OOD1

class newAndUsedStuffStore()

```
attributes
a list inventory. A list of item objects.

services
add(), does not output anything(void). Adds a new item object to the
inventory list.
findTitle(). Returns an item object corresponding to inputted title.
findAuthor(). Returns an item object corresponding to an inputted name.
display(), does not output anything (void). Displays either the entire
inventory, the item(s) that correspond to a given title, or the
books that correspond to a given author.
sell(), does not output anything (void). Removes all item objects
correspondng to a given title or author from the inventory
ArrayList.
```

class item()

```
attributes
String title. The title of the item object.
double price. The numerical price of the item object.
String condition. The state of the item object : new or used.

services
thatsMyTitle(). Returns true/false, a boolean, if item object's
title matches ser inputted searchTitle.
String toString(). Returns a human-readable string representation of
the item object.
```

class disc() extends item()

```
attributes
int releaseYear. The numerical year that the item object was released.

services
String toString. Returns a human-readable string representation of the disc.
```

class DVD() extends disc()

```
attributes
String studio. The studio where the DVD was produced.

services
String toString(). Returns a human-readable string representation of the DVD.
        Uses @Override.
```

class CD() extends disc()

```
    attributes
String band. The name of the band whose music is on the CD.

    services
    String toString. Returns a human-readable string representation of the CD.
            Uses @Override.
```

class book() extends item()

```
    attributes
    String author. The name of the author of the book.

    services
thatsMyAuthor(). Returns true/flase, a boolean, whether the user
    input authorName is the author of this Book.
    String toString(). Returns a human-readable string representation of the book.
```

class printBook() extends book()

```
    attributes
    String genre. The genre of the book : sci-fi or other.

    services
    String toString(). Returns a human-readable string representation of the printBook.
            Uses @Override. May use super().
```

class audioBook() extends book()

```
    attributes
    String reader. The reader of the audioBook.

    services
    String toString(). Returns a human-readable string representation of the audioBook.
            Uses @Override. May use super().
```

OOD2

This program will contain a superclass Item() that will be extended by two other abstract subclasses, Disc and Book. Two concrete classes will extend Disc, and two other concrete classes will extended Book. Main methods that Patrick will use are in NewAndUsedStuffStore class.

class NewAndUsedStuffStore()

```
imports java.util.List and java.util.ArrayList

    attributes
    ArrayList inventory. An ArrayList of item objects.

    services
    -> void add()
    purpose: Adds a new item object to the inventory ArrayList.
    input: the object to add, item
    results: updated list of item objects
    -> Item findTitle(String title)
    purpose: find an Item with a given title
    input: title to look for, a String
    returns: Item object corresponding to inputted title, null if not found
    -> Item findAuthor(String name)
    purpose: find an Item, specifically a Book, with a given author
    input: author name to look for, a String
    returns: an Item/Book object corresponding to inputted author name,
        null if not found
    -> void display()
    purpose: Displays the ArrayList inventory
    input: utilizes command line argument stating the way the inventory
        wants to be displayed as
    returns: a display of either the entire inventory, the item(s) that correspond to a
                given title, or the books that correspond to a given author.
    -> void sell(String s)
    purpose: removes all item objects correspondng to a given title or author from
        the inventory ArrayList.
    input: String s, the title or the author of the Item/s that will be removed
    returns: updated list of Item objects
```

abstract superclass Item()

```
    attributes
    String title. The title of the item object.
    double price. The numerical price of the item object.
    String condition. The state of the item object : new or used.

    services
    -> boolean thatsMyTitle(String searchTitle).
    purpose: checks if this Item has a given title
    input: String of the title we are looking for, searchTitle
```

```
    returns: true if this item's title matches searchTitle.
    -> String toString()
    purpose: produces a human-readable string representation of
        the Item object.
    input: just the Item object
    returns: String representation
```

class Disc() extends Item()

```
    attributes
    int releaseYear. The numerical year that the item object was released.

    services
    -> String toString(), @Override
            purpose: produces a human-readable string representation of
                    the Disc object.
            input: just the Disc object
            returns: String representation
```

class DVD() extends Disc()

```
    attributes
    String studio. The studio where the DVD was produced.

    services
    -> String toString(), @Override
            purpose: produces a human-readable string representation of
                    the DVD object.
            input: just the DVD object
            returns: String representation
```

class CD() extends Disc()

```
    attributes
    String band. The name of the band whose music is on the CD.

    services
-> String toString(), @Override
            purpose: produces a human-readable string representation of
                    the CD object.
            input: just the CD object
```

```
            returns: String representation
```

class Book() extends Item()

```
    attributes
    String author. The name of the author of the book.

    services
    -> boolean thatsMyAuthor(String authorName) -
    purpose: checks if this Book was written by a given author
            input: String of the author we are looking for, authorName
            returns: true if this Books's author matches authorName
-> String toString(), @Override
            purpose: produces a human-readable string representation of
                    the Book object.
            input: just the Book object
            returns: String representation
```

class PrintBook() extends Book()

```
    attributes
    String genre. The genre of the book : "sci-fi" or "other".

    services
-> String toString(), @Override
            purpose: produces a human-readable string representation of
                    the PrintBook object.
            input: just the Book object
            returns: String representation
```

class AudioBook() extends Book()

```
    attributes
    String reader. The reader of the audioBook.

    services
-> String toString(), @Override
            purpose: produces a human-readable string representation of
                    the AudioBook object.
            input: just the AudioBook object
            returns: String representation
```