

# CS 1302

## Summer 2016

### Project

For this project, you will be creating a graphical game of High Card. The project is worth 15% of your grade and the breakdown is:

- 5% for Status Update Presentation
- 75% for Code submitted for Final Delivery
- 20% for Final Presentation

You must pay attention to the requirements and ensure that your project meet each and all requirements.

#### **Rules of the game**

There is one deck of 52 cards (the pile of cards are face down). Each card has a suit and value. For example: the Ace of Spade has suit="Ace" and value="Spade". Aces are the highest and 2's are the lowest and the values go in the following order: Aces, Kings, Queens, Jacks, 10, 9, 8, 7, 6, 5, 4, 3, and 2. The Rank of suit goes from Spade (highest), Diamonds, Clubs, and Hearts.

A single game consists of 13 rounds. At the beginning of each round, the computer (dealer) is given the top card and is turned face-up. The human player (i.e., you) is given next card, which is also turned face-up. The player with the higher card is the winner. Play continues until no card remains.

#### **Gameplay and GUI requirements**

- At least 6 areas of GUI components in the MainArea
  - DeckArea, DealArea, PlayerArea, ComputerArea, RecordArea, OutcomeArea
- DeckArea: a deck of face down cards (hint, you only need one graphic of a face down card). Underneath the deck, there is a label that shows the number of cards that remain
  - At the beginning, the game begins with 52 cards
- DealArea
  - A "Deal" button

- When pressed, the top card is given to the ComputerArea and the next card is given to the PlayerArea
- ComputerArea and PlayerArea
  - Displays a face up card
- RecordArea
  - Displays the record of the current game played (i.e., the outcome of each round for the current game. Note, at max, there should only be 13 rounds)
- OutcomeArea
  - Displays the outcome of the round
- When 13 rounds have completed, a ContinueScreen is displayed
  - pop-up that behaves identical to the TitleScreen
    - displays the record from the previous game (this will be 13 rounds)
    - displays the record from the all games
    - Continue and Exit buttons

### **Coding Requirements**

- Inheritance is used
- The package keyword is used
- Polymorphism is used
- JavaFX is used (i.e., not awt or swing)
- Exceptions are implemented
  - when file is missing, displays an error on the ErrorScreen
    - creates an empty records.dat file
  - when img/{cardImage file} is missing, displays an error on the ErrorScreen
    - Cannot play game
- Read scores/records.dat file to display wins (first number) and losses (second number)
- Recursion is implemented
  - Writes outcome of run to records.dat file
- Program runs from Command Prompt in the following ways

- java {optional options} {NameofProgram}
  - Runs the program
- java {NameofProgram} –help
- javadoc –author {NameofProgram.java}

## Grading

### Status Update Presentation (5%)

5 minute informal presentation to demonstrate progress (10 pts)

### Final Delivery (75%)

Rules of the game meet requirements: 25 pts

Gameplay meets requirements: 25 pts

GUI requirements are met: 25 pts

Coding requirements are met: 75 pts

### Final Presentation (20%)

10 minute **professional** presentation (40 pts)

Topics to include

- UML or other class designs that illustrate relationship between classes
- MVC design illustration
- Flowchart illustration of when methods gets called
- Describe and explain your UI design
- Talk about something that was easy and why it was easy
- Talk about something that was difficult and how did you overcome it
- Brief Demonstration

Note: In your presentation, you must NOT show full code. You only need to show code samples or snippets.