

**Due:** Monday, April 23, 2012 by 11:59 PM

### Deliverables:

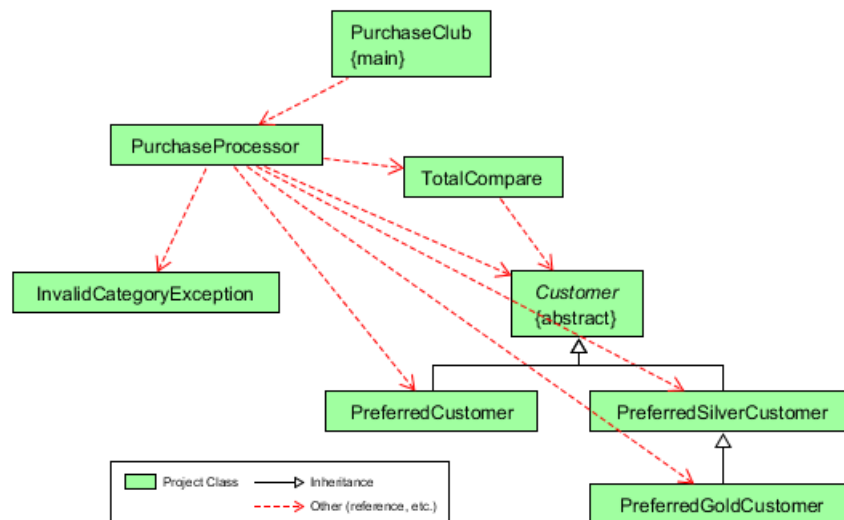
The following project files must be submitted by the due date and time specified above (see the Lab Guidelines for information on submitting project files). You should plan to start on the project not later than Wednesday in lab, and you should make sure that your files are successfully submitted to the “ungraded” assignment in Web-CAT during lab on Wednesday. **Projects sent via e-mail past the deadline at 11:59 PM will not be accepted without a university-approved excuse.**

Files to submit to Web-CAT:

- PurchaseClub.java (update with exception handling)
- PurchaseProcessor (update with exception handling)
- InvalidCategoryException (new exception class)
- TotalCompare (from Project 10)
- Customer.java (from Project 9)
- PreferredCustomer.java (from Project 9)
- PreferredSilverCustomer.java (from Project 9)
- PreferredGoldCustomer.java (from Project 9)

### Specifications

**Overview:** This week you will complete the third part of a three-part software project to process the monthly purchases made by members the “Purchase Club” online store. Part 3 of the project focuses on handling exceptions that are thrown as a result of erroneous input from the keyboard or from the data file. The main method in the PurchaseClub class, which reads in the file name from the keyboard, will need handle any IOException that results from attempting to open the file (e.g., if the file does not exist). The readCustomerFile method in PurchaseProcessor will need to handle exceptions that occur while processing the data file, including a new exception called InvalidCategoryException. The completed class hierarchy is shown in the UML class diagram below.



## Customer, PreferredCustomer, PreferredSilverCustomer, PreferredGoldCustomer, TotalCompare

**Requirements and Design:** There are no changes to these classes from Project 10.

### PurchaseProcessor.java

**Requirements:** The PurchaseProcessor class provides methods for reading in the data file and generating the monthly report. The readCustomerFile method should re-designed to handle exceptions in the data. A “good” line of data results in a new element added the customer ArrayList, and “bad” line of data results in the line + exception message being added the invalid input ArrayList. A new report method produces the Invalid Input Report.

**Design:** The readCustomerFile method from Project 10 should be redesigned to handle exceptions. PurchaseProcessor class has fields, a constructor, and methods as outlined below.

- **Fields:** no change from Project 10.
- **Constructor:** no change from Project 10.
- **Methods:** Only the readCustomerFile need to be reworked and generateInvalidReport needs to be added. See Project 10 for the full description of all of the methods in this class.
  - readCustomerFile has no return value and accepts the data file name as a String. If an exception occurs when attempting to open the data file, it should be ignored in this method so that it can be handled in the calling method (i.e., main). If a line from the file is processed successfully, a customer object of the appropriate category is added to the customer ArrayList. However, when an exception occurs as a result from erroneous data in a line from the file, it should be caught and handled as follows. The exception message should be concatenated to the line and then the resulting String should be added to the invalid input ArrayList<String> field in the class. The two exceptions that should be caught in this method are the InvalidCategoryException (described below) and the NumberFormatException. Note that the InvalidCategoryException must be explicitly thrown by your code if the category is not 1, 2, or 3; whereas, the NumberFormatException will be thrown automatically if the item scanned in the line from the file is not a double when Double.parseDouble expects it to be a double.
  - generateInvalidInputReport is the fourth report method. It has no return value and processes the invalid input ArrayList<String> to produce the Invalid Input Report as shown below in the example output.

