Module

Classes and Objects

Lesson

Classes and Objects

Lecture

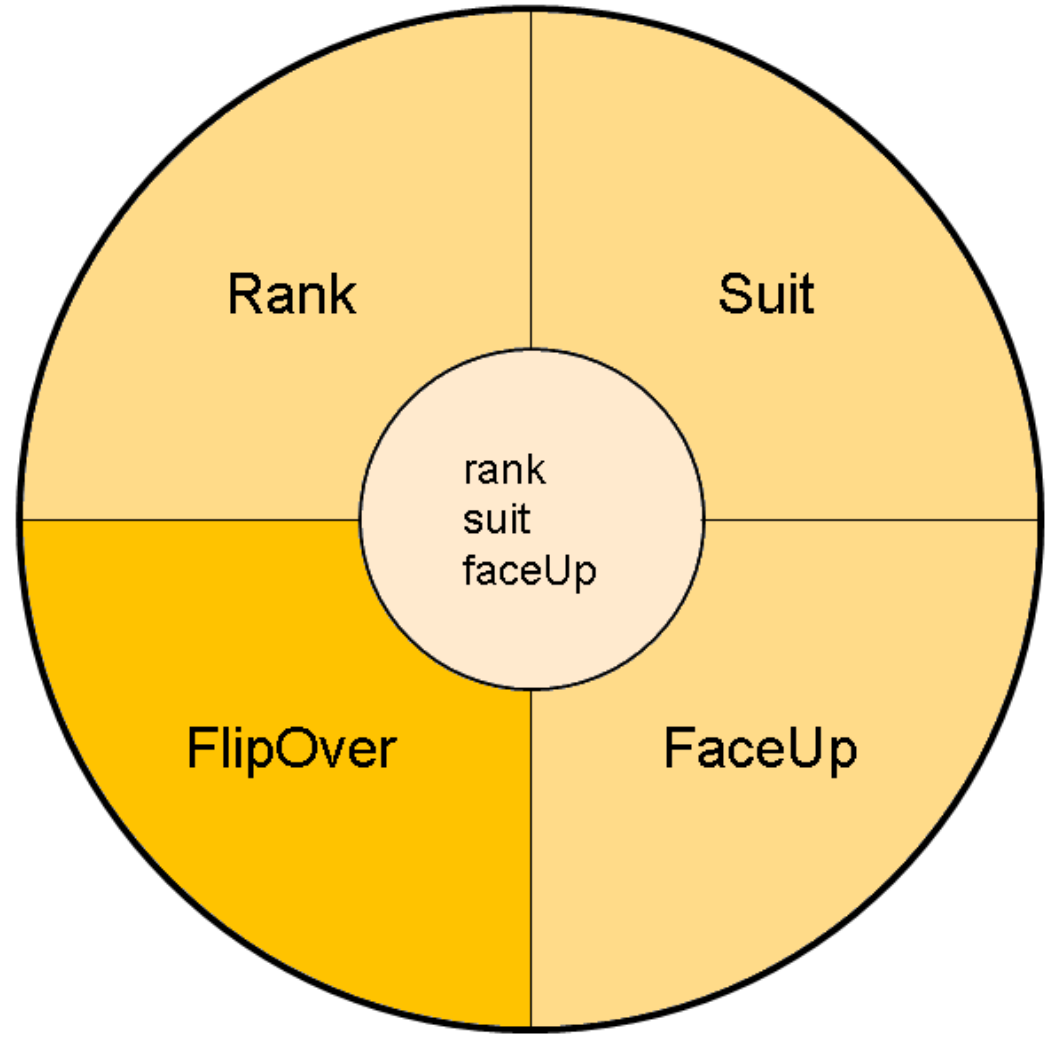
