

COMP 250

INTRODUCTION TO COMPUTER SCIENCE

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Week 3-4: OOD3 Other methods, mutable, final variables

Add WeChat edu_assist_pro

Giulia Alberini, Fall 2020

OBJECTS – QUICK REVIEW

Book.java

```
public class Book {  
    public String title;  
    public String author;  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

TestBook.java

```
public class TestBook {  
    public static void main(String[] args) {  
        Book b = new Book();  
        b.title = "Matilda";  
        b.author = "Roald Dahl";  
        System.out.println(b);  
    }  
}
```

Add WeChat edu_assist_pro

OBJECTS – QUICK REVIEW

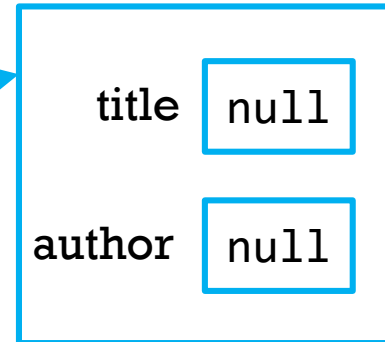
```
public class Book {  
    public String title;  
    public String author;  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
public class TestBook {  
    public static void main(String[] a  
        Book b = new Book();  
        b.title = "Matilda";  
        b.author = "Roald Dahl";  
        System.out.println(b);  
    }  
}
```



OBJECTS – QUICK REVIEW

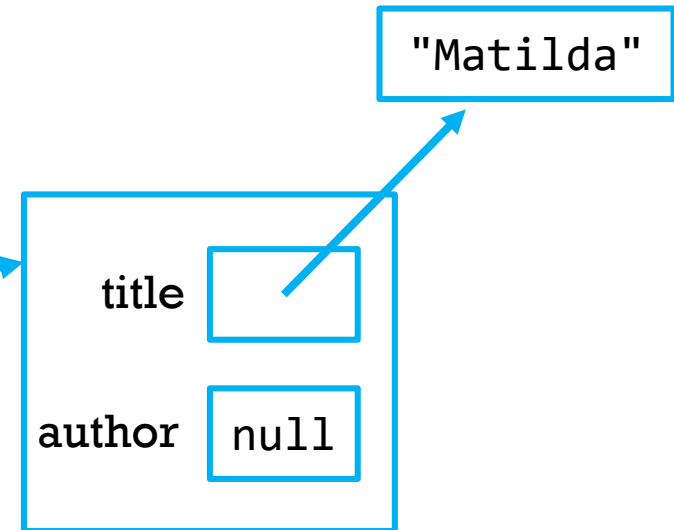
```
public class Book {  
    public String title;  
    public String author;  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
public class TestBook {  
    public static void main(String[] a)  
    {  
        Book b = new Book();  
        b.title = "Matilda";  
        b.author = "Roald Dahl";  
        System.out.println(b);  
    }  
}
```



OBJECTS – QUICK REVIEW

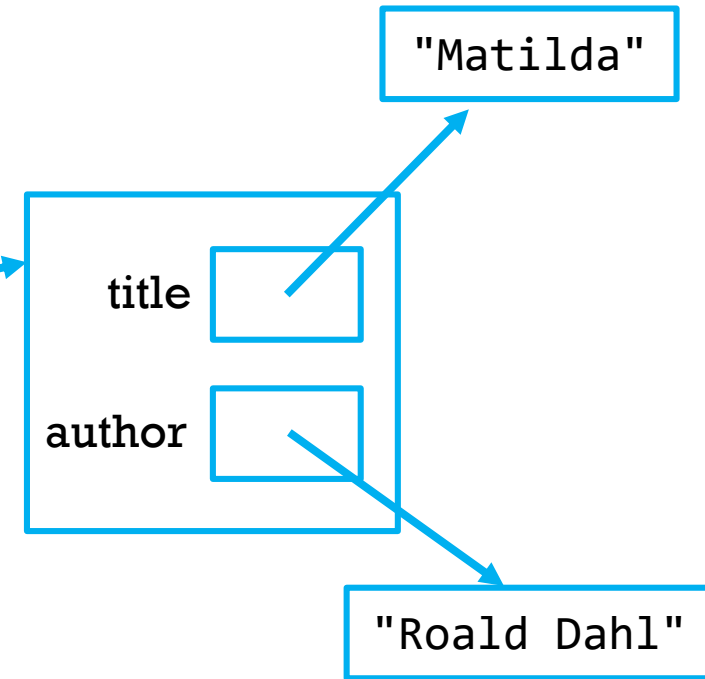
```
public class Book {  
    public String title;  
    public String author;  
}
```

```
public class TestBook {  
    public static void main(String[] a  
        Book b = new Book();  
        b.title = "Matilda";  
        b.author = "Roald Dahl";  
        System.out.println(b);  
    }  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



