Assignment 7
Due Thursday March 18

Your friend Electronic Eddie has decided to open a business that rents movies and games. Unfortunately, Eddie has very little startup money and cannot afford to buy the latest software package to manage his inventory. As a programmer without peer, you have come to Eddie's rescue and have volunteered to write a system for his business.

Your first step is to design a class hierarchy which includes the following classes:

Item , an abstract class, that implements Comparable with the following attributes:

a title (String) (use dashes to separate words, eg. Knives-Out
rental price – a fee (double)
status (int, 1 if in stock, 0 if currently rented)
the current renter's name (String)
use a comma to separate last and first (lastname,firstname)

The methods of Item are the standard getter and setter methods as well as the method

```
public String toString()  // returns the data with labels

public int  compareTo(Object o)
                        should be based on the title
```

Game (extends Item) with the following additional attribute:
age level (an integer from 3 to 16, 16 signifies 16+)

Movie (extends Item) with the following additional attributes:

playing time in minutes (int)
rating : G, PG, PG13, or R (String)
format: 'B' for BlueRay cassette, 'D' for DVD (char)

Each class implements to String () that returns all the data of the invoking object .

Once you have implemented the above classes, you should design and implement a class that utilizes the Item hierarchy, call it EddiesStore.

Your system should be menu-driven and include the following options:

a. Check out an item.
   Your system should query the user title of the item and the renter's name. If the item is already checked out your system should say so. This should update the information. It is now out.

b. Check in an item.
   Your system should ask for the title of the item. If it is already checked in, indicate that. Update the information

c. Search for an item by title.
   Since the rentals are not sorted by title, you might use sequential
   If the title is not in the inventory, say so.

e. Display the entire inventory, sorted by title.

f. Add a new item to the inventory.

g. Display the menu

h. Exit

When the program begins, the program should

- obtain data for each item from a file, data,txt, and
- store the data in an array of Item.
- Use SelectionSort.sort(…). It is on the class website under "Other Stuff"

You should create a file, say *data.txt* with 15 items in it -- 5 games and 10 movies. Use one line in a file for each item. The first entry on each line should be either G or M. Do not include spaces in the title of the movie or the renter's name. A typical entry
for movies has the format

M title running-time fee rating **B or D** **1 or 0(Status)** renter

For Games

G title fee **1 or 0(status)** age-level renter

For example:

```
M  Knives-Out  123  2.00 R B 1  Gooche,Agnes
G  Borderlands-3   4.50   1   17    Rose,David
```

.2.

Design an **interface Playgame** with two methods:

public void play(); and

public int getBet();

Remember the methods of an interface have no code.

Now make three classes Poker, Blackjack, and Craps that implement Playgame.

Each class has a variable bet (int) and a one argument constructor that sets the bet.

The getBet() method returns the bet

The play() method does not really play the game it

1. prints a statement like "Playing poker" (rather than play the real game)
2. reports win or loss --
   by getting a random int (0 for lose, 1 for win) and the amount won or lost -- that is the bet

Now make another class TestGame.

In this class
declare a single variable:
Playgame game -- **so that you are upcasting to an interface**.

This class asks the user which game he/she would like to play

The user enters "P", "B" or "C"

Depending on user input a specific game is created and assigned to game.

A bet is accepted and the game is played

The user can continually play games until he/she enters a choice

other than P, B or C. When that happens the program ends.

It does not matter which game the user chooses when the program runs.

When your program executes game.play(), the correct play() method will be chosen at runtime.

That is polymorphism. You do not need a downcast. You do not need instanceof

Here is typical output

```
Enter P,B or C -- anything else to quit: C
Enter Bet: 23
Playing Craps
You win 23

Enter P,B or C -- anything else to quit: B
Enter Bet:12
Playing BlackJack
You win 12

Enter P,B or C -- anything else to quit: P
Enter Bet: 4
Playing Poker
You lose 4

Enter P,B or C -- anything else to quit: X
```

3. Exceptions

Design a class ArrayTest with private data an array of 10 Strings.

You should include a one-argument constructor that accepts a String, which is the name of a file.

The constructor should read 10 names into the file.

The constructor should include a try-catch pair that checks whether the file name is valid. If not, the catch block should issue a message and terminate the program

You should have a method

String find(int index)

that accepts an array index and returns the name at position index.

Include a try-catch to catch an index which is out of bounds.

The catch block will issue a message but will not exit the program

Test your program with the following main method that

1. Asks the user for a file name
2. continually asks for an index (until 999 is entered) and prints the name at position index.

Here are the data for your file

```
Bart
Homer
Lisa
Krusty
Marge
Itchy
Scratchy
Selma
Flanders
Maggie
```

Here are two program runs. The first with a bad file name the second with bad indices.

```
Enter file name: dumbFile.txt
File not found
```

```
Enter file name: names.txt
Enter index 3
3: Krusty

Enter index 12
Index out of bounds 12
12: bad input

Enter index 4
4: Marge

Enter index 88
Index out of bounds 88
88: bad input

Enter index 9
9: Maggie

Enter index 999
```