Introduction to Classes and Objects

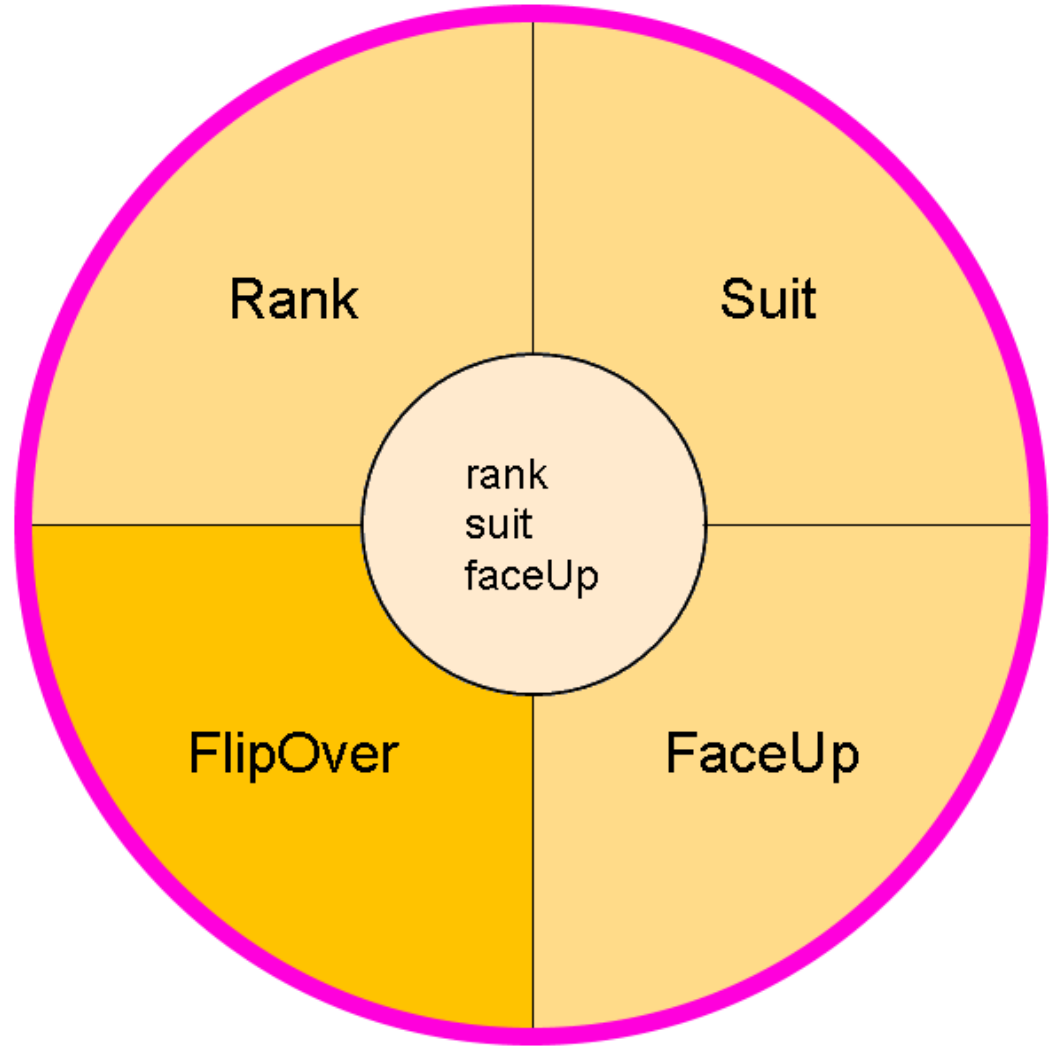
We’ll start this lecture by talking about objects