OBJECTS – QUICK REVIEW

```
public class Book {  
    public String title;  
    public String author;  
}
```

Assignment Project Exam Help

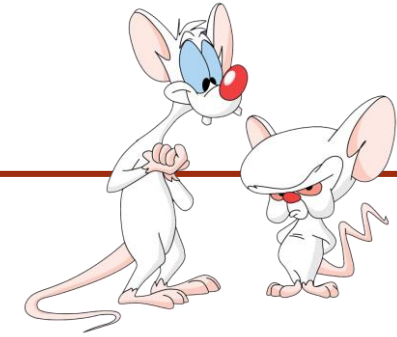
<https://eduassistpro.github.io/>

```
public class TestBook {  
    public static void main(String[] args) {  
        Book b = new Book();  
        b.title = "Matilda";  
        b.author = "Roald Dahl";  
        System.out.println(b);  
    }  
}
```

Add WeChat edu_assist_pro

A reference

WHAT ARE WE GOING TO DO IN THIS VIDEO?



OOD3

- Other methods
- Mutable vs Immutable
- `final` variables

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Assignment Project Exam Help

O

DS

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

STEP 3

```
public class ClassName {
```

```
// some data declared here
```

```
<modifier> <type> <variable_name>;
```

```
public ClassName (
```

```
//constructor
```

```
}
```

```
// declare other methods
```

```
}
```

Assignment Project Exam Help

Data

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Method to create an
object

Other methods

File name: **ClassName.java**

OTHER METHODS

- You write all the other methods as before, except:
 - If the method is static, it can access the class variables.
 - If the method is not static, it can access both the class and the instance variables

RECOMMENDED EXERCISES

In the `Patient` class:

- add a method `printName` that prints out the name of the patient.
- add a method `addTemp` that takes a `double` and appends it at the end of the array of temperatures. To do this, you need to create a new array with a size that is *greater* than the current size, copy all the values over, append the double, and update the value of the corresponding field.

Assignment Project Exam Help

GE

DS

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

GETTERS AND SETTERS

- In general, all fields should be declared as `private`. We can then declare `public` methods to regulate how they can be accessed.

<https://eduassistpro.github.io/>

- Those methods that get the value of a field are formally called "accessors", but commonly referred to as **getters**.

- Those methods that allow you to modify the value of a field are formally called "mutators", but commonly referred to as **setters**.

GET (ACCESSOR) METHOD

Most getters have a very similar format:

```
public <type> getF  
    return this.fiel  
  
}
```

SET (MUTATOR) METHOD

Most setters have a very similar format:

```
public void setField(<          ue) {  
    this.field = value;  
}
```

SET, GET, AND REFERENCES



Careful when writing get/set methods:

Assignment Project Exam Help

- If you are returning <https://eduassistpro.github.io/> if you are actually giving access to the thing you intend. Add WeChat edu_assist_pro
- If you are setting a value, be sure that you are using the thing you intend.

TRY IT!

Assignment Project Exam Help

- Modify the `Patie` and set methods.
How should you <https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`

ENCAPSULATION

- Process of wrapping data and the code acting on that data in one unit. The idea is to better control the data.

Assignment Project Exam Help

- What to do?

<https://eduassistpro.github.io/>

- Make all the fields `private`

Add WeChat edu_assist_pro

- Provide getters and setters as needed

- Note: through the methods we can do data validation, while we have little control over the data stored in a `public` field.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

PRIVATE AND INFORMATION DISPLAY

- When your fields are all private, it might be difficult to display the content of an Object.
<https://eduassistpro.github.io/>
- If we try to use `println` on the Object, it will see a reference.
[Add WeChat: edu_assist_pro](#)
- References are not very useful when trying to debug a code.

toString() AND println()

- Each time we use `println()` with objects, it calls the method `toString()` on the object and displays the result.

Assignment Project Exam Help

- By default, `toString()` <https://eduassistpro.github.io/> representing the address at which the contents of the Object are stored.

Add WeChat edu_assist_pro

- This is not the case for all the Objects, though. If the method `toString()` is included inside the class, then when we use `println()` on variables of such type, not the default `toString()`, but the one from the class is used.

THE toString() METHOD

You can write a `toString()` method in any class.

It must have the following header:

```
public String toString()  
    // returns a value of type String  
}
```

If you do that, then when you call `print()` / `println()` on an instance of that class, this method is called automatically!

TRY IT!

In the `Patient` class, add a `toString()` method that returns a `String` in the following format:

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Name: [name]

Age: [age]

Reported Temperatures: [list of temperatures]

Now, try to use `print` on an object of type `Patient`.

