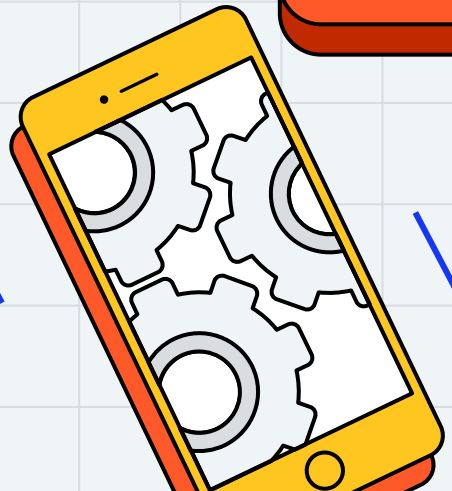
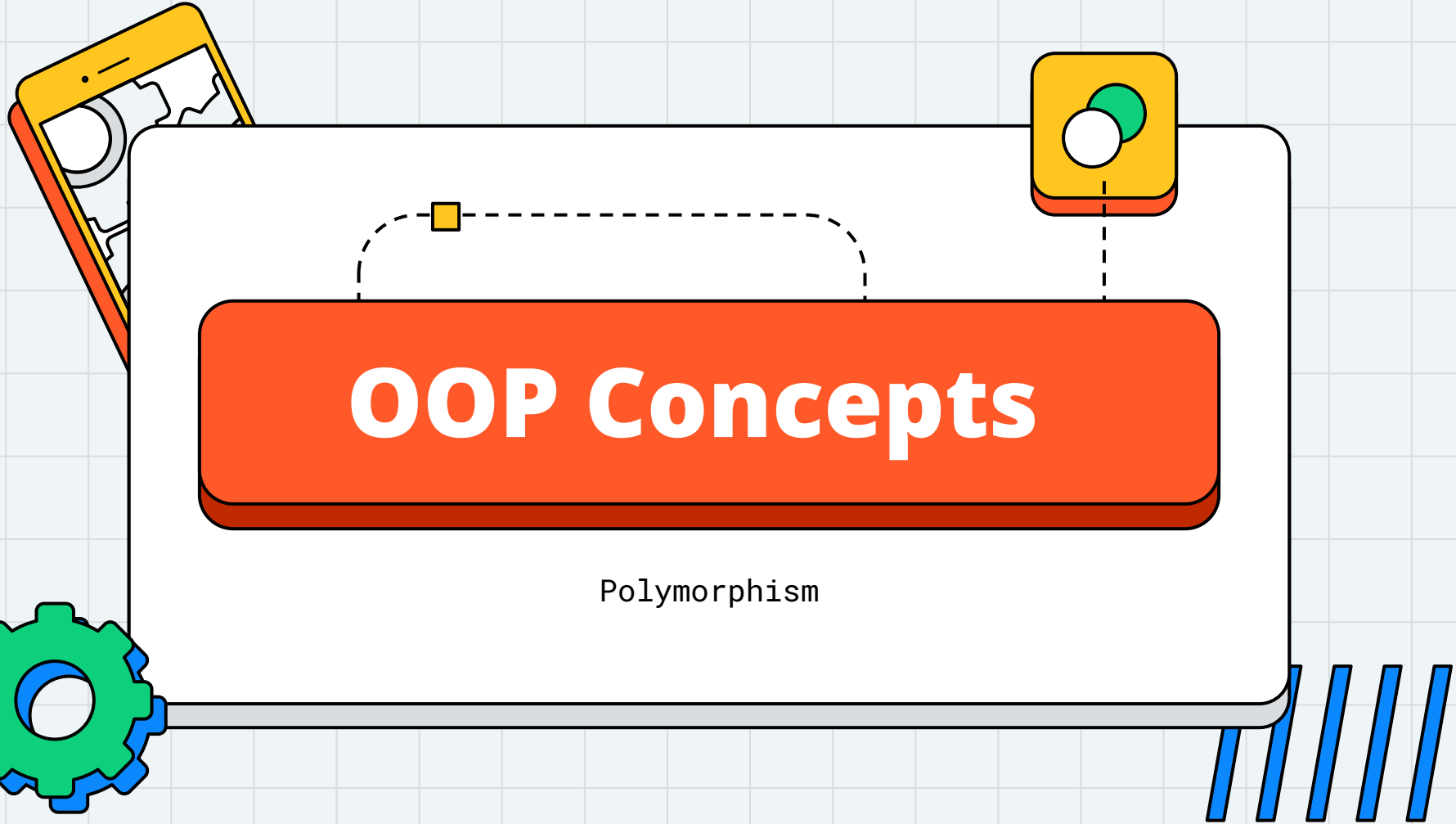
Application Programming



