

Inheritance

Introduction

Lecture Contents

- Why Use Inheritance
 - School Registry Example
- Inheritance and Constructors
- Inheritance and Instance Methods

Review

- ***encapsulation***: the bundling together of closely related data, along with the methods that act on that data
- ***abstraction***: hiding the implementation details while providing a simple interface

Inheritance Example: School Registry

- Classes to store information about each student / parent.

Student
<pre>-studentNum: int -name: String -parent: Parent</pre>
<pre>+Student(num: int, name: String) +Student(num: int, name: String parent: Parent) +getName(): String +setName(name: String) +getStudentNumber(): int +setStudentNumber(number: int) +getParent(): Parent +setParent(parent: Parent) +toString(): String</pre>

Parent
<pre>-name: String -contactInfo: String</pre>
<pre>+Parent(name: String contactInfo: String) +getName(): String +setName(name: String) +getContactInfo(): String +setContactInfo(info: String) +toString(): String</pre>

What can you say
about these classes?

Inheritance Example: School Registry

What disadvantages would creating a single class to store both Student info and Parent info?

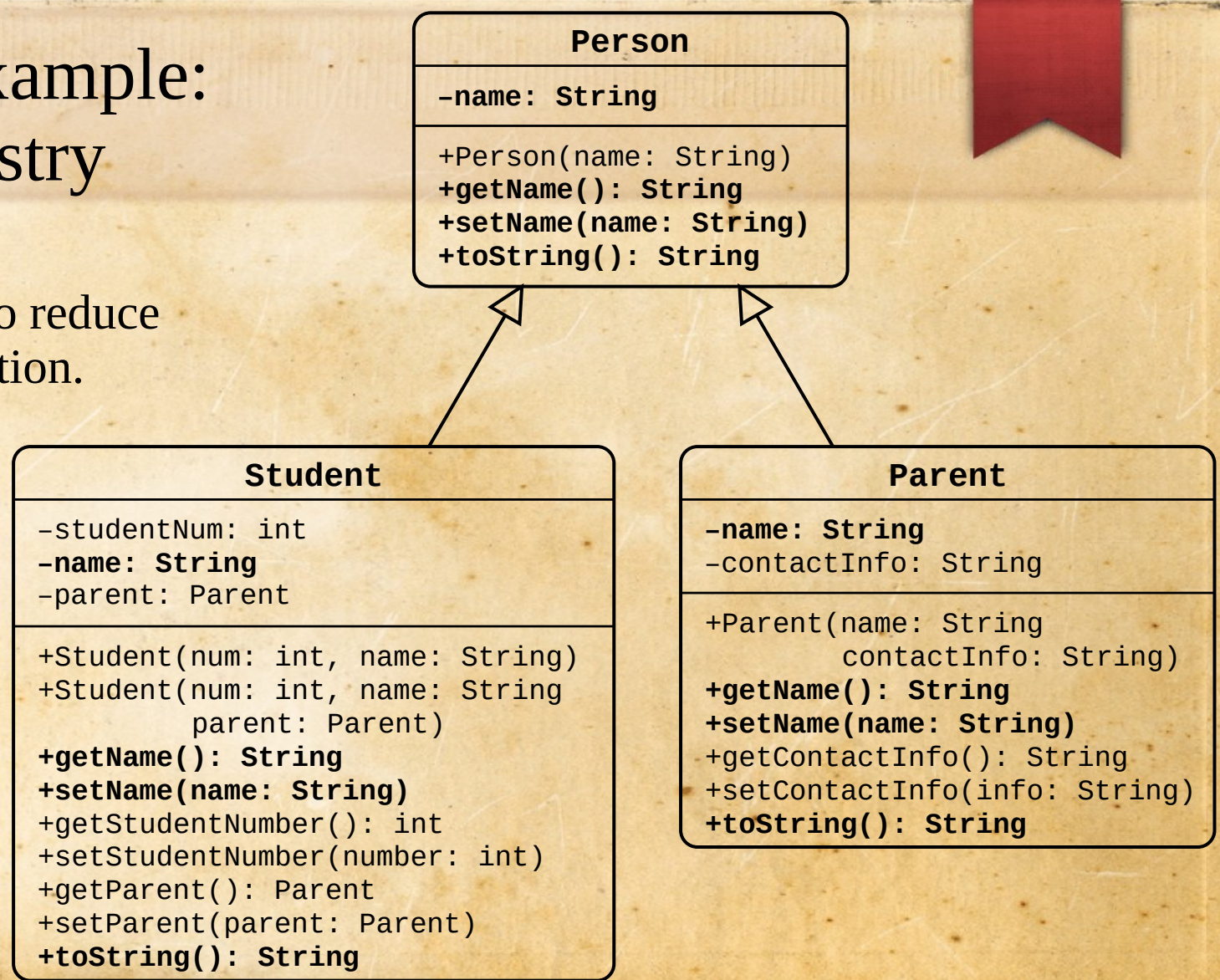
Person

```
-studentNum: int  
-name: String  
-parent: Parent  
-contactInfo: String
```

```
+Person(name: String)  
+getName(): String  
+setName(name: String)  
+getStudentNumber(): int  
+setStudentNumber(number: int)  
+getContactInfo(): String  
+setContactInfo(info: String)  
+getParent(): Parent  
+setParent(parent: Parent)  
+toString(): String
```

