, discussion, user guide, test plan and test cases of your following work. Store the report in Folder "report\" of your zip file.
     A. The assumption made during analysis and design of the application
     B. System design on your application with class diagram
     C. Discussion on the design patterns that applied on your program
     D. User Guide and Test Plan with Test Cases (describe how your program works and develop different test cases for testing each functionality of your program – please include all the required screen dumps).

3. Submit according to the guideline on the top part of cover page. **Late submission will NOT be accepted**.

Extra Reference

Testing Method

This sample run is served for reference only. You are free to design your own user interface. But to make the testing environment simple and to apply the "Copy and Paste" testing method described on page 12 easily, you are advised to accept user input at the command prompt as shown in the sample run below.

Sample Run of assignment

You may follow the design of user interface shown in this sample run in DOS command prompt.

User's inputs are in bold face.

1. Create ToyProduct record (c)

```
Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy,  d = display toy,  p = purchase toy,  s = sell toy,
u = undo,  r = redo,  l = list undo/redo,  x = exit system

c
Enter toy type (ro=Robot/rc=Remote Control Car):
ro
Enter id, name and height(cm):
1001, Z Gundam, 25
New toy product record created.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy,  d = display toy,  p = purchase toy,  s = sell toy,
u = undo,  r = redo,  l = list undo/redo,  x = exit system

c
Enter toy type (ro=Robot/rc=Remote Control Car):
rc
Enter id, name and maximum speed(km/hr):
2001, Ferrari F7, 40
New toy product record created.
```

2. Display one ToyProduct record (s)

```
Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy,  d = display toy,  p = purchase toy,  s = sell toy,
u = undo,  r = redo,  l = list undo/redo,  x = exit system

d
Enter ID (* to display all):
1001

Toy product information
ID: 1001
Name: Z Gundam
Quantity: 0
Height(cm): 25
Cost($): 0
Price($): 0
```

Display all records

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

d
Enter id (* to display all):
*****

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	0	25cm	$0	$0
2001	Ferrari F7	0	40km/hr	$0	$0

3. Purchase ToyProduct (p)

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

p
Enter code:
1001
Quantity to be purchased:
200
Purchasing cost:
250
Purchased 200 boxes of Z Gundam. Current quantity is 200. Current cost is $250. Price is $0.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

p
Enter code:
2001
Quantity to be purchased:
100
Purchasing cost:
300
Purchased 100 boxes of Ferrari F7. Current quantity is 100. Current cost is $300. Price is $0.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

d
Enter code (* to display all):
*****

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	200	25cm	$250	$0
2001	Ferrari F7	100	40km/hr	$300	$0

4. Sell ToyProduct (s)

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

s
Enter code:
1001
Quantity to be sold:
100
Selling price:
450
Sold 100 boxes of Z Gundam. Current quantity is 100. Selling price is $450.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

s
Enter code:
2001
Quantity to be sold:
30
Selling price:
500
Sold 30 boxes of Ferrari F7. Current quantity is 70. Selling price is $500.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

d
Enter code (* to display all):
*****

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	100	25cm	$250	$450
2001	Ferrari F7	70	40km/hr	$300	$500

Selling an invalid quantity (current quantity < selling quantity) from ToyProduct will display a warning message. Note that this invalid operation should not be shown in Undo List afterward.

```
Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy,  d = display toy,  p = purchase toy,  s = sell toy,
u = undo,  r = redo,  l = list undo/redo,  x = exit system

s
Enter code:
1001
Quantity to be sold:
200
Selling price:
500
Invalid quantity (current quantity < selling quantity).
```

5. Display the Undo/Redo List (l)

```
Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy,  d = display toy,  p = purchase toy,  s = sell toy,
u = undo,  r = redo,  l = list undo/redo,  x = exit system

l

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300
Sell 100 1001 Z Gundam 450
Sell 30 2001 Ferrari F7 500

Redo List:
Empty
```

6. Undo the last command in the Undo List (u)

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

u
Undo completed.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

l

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300
Sell 100 1001 Z Gundam 450

Redo List:
Sell 30 2001 Ferrari F7 500

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

u
Undo completed.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

l

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300

Redo List:
Sell 30 2001 Ferrari F7 500
Sell 100 1001 Z Gundam 450

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,

u = undo, r = redo, l = list undo/redo, x = exit system

d
Enter code (* to display all):
*****

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	200	25cm	$250	$0
2001	Ferrari F7	100	40km/hr	$300	$0

7. Redo the last command in the Redo List (r)

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

l

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300

Redo List:
Sell 30 2001 Ferrari F7 500
Sell 100 1001 Z Gundam 450

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

r
Redo completed.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

d
Enter code (* to display all):
*****

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	100	25cm	$250	$450
2001	Ferrari F7	100	40km/hr	$300	$0

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

l

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300
Sell 100 1001 Z Gundam 450

Redo List:
Sell 30 2001 Ferrari F7 500

8. Exit the System (x)

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

x

Exit...