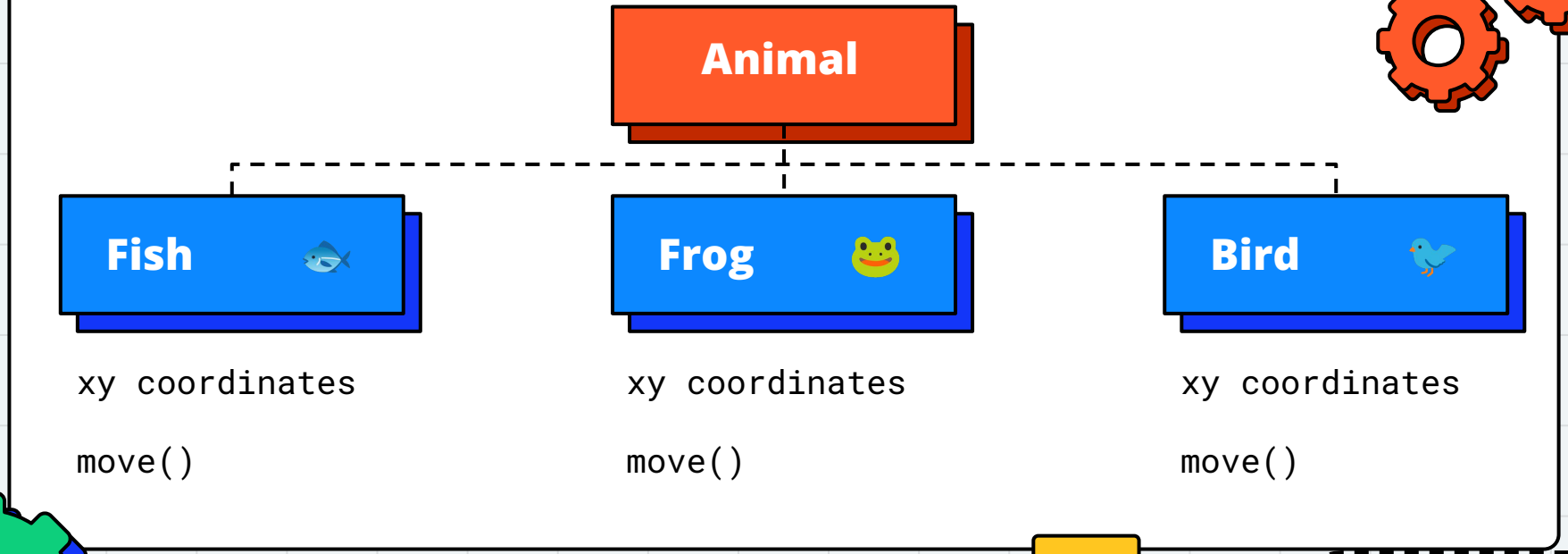
Hend Alkittawi


OOP Concepts
Polymorphism

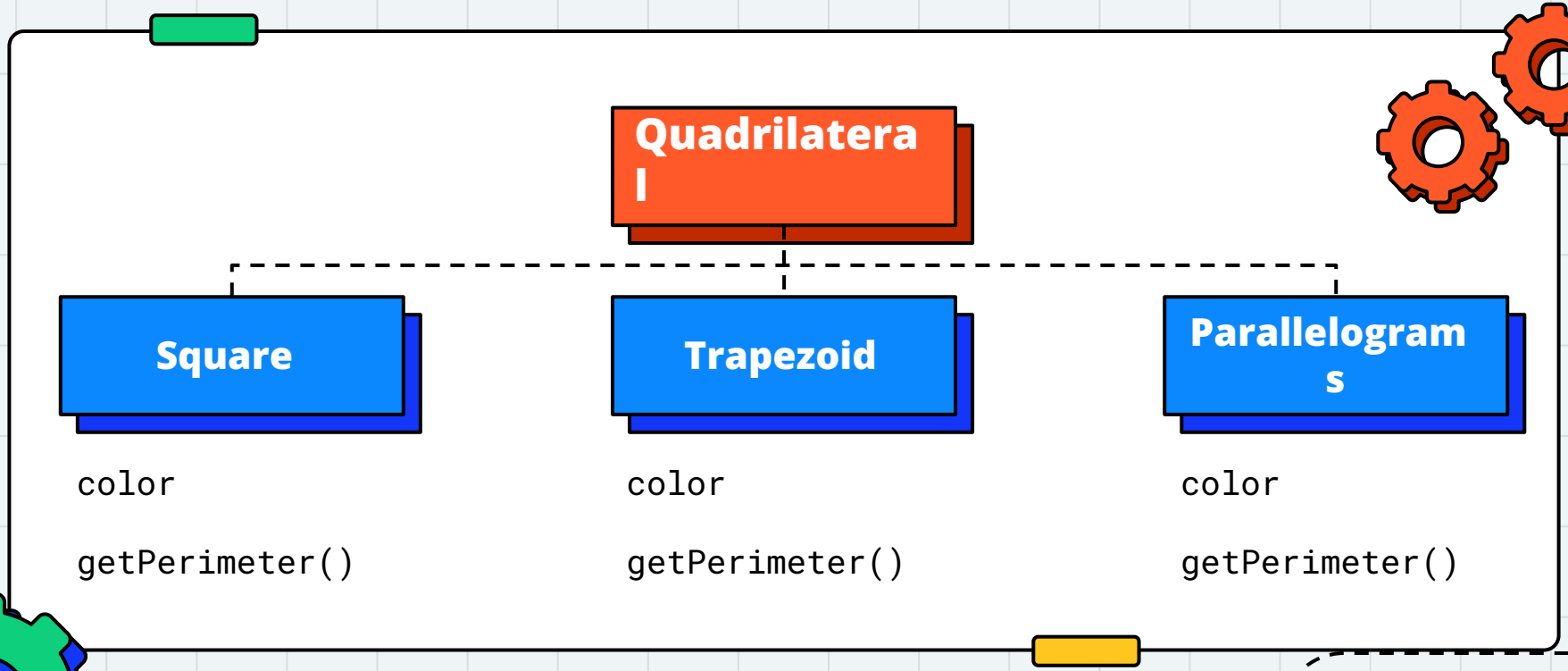
POLYMORPHISM

- Polymorphism enables you to write programs that processes objects that share the same superclass either directly or indirectly as if they were all objects of the superclass.
- The polymorphism occurs when a program invokes a method through a superclass variable, at execution time the correct subclass version of the method is called based on the type of reference stored in the superclass variable.
- A program can determine the type of an object at execution time and act on that object accordingly.

