Code-O-Ween Exercise

1) Create an abstract class called Kid
    a) properties: name. age
    b) overridable method Say greeting “Hi, my name is {name}, I’m {age} years old”
    c) overload default constructor
        i) set age and name properties in body
2) Create an interface CandyTaker
    a) property candy collection of strings
    b) property costume string
    c) method trick that returns string
    d) method treat that returns string
3) Create TrickOrTreater class that inherits from Kid and implements CandyTaker
    a) overload default ctor to
        i) accept string costume
        ii) also require base class props
        iii) set instance property costume to ctor parameter
    b) if age (<6 trick() returns ‘being cute’) (>=6 and <13 returns ‘flossing’) (>=13 returns ‘flaming poo bag’)
    c) overrides SayGreeting “ToT I’m {age} my trick is {trick()} {treat()} :)”
    d) treat returns “check out my {costume}, candy please :)”
4) Create a collection of ToT’ers setting different properties
    a) loop through collection
        i) call SayGreeting
        ii) add something to the candy collection of each
        iii) what other properties/methods are available outside of the ToTer class?
        iv) how could we limit access or reading/writing (encapsulation)
    b) loop through ToT collection with another type of loop
        i) call trick() on even iterations and treat() on odd
        ii) remove candy from candy collection
5) Bonus!
    a) How can we prevent parents (other classes) from poaching Candy from the candy collection?
    b) What other classes might inherit from Kid?
    c) What could be a more specific implementation of the TrickOrTreater class, and how would we define that relationship?