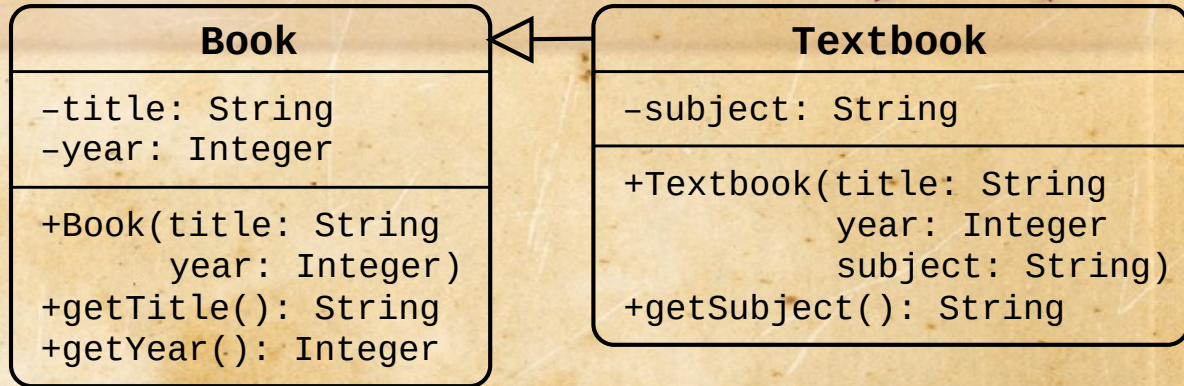

Inheritance Example: School Registry

Inheritance to reduce
code duplication.



Constructor

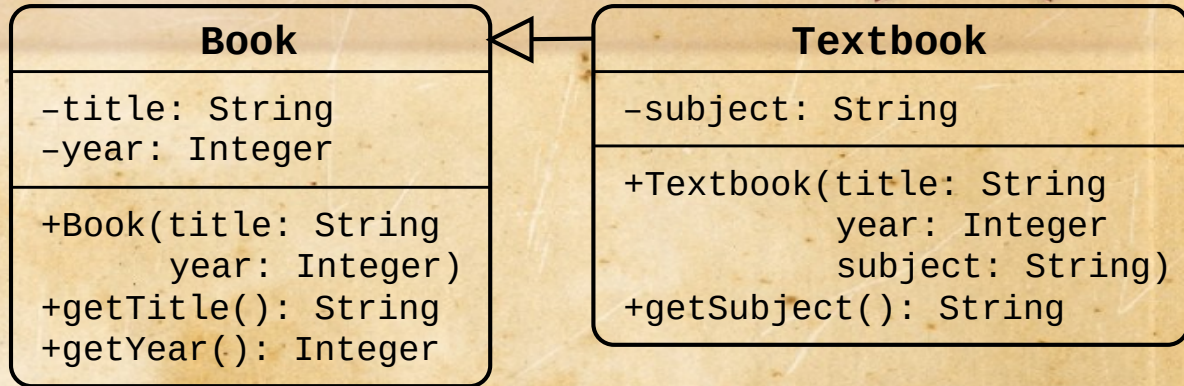
- What happens when a class has no explicit constructor defined?



Constructor

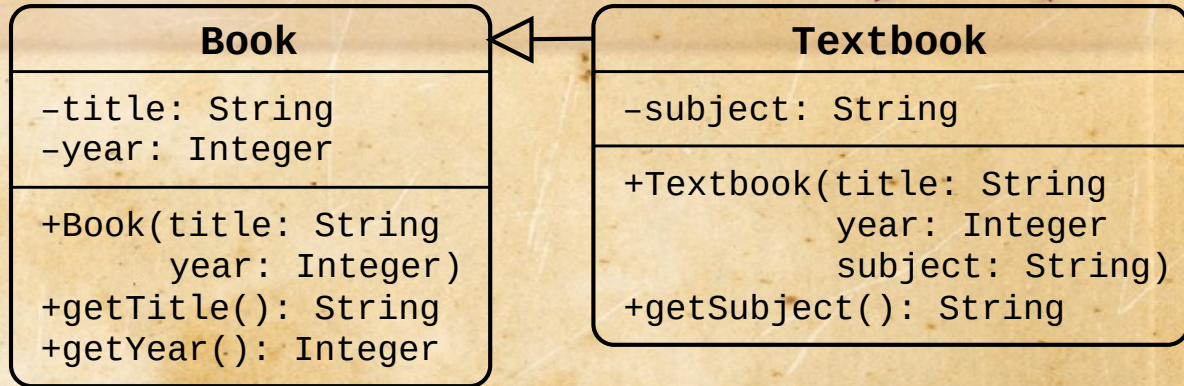
- What happens when a class has no explicit constructor defined?
- Default constructor (from previous knowledge):

```
public Textbook() {  
    this.subject = null;  
}
```