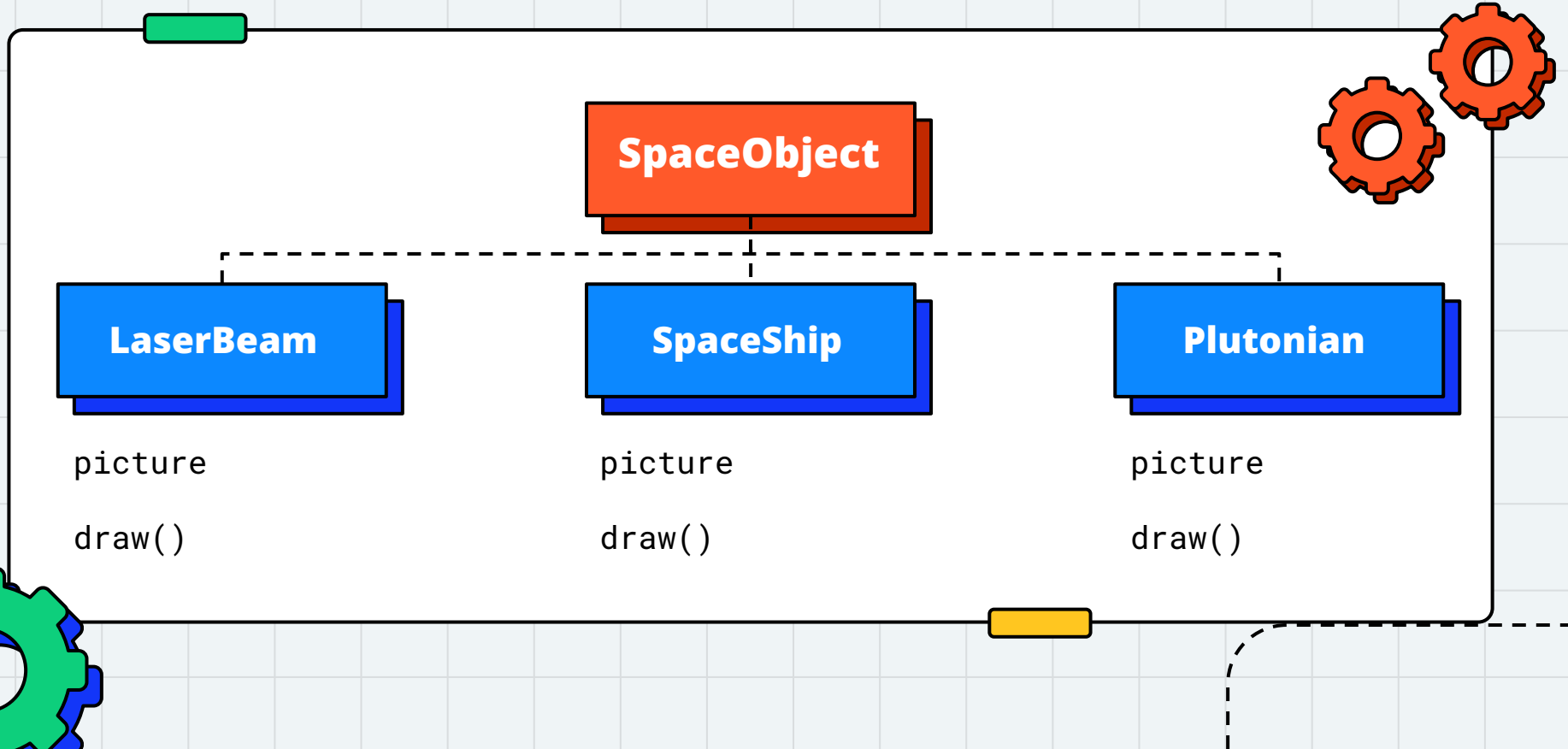
POLYMORPHISM



POLYMORPHISM



POLYMORPHISM



```java
public class Animal {

  private String name;

  public Animal(String name) { this.name = name; }

  public String getName() { return name; }

  public void setName(String name) { this.name = name; }

  public void eat() { System.out.println(getName() + " eats food."); }

  public void makeSound() { System.out.println("Animal makes a sound."); }
}
```

```java
public class PolymorphismDemo {

  public static void main(String[] args) {

    // Animal reference but Dog object
    Animal myAnimal = new Dog("Thunder", 30);
    // Output: Thunder the dog barks.
    myAnimal.makeSound();

    // Animal reference but Cat object
    myAnimal = new Cat("Whiskers", "Persian");
    // Output: Whiskers the cat meows.
    myAnimal.makeSound();
  }
}
```

```java
public class Dog extends Animal {

  private int exerciseNeeds; // in min/day

