CS 113 – Computer Science I

Lecture 13 – Objects

Adam Poliak
10/27/2022
Announcements

- Assignment 06
  - Due tonight (thursday 10/27)

- Mid-semester feedback

- Pythontutor.com - https://pythontutor.com/java.html
Access modifiers

Specify the access-level of instance variables/methods

- **public**
  - code outside of the class can access the variable/method

- **private**
  - code outside of the class cannot access the variable/method

- **protected**
  - Allow subclasses to access data in parent class

Default in Java is **public**
Access modifiers

Default in java is `public`

In this class, make instance data private
Class inheritance

Review:

- Classes are like categories
- Objects are like examples of the categories

Classes can be arranged hierarchically where,

  a child class "inherits" from a parent class
Inheritance: feature for organizing classes into hierarchies

- Animal
  - Reptile
    - Snake
    - Tree Lizard
  - Bird
    - Flamingo
    - Crow
    - Penguin
  - Fish
    - Shark
      - Hammerhead
Inheritance: subclasses refine behavior/state

Subclasses can override methods from parent class
Exercise

1. Implement getter functions for instance variables inside Animal

2. In Zoo.java, call the getters and output the values to console
Polymorphism

Program can treat all objects that extend a base class the same

Java automatically calls the specific methods for each subclass
public class Zoo {
    public static void main(String[] args) {
        Animal animal1 = new Animal();
        animal1.locomote();
        Animal animal2 = new Reptile();
        animal2.locomote();
    }
}

public class Animal {
    public Animal() {
    }
    public void locomote() {
        System.out.println("I am moving!");
    }
}

public class Reptile extends Animal {
    public Reptile() {
    }
    public void locomote() {
        System.out.println("I am walking!");
    }
}
Exercise: What is the output of this program?

```java
public class Zoo {
    public static void main(String[] args) {
        Animal animal1 = new Animal();
        animal1.locomote();

        Animal animal2 = new Fish();
        animal2.locomote();
    }
}
```

```java
public class Animal {
    public Animal() {
    }
    public void locomote() {
        System.out.println("I am moving!");
    }
}
```

```java
public class Fish extends Animal {
    public Fish() {
    }
    public void locomote() {
        System.out.println("I am swimming!");
    }
}
```
Question: How would we implement Minion?
Inheritance

- Player
- Entity
- NPC
  - Shop Keeper
  - Quest Giver
  - Orc
    - Minion
    - King
Exercise: Implement a Bird animal
OOP Example & Design: Vending machine
OOP Design: Vending machine
Defining the snack class

```java
public class Snack {
    private int mQuantity;
    private double mCost;
    private String mName;

    public Snack(String name, int quantity, double cost) {
        mQuantity = quantity;
        mCost = cost;
        mName = name;
    }

    public String getName() {
        return mName;
    }

    public void buy() {
        if (mQuantity > 0) {
            mQuantity--;
        }
    }
}
```
Testing the Snack class

```java
public static void main(String args[])
{
    Snack snack = new Snack("Slurm", 10, 1.5);
    System.out.println("Snack: "+snack.getName());
}
```