Module 4
Classes and Objects

Lecture
Intro to Classes and Objects

Module 4 Learning Objectives

Bloom Level	Number	Name	Description	Course Learning Objectives
3: Apply	1	Use Provided Classes	Develop a console application that uses provided classes	Basic OO Concepts,Basic Programming Concepts
6: Create	2	Design New Class	Design a new class	Basic OO Concepts

In this module, we'll talk about the foundational concepts in the Object-Oriented paradigm

The big idea: software can be implemented as a set of interacting objects

Examples of interacting entities in the game world

In-Lecture Quiz

Which of the following is NOT one of the things an object has

- A: State
- B: Country
- C: Behavior
- D: Identity

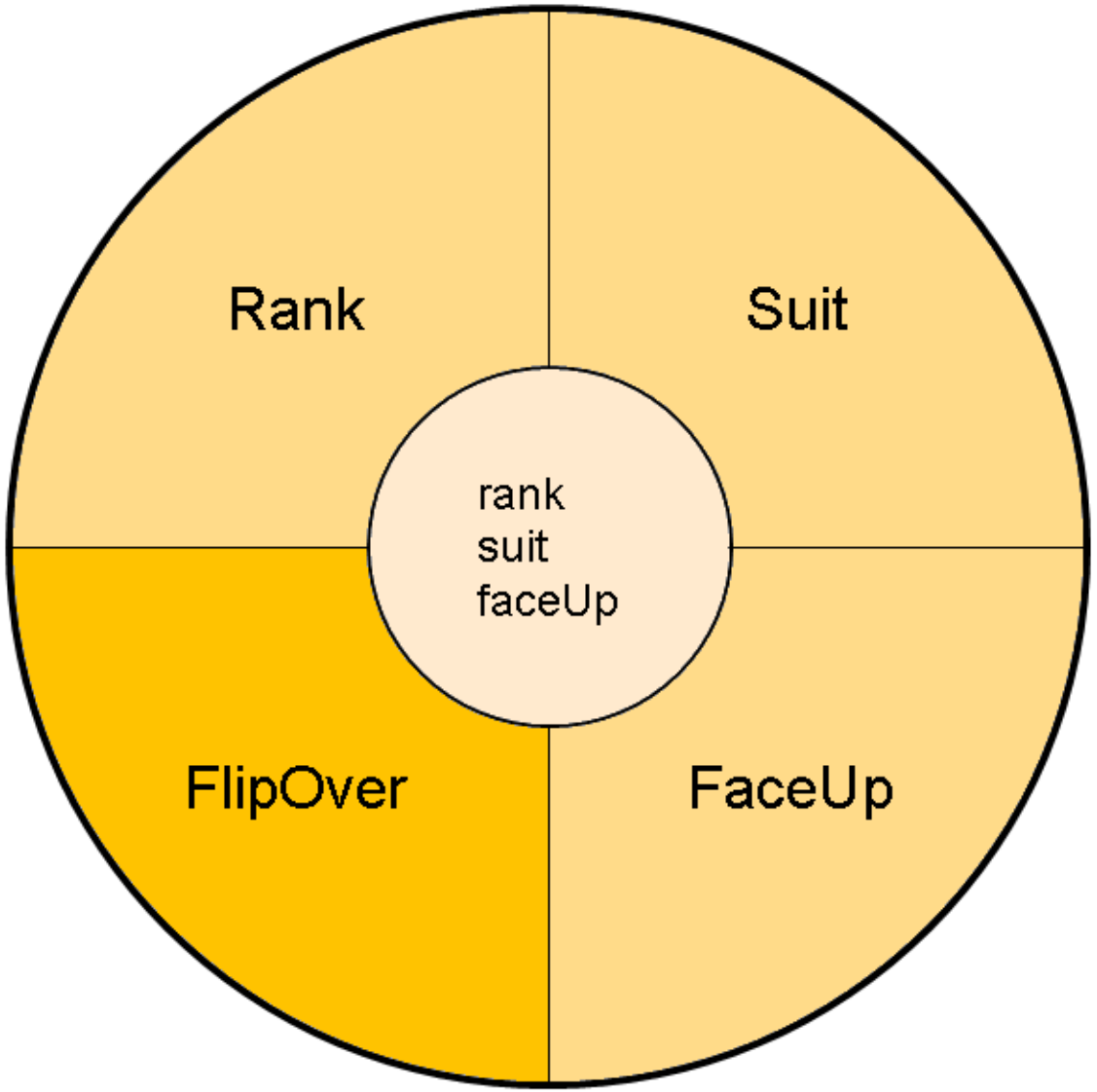
- State
  - Characteristics of the object
- Behavior
  - What we can do to the object
  - What we can tell the object to do to itself
- Identity
  - So we can distinguish one object from another
  - Memory address

- 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)

