### **Object-Oriented Project Part 3 - GROUP**

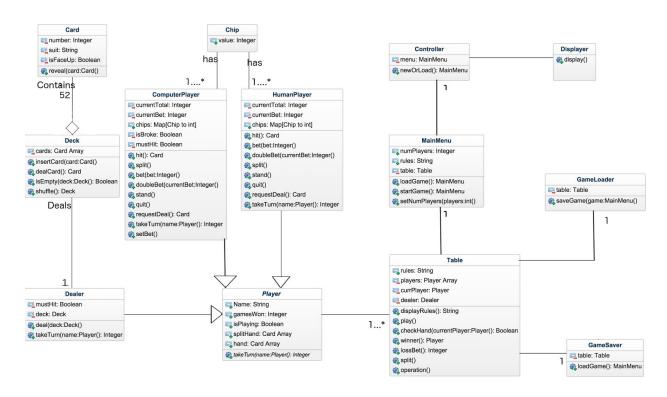
**Team:** Nolawee Mengist Shawn Polson

Theerarun Tubnonghee (Steve)

Jose Escobar

Title: Blackjack!

#### Class Diagram from Part 2:



# Refactor: (the highlights)

#### Chip:

Since the Chip class was only containing a single value attribute as an int, which is an example of an anti-pattern, we refactored it to "Chips." Chips is now a class that represents chips was a Map from a chip's value  $\rightarrow$  number of those chips, with associated methods like addChip or takeChip.

#### Design Patterns: State and Memento

State: We looked at phases of playing Blackjack, and we found that we can use State to handle the turns and phases: BettingPhase, PlayHandPhase, SplitHandPhase, and DealerHandPhase. The State will have a handle method called play().

Memento: We need to save the state of the Table, and we found that Memento can handle the saveGame and loadGame methods. GameOriginator will serve as the Originator, Memento will serve as getting/settingState, and our Database will serve as Caretaker.

## **New Class Diagram:**

