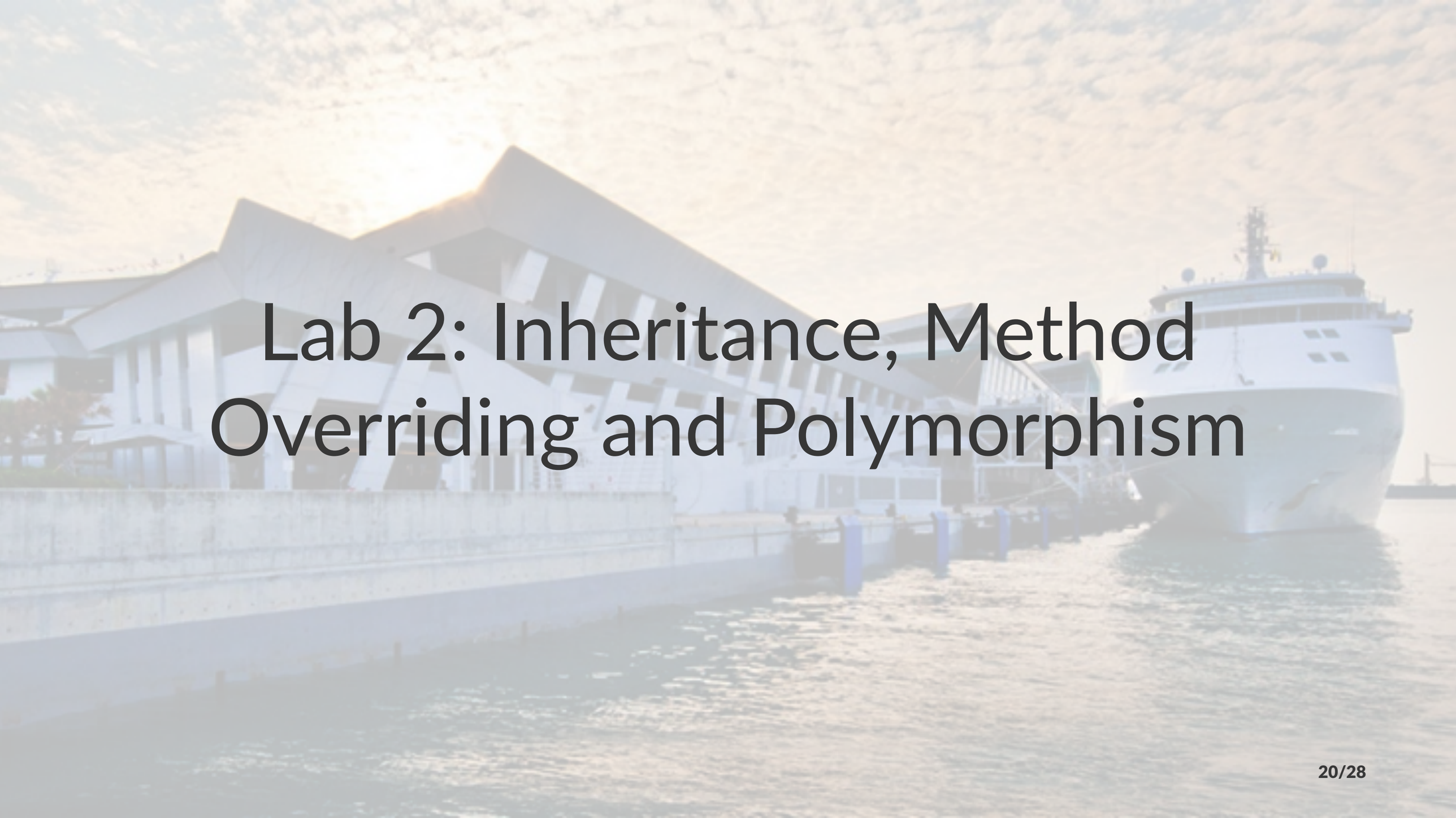

Lab 2: Inheritance, Method Overriding and Polymorphism

Inheritance

- **Superclasses and Subclasses**
  - Inheritance enables you to define a general class (i.e., a superclass) and later extend it to more specialized classes (i.e., subclasses).
  - is-a relationship
- The keyword `super` refers to the superclass and can be used to invoke the superclass's methods and constructors.

Method Overriding

- To override a method, the method must be defined in the subclass using the same signature as in its superclass.
- **Overloading** means to define multiple methods with the same name but different signatures. **Overriding** means to provide a new implementation for a method in the subclass.
- Every class in Java is descended from the `java.lang.Object` class.
  - `toString()` method

Spot the Difference!

```java
public class Test1 {
    public static void main(String[] args) {
        A a = new A();
        a.p(10);
        a.p(10.0)
    }
}

class B {
    public void p(double i) {
        System.out.println(i * 2);
    }
}

class A extends B {
    public void p(double i) {
        System.out.println(i);
    }
}
```

```java
public class Test2 {
    public static void main(String[] args) {
        A a = new A();
        a.p(10);
        a.p(10.0)
    }
}

class B {
    public void p(double i) {
        System.out.println(i * 2);
    }
}

class A extends B {
    public void p(int i) {
        System.out.println(i);
    }
}
```

Method Overloading

1. Different number of arguments
2. Different types of arguments.
3. Different order of arguments.

✅

```
void eat(Food food, Drink drink);

void eat(Food food);
void eat(Food food, Food snack);
void eat(Drink drink, Food food);
```

❌

```
void eat(Food food);

void eat(Food snacc);
Human eat(Food food);
```

Why use `@Override`?

- This annotation denotes that the annotated method is required to override a method in its superclass.
- If a method with this annotation does not override its superclass's method, the compiler will report an error.
- For example, if toString is mistyped as tostring, a compile error is reported. If the `@Override` annotation isn't used, the compiler won't report an error.

Polymorphism

- Polymorphism means that a variable of a supertype can refer to a subtype object.
- A subclass is a specialization of its superclass; every instance of a subclass is also an instance of its superclass, but not vice versa.
- Every circle is a geometric object, but not every geometric object is a circle.

```
void displayObject(GeometricObject object) {
    System.out.println(String.format("%s %s of dimension %s",
                                     object.getColor(),
                                     object.getShape(),
                                     object.getDimension()));
}

displayObject(new Circle(1, "red"));
displayObject(new Rectangle(1, 1, "black"));
displayObject(new Object()); // a superclass
```

```
[Output]
red circle of dimension 1
black rectangle of dimension 1x1
| Error:
| incompatible types: java.lang.Object cannot be converted to GeometricObject
| displayObject(new Object())
|               ^----------^
```

```
elinks index.html
```

Tips

1. What types of Cruises and Loaders are there?
2. What relationships do they share?
3. If done well, your implementation should be easily extended into Level 6 (Secret Level), released later on CodeCrunch.