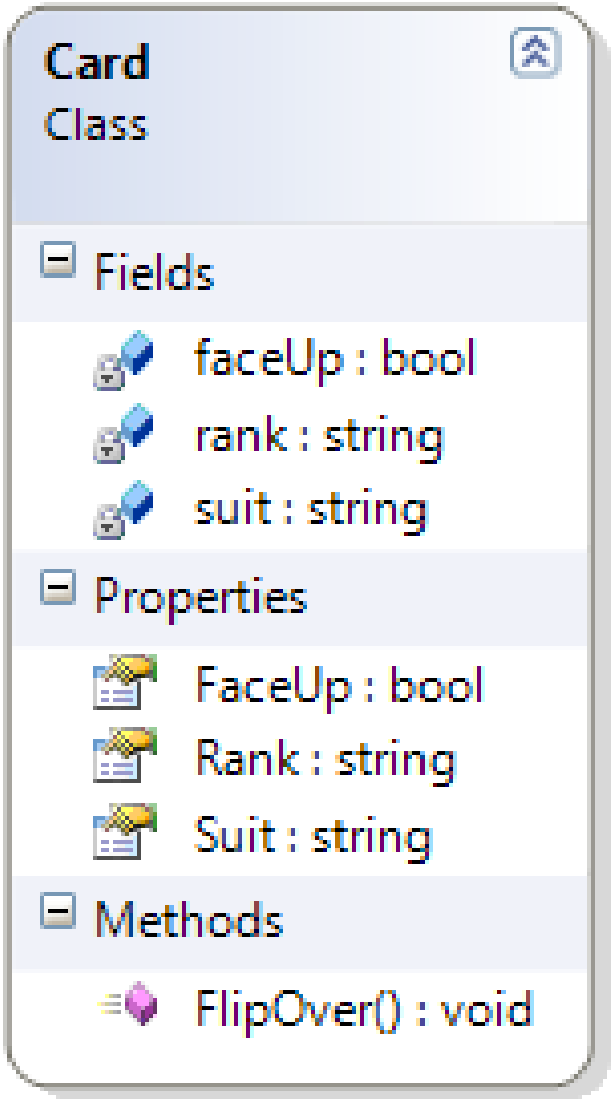
In-Lecture Quiz

We store the state of an object in

- A: meadows
- B: pastures
- C: fields
- D: pocketses


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 : bool
Rank : string
Suit : string
Methods
FlipOver() : void

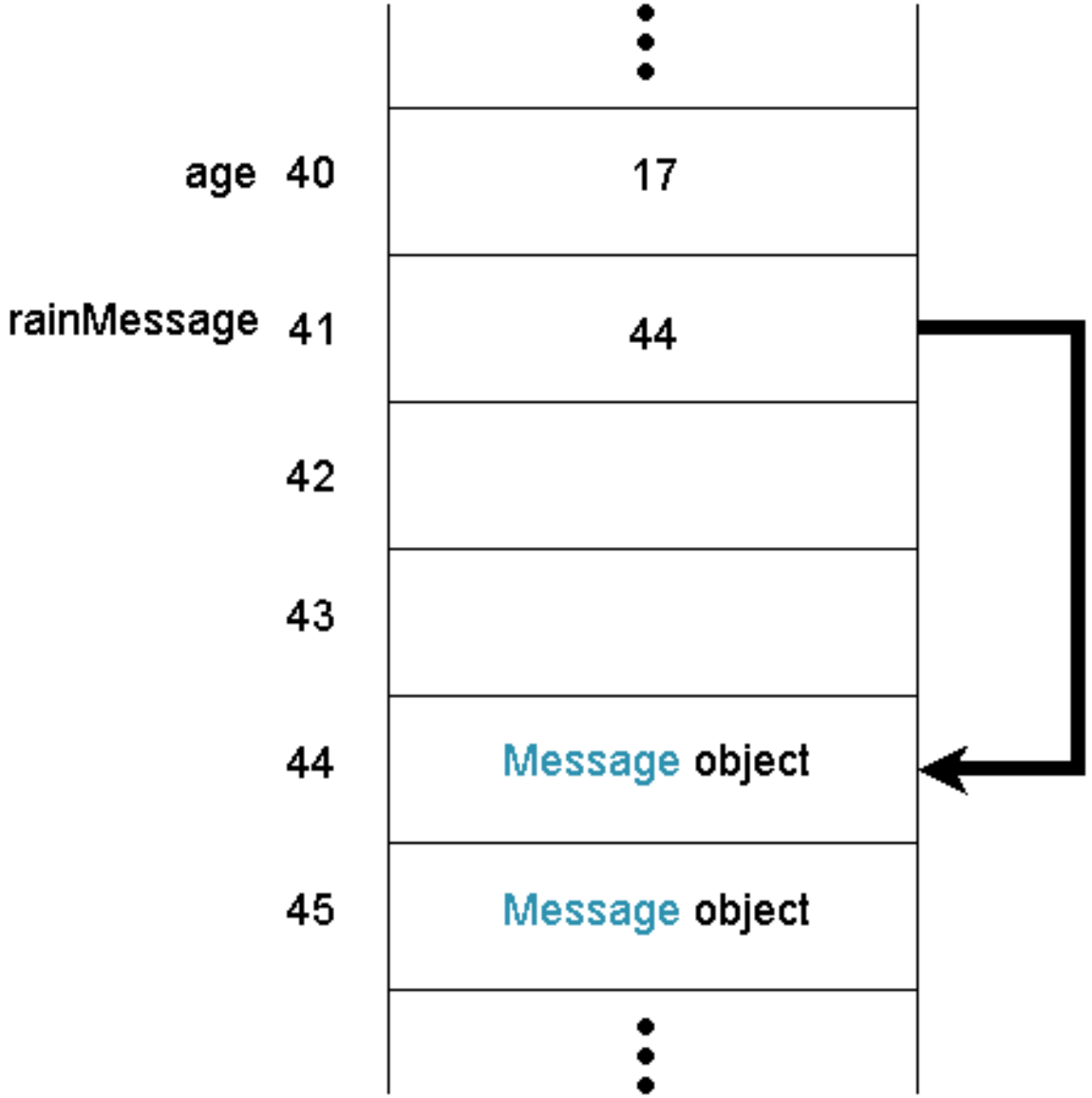
In-Lecture Quiz

UML stands for

- A: Unlikely Macho Leeroy
- B: Unfathomable Mathematical Lemma
- C: Integrated Development Environment
- D: Unified Modeling Language

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
rainMessage 41
44
42
43
44
Message object
45
Message object

- Recap
  - OO provides classes (reference types in C#) we can use to create objects
  - Objects interact with each other to implement our game
- Next Time
  - We'll start using classes and objects in C#