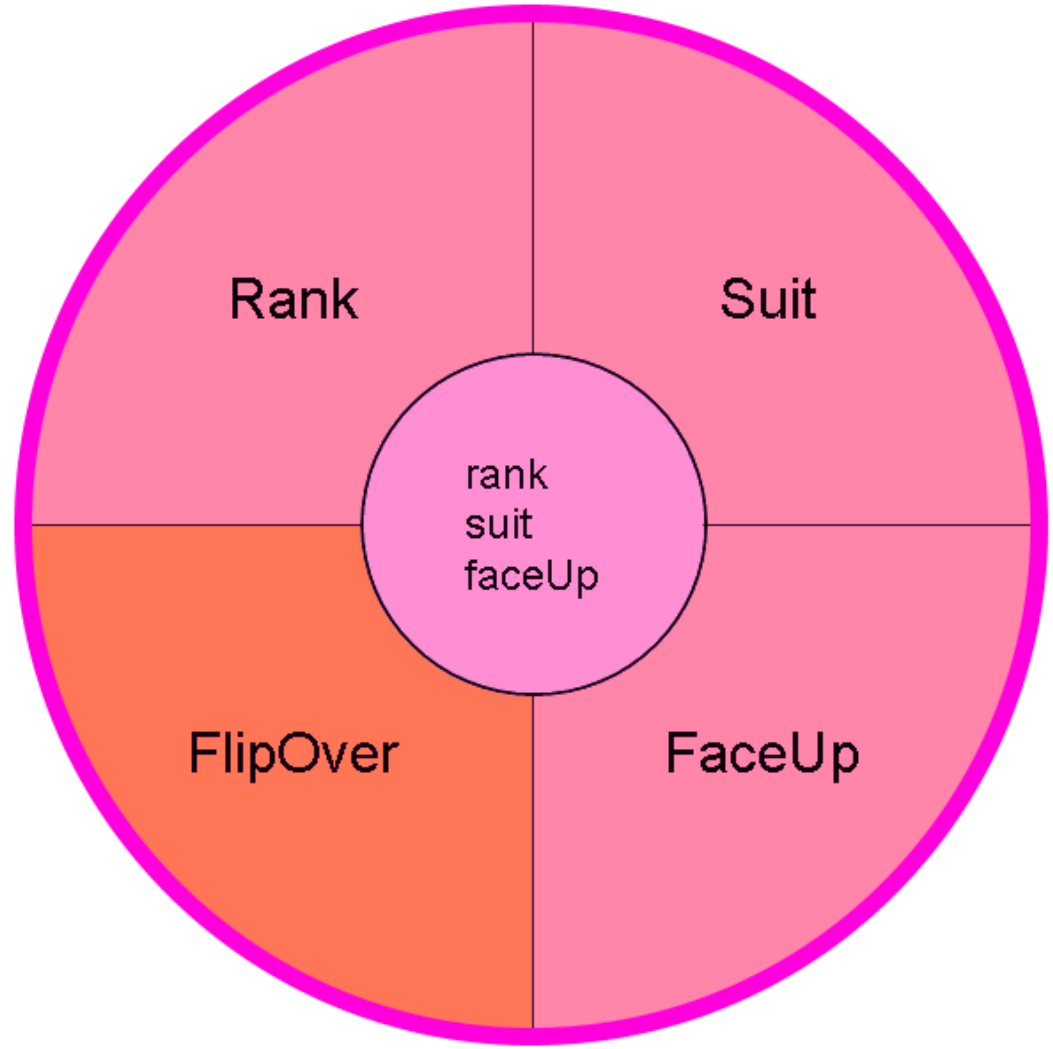
Before we start talking about the details, you can think of objects as software representations of tangible things (ships, asteroids, weapons, projectiles, etc.)

