

# COSC 201 – Assignment #1

## Fall 12

**Objective:** Write a program that will take in a text filename via the command line. That file will contain a list of games that will need to be placed in a database based on the type of game, one per line, formatted **title, genre, number of players, genre specific information, age recommendation, estimated time, parent specific information 1, parent specific information 2**. For your maximum flexibility in expansion, there should be an abstract class called **Game**, and three parent classes (extending Game) called **Board, Card, and Video**. All specific games will be implemented as classes that extend each of these and will need to include at least the specific field noted below and two specific methods of your own devising.

### **Implementation Details:**

**Game** should include fields for the following: number of players, title, genre, age recommendation, estimated time, with appropriate getters and setters. It will also need abstract methods for toString (return of String, assists in printing), isWin (return of Boolean, tells whether or not the current player has won), and anything else needed.

**Card** will include fields for number of cards. There will be no second parent specific information for **Card**.

**Board** will include fields for size of board and number of pieces.

**Video** will include fields for console and installation size.

### **Genre specific information:**

Type (Parent in italics)	Examples	Required Field
<i>Card Games:</i>		
Banking	Blackjack, Baccarat	Bet
Trick	Euchre, Spades	Number of tricks
Patience	Klondike, Pyramid	Remaining cards
<i>Board Games:</i>		
Tile	Mahjong, Dominoes	Number of tiles
Capture	Chess, Checkers	Current board state
Strategy	Risk, Axis and Allies	Number of pieces remaining
<i>Video Games:</i>		
FPS	Halo, Bioshock	Number of angry 14 year olds
RPG	Final Fantasy, Skyrim	Number of strange characters
Sport	FIFA, Madden	Current team

Each genre specific class will need to implement the isWin and toString abstract method from the base class. Note that since you'll be doing the implementation of isWin in the genre classes, you'll need to do something with isWin and toString in the parent classes (Board, Card, Video).

Genre specific classes will need to create fields for the above information, as well as implement

two extra methods of your own devising that are relevant to the class in question. Feel free to use any user input you'd like in these methods.

The abstract class **Game** should include anything additional required due to your testing routines (see below). In all cases, additional fields and methods should be relevant and should do something significant.

Your driver class will hold all of your ArrayLists (see below) and any methods that you'll need to manipulate those ArrayLists. This is where your main method will be and where you'll do the file I/O and any printing to the console. Basically, all the other classes are just objects that you'll use in the driver class, storage objects if you will.

**Expectations:** Your code will need to be neat, concise, well documented and above all, correct (see Testing). All classes should have headers and each method should have comments describing the method's function. Any novel or possibly confusing code should be explained, as I do get confused and distracted easily.

**Testing:** Your driver should open the text file and add games to ArrayLists of each of the genre types (a total of 9 ArrayLists) – this represents your database. Your driver should then output the number of games in each category, the number of games in Board, Card, and Video and then ask the user for a genre name. Once a genre name is inputted, the driver should output an alphabetical sort of all of the appropriate games by title. I will be testing this with variously sized files, including an empty file, no file passed via command line and a file that does not exist. You will need to perform basic error handling, but you can assume that the file is properly formatted.

**Quick Note:** You should have a total of 14 classes (9 child classes, 3 parents, 1 abstract base class, 1 driver class). If you do not have at least 14 classes, then you have a question you should ask either the TA or me.

Grading rubric will be given out at least a week ahead of the due date. You may work in teams of two for this assignment.

**Learning Targets:** String manipulation, file I/O, inheritance, abstract classes

**DUE:** October 10<sup>th</sup>, 11:59 pm Eastern.