**CSD 3464 s2 Monday 8 am Class 2021F Programming Java SE**

**Total Assignment: Date Due: October 11**

**How to submit: via GitHub (Instructions will be provided)**

**Resources:**

<https://github.com/computationalknowledge/JavaProgramming2021F>

Java Visual Learning Roadmap: (see Moodle for LINK)

PDF Book: Learn Java The Hard Way (see Moodle for LINK)

Google Doc with Design Notes:

<https://docs.google.com/document/d/1WbuanXXoxBfT9Vu9oOi76QR5uMIEsYAMO9fUO26D03k/edit?usp=sharing>

**Assignment 1: 3% of Final Grade**

In this assignment you will write a computer program to let people play the Card Game of Fizbin, which is played by the Players on Beta Antares 4.

[**https://www.catsatcards.com/Games/Fizzbin.htm**](https://www.catsatcards.com/Games/Fizzbin.htm)

**The Game of Fizbin** [**https://youtu.be/\_DeIExLcURQ**](https://youtu.be/_DeIExLcURQ)

Chart, map, scatter chart

Description automatically generated

**A picture containing text, person, person, indoor

Description automatically generated**

For MS 1, Instructor will provide a code framework.

You must write a CARD class that works with my code.

**Learning Outcomes**

Constructing a well-formed Java Object-oriented Program: community of objects

Implementing an Algorithm in a Java Program based on a set of provided instructions

Working with Text Strings and Substrings

Working with Data Structures

Develop Software Architecture Skills

**Implement High Cohesion** / **Low Coupling Design Pattern**

Connect your Objects via the 2 connection mechanisms of:

Composition

Inheritance

Object Oriented Analysis and Design for FIZBIN

**Start by Asking: What TYPES exist in my SUD System Under Design that I need to MODEL in software?**

**What else do you need to make your FIZBIN game work?**

**The Knowledge Principle of OO Design:**

**Create TYPES that own the Data and Behaviors of delivering the requirements of the Algorithm you are implementing.**

**There will be Milestone 1 due OCTOBER 4 (This is just to make sure that you have version 1 of your Fizbin Application ready to work on for next class)**

**For Milestone 1:**

Design the CARD Class for your game.

What data and behaviors does a card need to have?

Remember that we said:

With PP 3 and earlier languages (like C), we had to create complicated chains of IMPERATIVE code to control how the program ran.

In PP 5 – the OBJECT ORIENTED Paradigm – we let the objects be in charge of local self control of how the elements (the objects) of the program interact together to deliver the Algorithm (or business requirements).

Graphical user interface, text, application, email

Description automatically generatedHow will you give methods to your CARD class to let the cards be in charge of delivering the Rules of The Game? (It is YOUR Game – you must design these rules! It will be more meaningful to you if you invent the rules so you will understand where your algorithm came from).

https://boardgamegeek.com/thread/137606/fizzbin-rules

Table

Description automatically generated

The Rules  
  
Players: Four.   
  
Setup: Six cards are dealt out to all players.  
  
Turns: The player on the dealer's left gets the first turn, and then play proceeds counter-clockwise around the table. There is one discard pile for each player (placed in front of the player). To play, you may draw from any one of the four discard piles, or you may draw from the stack of remaining cards.  
  
If the card you draw is a king or a two, you may draw again.

Discard Cards: If you draw a queen or a four or a king or a two, you must discard it to your pile.  
  
After you have drawn your card(s) and you have discarded the queen or four or king or two (if necessary), you must discard another card to any one of the **four piles.**  
  
If you drew from the stack, you may discard that card. If you drew from a discard pile, you must keep that card (though you can discard it on your next turn if you wish).  
  
Play continues until the stack is exhausted. At that time, the cards of all players are revealed and compared by the following scale:  
  
**Royal Fizzbin:**  
  
The best hand. It consists of a king and a two or a queen and a four, and any two pairs of aces, jacks, or sevens.  
  
**Full Fizzbin:**  
  
Second best. It is a king and a two (or a queen and a four at night), and any one pair of aces, jacks, or sevens.  
  
**Three-Quarters Fizzbin:**  
  
Next best. It is a three, five, six, eight, nine and ten, all red, except at night when they must be all black.  
  
**Half Fizzbin:**  
  
Lowest hand that can win. Consists of any pair of sevens, jacks or aces.  
  
**Shronk:**  
  
The worst hand of all. It is three or four of a kind. Automatically loses, no matter what anybody else has.  
  
Players are encouraged to add new rules to Fizzbin and create their own variations of the game.

**What to do:**

Like any computer application:

You are contemplating 2 questions:

|  |
| --- |
| How to handle the Flow of Control |
| How to store and manage the Data and behaviors of the elements (TYPES) of our Application:  What entities do we have in this business domain that we must model in Software:   * + Cards   + Players   + Can you think of the other entities we need? We will learn to use a special tool called Unified Modelling Language to help ourselves figure this out. |
| Set up a simple community of objects to let Players play FIZBIN with the Computer to simulate your Card Game in Software |
| You are going to set up your Set of Cards for FIZIN  Construct a Fizbin Card Deck **Class**  Make object references for each card |
| Interact with Players via the Command Prompt (In future iterations of our game, we will add new functionality to provide a GUI Graphical User Interface. |
| How to handle the Data:  **[See the Video on the Lions and Tigers simulation for ideas.]**  For now: We will use simple Java Data Structures such as Array List. (In the next assignment, we will implement reading / writing files, and using a SQLite Database).  We store the Object References to Players and Cards in the Data Structures. |

The Project Plan:

|  |  |
| --- | --- |
| Development Iteration | Deliverables: |
| Milestone 1  This will be counted as Assignment 1 | Let’s design our Deck of Cards:  Business Rules that are implementing:  Let’s develop a six card deck:  **For Milestone 1, you must design a CARD class and a Collection Class for the Deck**.  I will provide a starter code template.  You must write a Card class that works with my code. |
| Milestone 2  This will be counted as Assignment 2 | Make a UML Dia  gram to plan the Design of our Game’s Architecture.  Design the Game Controller. |
| Milestone 3 | Design the Player Class and the Collection Class to store your Players in. |

**Let’s design our Deck of Cards:**

Need one class to be the ADT (Abstract Data Type for a CARD)

What is the other data structure in addition to modeling a Card that we need?

We need a COLLECTION for ALL the cards: ArrayList

Business Rules that are implementing:

Let’s develop a six card deck:

Work Step Summary [3 Step Procedure for writing a software application]

(When something becomes a procedure it stops being a problem)

|  |  |
| --- | --- |
| Write classes to model your Data | What classes do we need to model for our FIZBIN Card game:  Card  Aggregation or Collection object to keep everything together  Player |
| Implement our Algorithms | This is done INSIDE the Object Methods  You write Methods to implement your Algorithms. |
|  | Your algorithm is delivered by methods inside the objects.  THIS IS WHY WE SAY THAT IN OO APPLICATIONS, we develop the application by controlling the flow of DATA.  In the older paradigms such as Imperative Programming ( C programming), we focused on controlling the Flow of Execution. |
| We need to write a CONTROLLER class to do the work choreography of pushing all the other objects around to deliver the Work Output of the Application.  Another responsibility of the CONTROLLER is to manage the data plumbing and wiring to get the right data to where it needs to be, in the right format and in the right way. |  |