- State
  - Characteristics of the object
- Behavior
  - What we can do to the object
  - What we can tell the object to do to itself
- Identity
  - So we can distinguish one object from another
  - Memory address

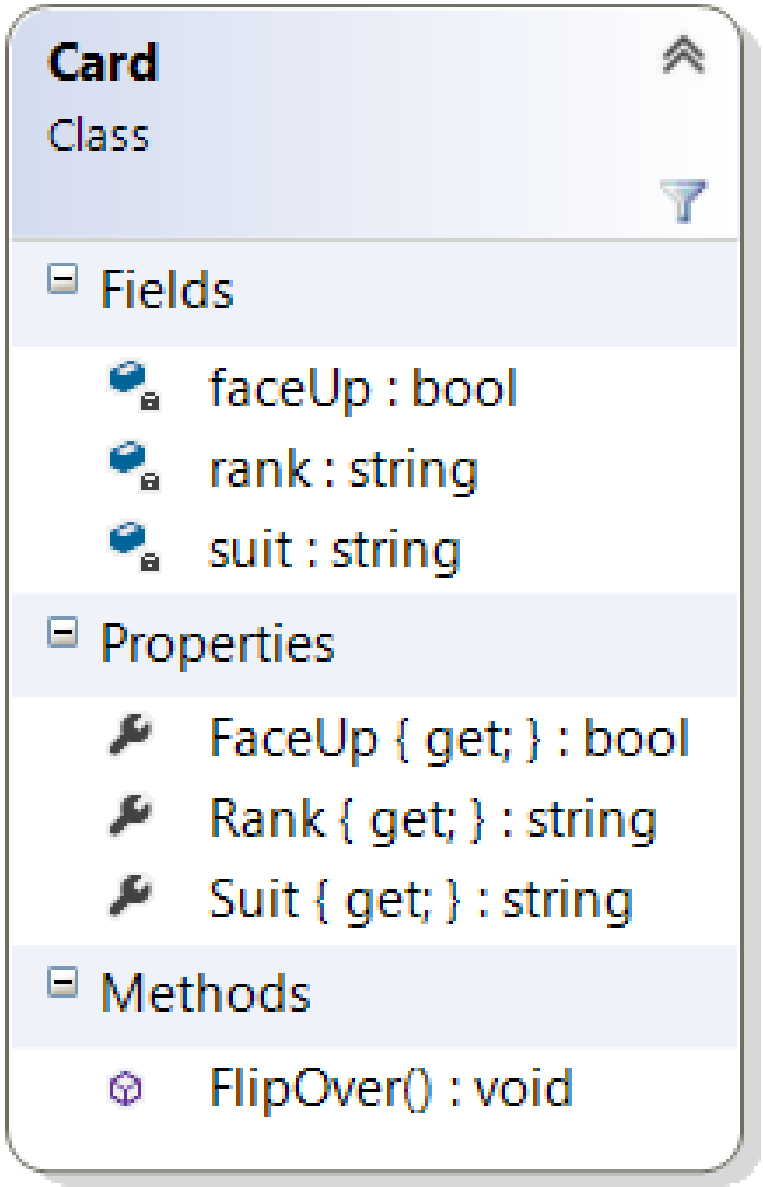
- Playing Card
  - State: rank, suit, face up or not
    - Stored in fields
    - Accessed through properties
  - Behavior: flip over
    - Accessed through methods
  - Identity: when we create a new card object (instantiation)