End of Sample Run

You can ease the testing by using the '**Copy and Paste**' method rather than inputting data manually. Prepare a text file, which includes all user inputs in a test run. By using the '**Copy and Paste**' method, you can automatically input in the command prompt window and then get the result automatically (without the input data echoed). The following is an example of the text file for user inputs.

Sample User Inputs for a Test Run

```
c
ro
1001
Z Gundam
25
c
rc
2001
Ferrari F7
40
d
1001
d
*
p
1001
200
250
p
2001
100
300
d
*
s
1001
100
450
s
2001
30
500
d
*
s
1001
200
500
l
u
l
u
l
d
*
l
r
d
*
l
x
```

Expected Output of the Test Run

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter toy type (ro=Robot/rc=Remote Control Car): Enter id, name and height(cm): New toy product record created.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter toy type (ro=Robot/rc=Remote Control Car): Enter id, name and maximum speed(km/hr): New toy product record created.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter ID (* to display all):

Toy product information
ID: 1001
Name: Z Gundam
Quantity: 0
Height(cm): 25
Cost($): 0
Price($): 0

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter ID (* to display all):

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	0	25cm	$0	$0
2001	Ferrari F7	0	40km/hr	$0	$0

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code: Quantity to be purchased: Purchasing cost: Purchased 200 boxes of Z Gundam. Current quantity is 200. Current cost is $250. Price is $0.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]

c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code: Quantity to be purchased: Purchasing cost: Purchased 100 boxes of Ferrari F7. Current quantity is 100. Current cost is $300. Price is $0.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code (* to display all):

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	200	25cm	$250	$0
2001	Ferrari F7	100	40km/hr	$300	$0

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code: Quantity to be sold: Selling price: Sold 100 boxes of Z Gundam. Current quantity is 100. Selling price is $450.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code: Quantity to be sold: Selling price: Sold 30 boxes of Ferrari F7. Current quantity is 70. Selling price is $500.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code (* to display all):

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	100	25cm	$250	$450
2001	Ferrari F7	70	40km/hr	$300	$500

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code: Quantity to be sold: Selling price: Invalid quantity (current quantity < selling quantity).

Toy Inventory Management System (TIMS)

Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300
Sell 100 1001 Z Gundam 450
Sell 30 2001 Ferrari F7 500

Redo List:
Empty

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Undo completed.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300
Sell 100 1001 Z Gundam 450

Redo List:
Sell 30 2001 Ferrari F7 500

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Undo completed.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300

Redo List:
Sell 30 2001 Ferrari F7 500
Sell 100 1001 Z Gundam 450

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code (* to display all):

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	200	25cm	$250	$0
2001	Ferrari F7	100	40km/hr	$300	$0

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300

Redo List:
Sell 30 2001 Ferrari F7 500
Sell 100 1001 Z Gundam 450

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Redo completed.

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,
u = undo, r = redo, l = list undo/redo, x = exit system

Enter code (* to display all):

Toy product information

ID	Name	Quantity	Other Info	Cost	Price
1001	Z Gundam	100	25cm	$250	$450
2001	Ferrari F7	100	40km/hr	$300	$0

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy, d = display toy, p = purchase toy, s = sell toy,

```
u = undo,    r = redo,    l = list undo/redo,    x = exit system

Undo List:
Create 1001 Z Gundam
Create 2001 Ferrari F7
Purchase 200 1001 Z Gundam 250
Purchase 100 2001 Ferrari F7 300
Sell 100 1001 Z Gundam 450

Redo List:
Sell 30 2001 Ferrari F7 500

Toy Inventory Management System (TIMS)
Please enter command: [c | d | p | s | u | r | l | x]
c = create toy,    d = display toy,    p = purchase toy,    s = sell toy,
u = undo,    r = redo,    l = list undo/redo,    x = exit system

Exit...
```

Requirement for Scanner usage

Wrong Scanner usage (more than one object of Scanner is created for reading keyboard input):

```
// create new Scanner objects in loop
do {
	Scanner sc = new Scanner( System.in);
	choice = sc.nextInt();
} while (choice != 1);
```

Correct Scanner usage (only one Scanner object is created for reading keyword input):
Following is an example program to use a Global Scanner object or pass as a parameter to do the input.

```
import java.util.Scanner;

public class Test {
   //Global declaration for Scanner
   public static Scanner sc = new Scanner(System.in);

   public static void main(String args[]) {
      int x;
      System.out.print("Enter x:");
      x = sc.nextInt();
   }

   public static void method1() {
      int y;
      System.out.print("Enter y:");
      y = sc.nextInt();
   }

   public static void method2(Scanner sc) {
      int y;
      System.out.print("Enter y:");
      y = sc.nextInt();
   }

}
```

***** END *****