**Code and Test:** See examples of exception handling in the text and the lecture notes. Download purchase\_data2.txt from the lab web page to test your program.

### InvalidCategoryException.java

**Requirements and Design:** The InvalidCategoryException class defines a new subclass of Exception. The constructor invokes the super constructor with the message: "\*\*\* invalid category \*\*\*". See examples of creating user defined in from text and lecture notes.

**PurchaseClub.java**

**Requirements:** The PurchaseClub class contains the main method for running the program.

**Design:** The Preferred class has fields, constructor, and methods as outlined below.

(1) **Fields:** none

(2) **Constructor:** none.

(3) **Methods:** The methods in Customer are inherited; the following methods need to be added:

- o `main` reads in the file name from the user the uses the method in the PurchaseProcessor class to read in the data file and generate the four reports as shown in the example output beginning on the next page. When an IOException is thrown in the readCustomerFile method in the PurchaseProcessor class, the catch clause in main should print a message (“\*\*\* Attempted to read file: ” along with the exception’s message) and then loop back and request the file name again. For example, if the user entered “nofile.txt” and this file didn’t exist, the exchange would look like the following.

```
Enter File Name: nofile.txt
*** Attempted to read file: myfile.txt (The system cannot find the file specified)
Enter File Name:
```

**Code and Test:** See examples of exception handling in the text and the lecture notes. Download purchase\_data2.txt from the lab web page to test your program.

**Example Output**

```
----jGRASP exec: java PurchaseClub

Enter File Name: nofile.txt
*** Attempted to read file: nofile.txt (The system cannot find the file specified)
Enter File Name: purchase_data2.txt
-----
Monthly Purchase Club Report
-----

Preferred Customer
AcctNo/Name: 10001 Smith, Sam
Subtotal: $548.00
Tax: $43.84
Total: $591.84
Award Points: 548

Preferred Silver Customer
AcctNo/Name: 10005 Smith, Jackie
Subtotal: $351.01
Tax: $28.08
Total: $379.09
Award Points: 3510
(includes 0.05 discount rate applied to Subtotal)

Preferred Gold Customer
AcctNo/Name: 10006 Williams, Jo
Subtotal: $1,273.29
Tax: $101.86
Total: $1,375.15
Award Points: 26460
(includes 0.15 discount rate applied to Subtotal)
(includes 1000 bonus points added to Award Points)

Preferred Customer
AcctNo/Name: 10007 Brown, Kit
Subtotal: $598.00
Tax: $47.84
Total: $645.84
Award Points: 598

-----
Monthly Purchase Club Report (by Name)
-----

Preferred Customer
AcctNo/Name: 10007 Brown, Kit
Subtotal: $598.00
Tax: $47.84
Total: $645.84
Award Points: 598

Preferred Silver Customer
AcctNo/Name: 10005 Smith, Jackie
Subtotal: $351.01
Tax: $28.08
Total: $379.09
Award Points: 3510
(includes 0.05 discount rate applied to Subtotal)
```

Preferred Customer  
AcctNo/Name: 10001 Smith, Sam  
Subtotal: \$548.00  
Tax: \$43.84  
Total: \$591.84  
Award Points: 548

Preferred Gold Customer  
AcctNo/Name: 10006 Williams, Jo  
Subtotal: \$1,273.29  
Tax: \$101.86  
Total: \$1,375.15  
Award Points: 26460  
(includes 0.15 discount rate applied to Subtotal)  
(includes 1000 bonus points added to Award Points)

-----  
Monthly Purchase Club Report (by Total)  
-----

Preferred Gold Customer  
AcctNo/Name: 10006 Williams, Jo  
Subtotal: \$1,273.29  
Tax: \$101.86  
Total: \$1,375.15  
Award Points: 26460  
(includes 0.15 discount rate applied to Subtotal)  
(includes 1000 bonus points added to Award Points)

Preferred Customer  
AcctNo/Name: 10007 Brown, Kit  
Subtotal: \$598.00  
Tax: \$47.84  
Total: \$645.84  
Award Points: 598

Preferred Customer  
AcctNo/Name: 10001 Smith, Sam  
Subtotal: \$548.00  
Tax: \$43.84  
Total: \$591.84  
Award Points: 548

Preferred Silver Customer  
AcctNo/Name: 10005 Smith, Jackie  
Subtotal: \$351.01  
Tax: \$28.08  
Total: \$379.09  
Award Points: 3510  
(includes 0.05 discount rate applied to Subtotal)

-----  
Invalid Input Report  
-----

0 10002 Jones Pat 34.5 100.0 63.50 300.0 \*\*\* invalid category \*\*\*  
3 10003 King Kelly 34.5 100.0 63.50 \$300.0 \*\*\* invalid purchase amount \*\*\*  
3 10004 Jenkins Jordan 34.5 100.a 63.50 300.0 100.0 \*\*\* invalid purchase amount \*\*\*

----jGRASP: operation complete.