Rank
Suit
rank
suit
faceUp
FlipOver
FaceUp


Rank
Suit
rank
suit
faceUp
FlipOver
FaceUp

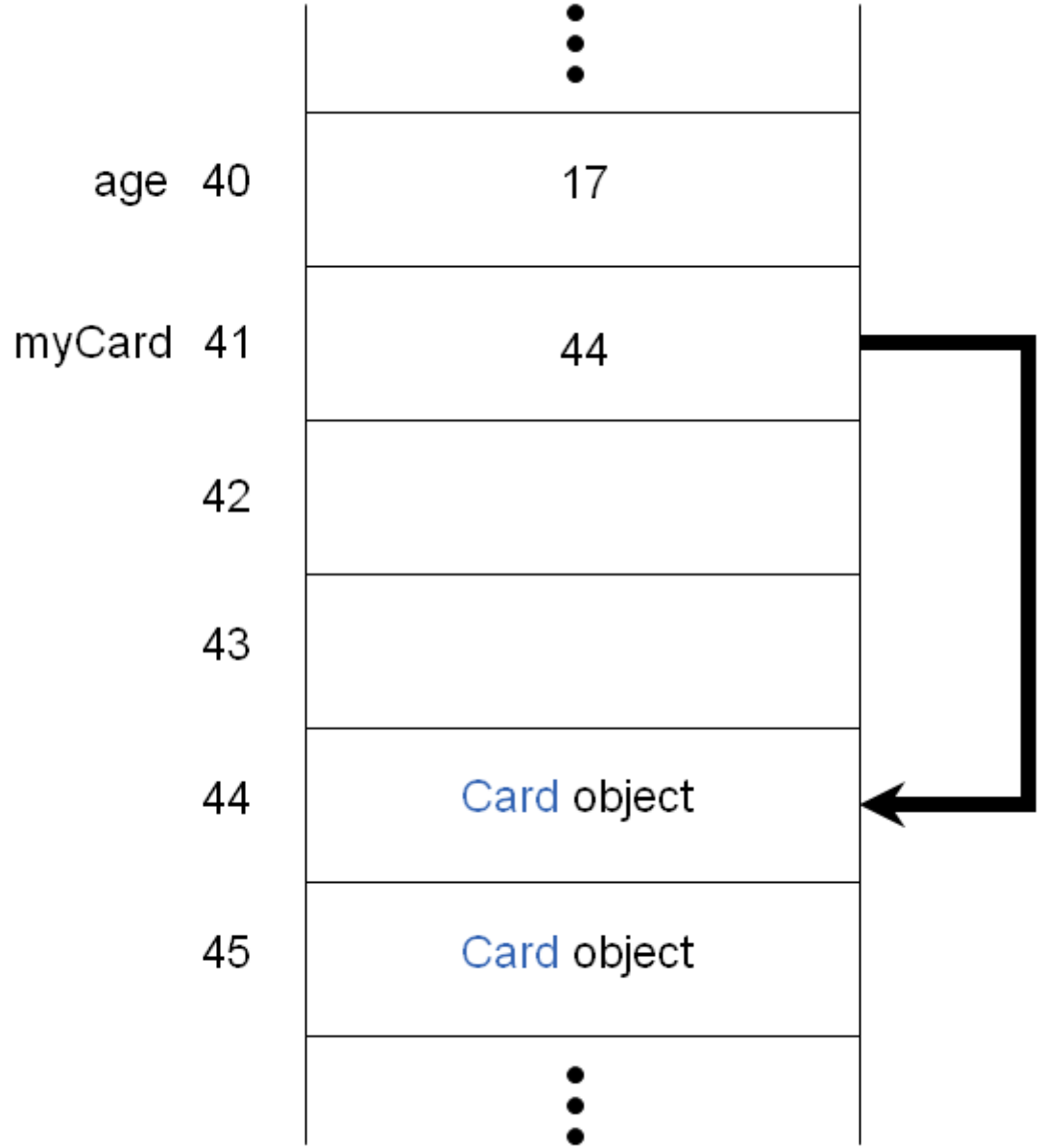

Rank
Suit
rank
suit
faceUp
FlipOver
FaceUp

- Class
  - Template for creating objects
  - Defines the fields, properties, and behavior of every object of the class
- Object
  - Actual instance of the class in memory
  - Each object stores its own state
    - Different card objects have different ranks and suits, for example


Card
Class
Fields
faceUp : bool
rank : string
suit : string
Properties
FaceUp { get; } : bool
Rank { get; } : string
Suit { get; } : string
Methods
FlipOver() : void

Before we finish off this lecture, note that classes are reference types, not value types

The ones and zeros at the memory location allocated to the variable aren’t the value of the variable, they’re a reference to the location in memory where the actual object is


age
40
17
myCard
41
44
42
43
44
Card object
45
Card object

- Recap
  - OO provides classes (reference types in C#) we can use to create objects
  - Objects interact with each other to implement our game