  public Dog(String name, int exerciseNeeds) {
    super(name);
    this.exerciseNeeds = exerciseNeeds;
  }
  public int getExerciseNeeds() {
    return exerciseNeeds;
  }
  public void setExerciseNeeds(int exerciseNeeds) {
    this.exerciseNeeds = exerciseNeeds;
  }
  @Override
  public void makeSound() {
    System.out.println(getName() + " the dog barks.");
  }
}
```

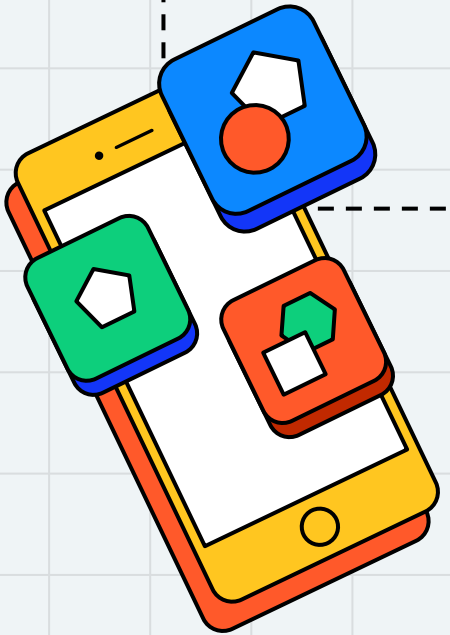
```java
public class Cat extends Animal {

  private String breed;

  public Cat(String name, String breed) {
    super(name);
    this.breed = breed;
  }
  public String getBreed() {
    return breed;
  }
  public void setBreed(String breed) {
    this.breed = breed; }
  @Override
  public void makeSound() {
    System.out.println(getName() + " the cat meows.");
  }
}
```

CODE DEMO

- Create classes to demo polymorphism concepts!



DO YOU HAVE ANY QUESTIONS?

hend.alkittawi@utsa.edu

By Appointment

Online