Constructor

- What happens when a class has no explicit constructor defined?
- Default constructor calls the super class constructor:

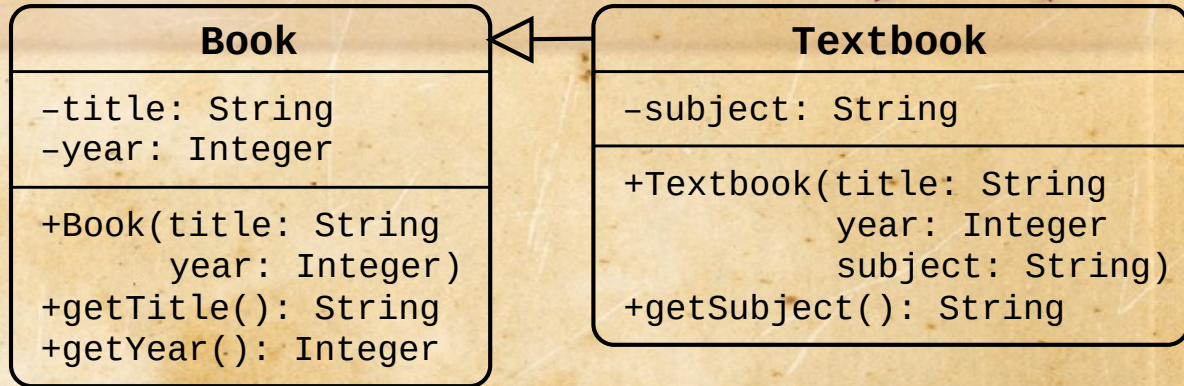


```
public Textbook() {
    super();
    this.subject = null;
}
```

What will happen during compilation?

Constructor

- What happens when a class has no explicit constructor defined?
- Default constructor calls the super class constructor:

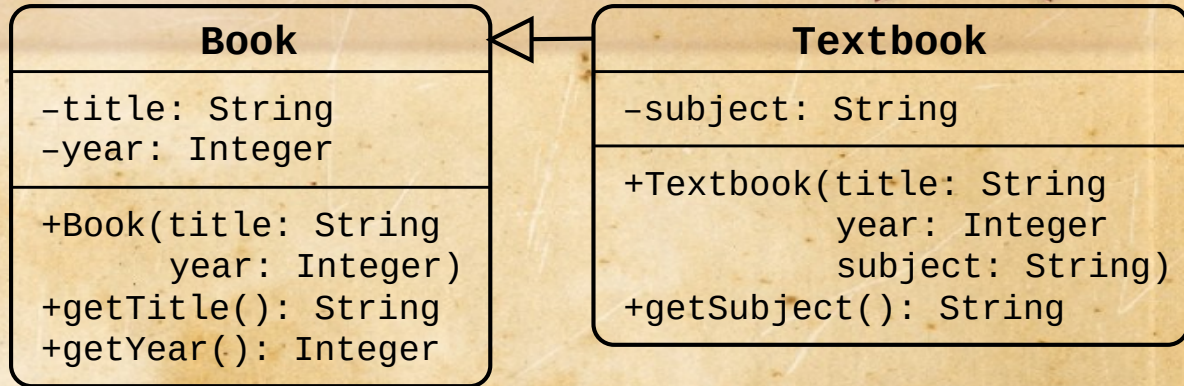


```
public Textbook() {
    super();
    this.subject = null;
}
```

ERROR: No zero-parameter constructor in class Book.

Constructor

- Explicitly declared constructor.



```
public Textbook(String title,
                int year,
                String subject) {
    super(title, year);
    this.subject = subject;
}
```

The superclass constructor is called with the `super` keyword.

Calling Superclass Instance Methods

- Superclass `toString` method:

```
@Override
public String toString() {
    return "Title: " + title + "\n" +
           "Year: " + year + "\n";
}
```

Notice the similarity to usage of the `this` keyword.

- Subclass `toString` method:

```
@Override
public String toString() {
    return super.toString() +
           "Subject: " + subject + "\n";
}
```


Inheritance

Introduction