The background of the image features a series of concentric circles in a light gray color, centered around the middle of the frame. Overlaid on this background is a large, solid brown rectangle that occupies the central portion of the image. Inside this brown rectangle, the word "MUTABLE" is written in a large, white, sans-serif font. Below the word, there are three lines of red text: "Assignment Project Exam Help", a URL "https://eduassistpro.github.io/", and "Add WeChat edu_assist_pro". At the bottom of the brown rectangle, there is a solid brown horizontal bar.

MUTABLE

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

IMMUTABLE REFERENCE TYPES

- If when you create a class, you make all the fields private, and you provide methods, then you have an Immutable Type!

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

- If the only way to assign values to fields is through the constructor, then the values of the Object cannot be changed after it has been created.

MUTABLE

Mutator methods change the content of the object.

Assignment Project Exam Help

```
Cat myCat = new Cat("Small");  
myCat.setName("Small");
```

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

The content of object reference by `myCat` has changed by changing the name of the cat.

MUTABLE

If we add a second `Cat` variable referencing to the same cat

Assignment Project Exam Help

```
Cat myCa https://eduassistpro.github.io/
Cat aCat = myCat;
myCat.setName("Small Cat");
```

Then, also the content of `aCat` is changed after the last instruction.

IMMUTABLE

Assignment Project Exam Help

```
String s = https://eduassistpro.github.io/  
s.charAt(0) = 'r'; // c me error!  
Add WeChat edu\_assist\_pro
```

There is no method in `String` that allows us to set the value of a character.

IMMUTABLE

Assignment Project Exam Help

```
Str  
Str  
t = "dogs";
```

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

The value of `s` does not change!

MUTABLE VS IMMUTABLE

- Mutable objects: more flexible in what they allow users to do with the object and more efficient because you don't create a new object each time you want to modify it. This can be error-prone.
<https://eduassistpro.github.io/>
- Immutable: easier to keep track of objects when the contents of the object change. They can save you from long hours of debugging.
Add WeChat edu_assist_pro
- Immutable reference types behave like primitive data type.

CONSTRUCTORS AND MUTABLE REFERENCE TYPES

Remember the Patient class:

```
public class Patient{  
    private String name;  
    private int age;  
    private double[] temps;  
  
    public Patient(String n, int a, double[] t) {  
        this.name = n;  
        this.age = a;  
        this.temps = t;  
    }  
}
```

Code like this can cause issues!

EXAMPLE

Patient.java

```
public class Patient{
    private String name;
    private int age;
    private double[] temps;

    public Patient(String n, int a, double[] t){
        this.name = n;
        this.age = a;
        this.temps = t;
    }
}
```

TestPatient.java

```
public class TestPatient{

    void main(String[] args) {
        tempsForP= {37.8, 38.6, 40.0, 37.4, 36.5};
        w Patient("John", 42, tempsForP);
    }

}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

The last statement in TestPatient.java will change the value of the field temps even though it is a private field!

GETTERS AND MUTABLE REFERENCE TYPES

Let's add a get method for the field temps:

```
public class Patient {  
    private String name;  
    private i  
    private d  
  
    public Patient(String n, double t) {  
        ...  
    }  
    public double[] getTemps() {  
        return this.temps;  
    }  
}
```

Also code like this can cause issues!

EXAMPLE

Patient.java

```
public class Patient{
    private String name;
    private int age;
    private double[] temps;

    public Patient(String n, int a, double[] t){
        ...
    }
    public double[] getTemps() {
        return this.temps;
    }
}
```

TestPatient.java

```
public class TestPatient{

    void main(String[] args) {
        tempsForP= {37.8, 38.6, 40.0, 37.4, 36.5};
        w Patient("John", 42, tempsForP);
        .getTemps();

        x[0] = 0.0;
    }
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Similarly, the last statement in `TestPatient.java` will change the values of the field `temps` even though it is a private field!

GUIDELINE

- Don't write a constructor initializing a mutable reference type without making a copy!
<https://eduassistpro.github.io/>
- Don't add a get/set method for a mutable reference type without making a copy!

EXAMPLE

Patient.java

```
public class Patient{
    private String name;
    private int age;
    private double[] temps;

    public Patient(String n, int a, double[] t) {
        ...
    }
    public double[] getTemps() {
        // create a copy
        int n = this.temps.length;
        double[] c = new double[n];
        for(int i=0; i<n; i++) {
            c[i] = this.temps[i];
        }
        return c;
    }
}
```

TestPatient.java

```
public class TestPatient{
    public static void main(String[] args) {
        psForP= {37.8, 38.6, 40.0, 37.4, 36.5};
        new Patient("John", 42, psForP);
        d = new Patient().getTemps();
        x[0] = 0.0;
    }
}
```

Now it doesn't matter because the get method returns a copy of the array.

Assignment Project Exam Help

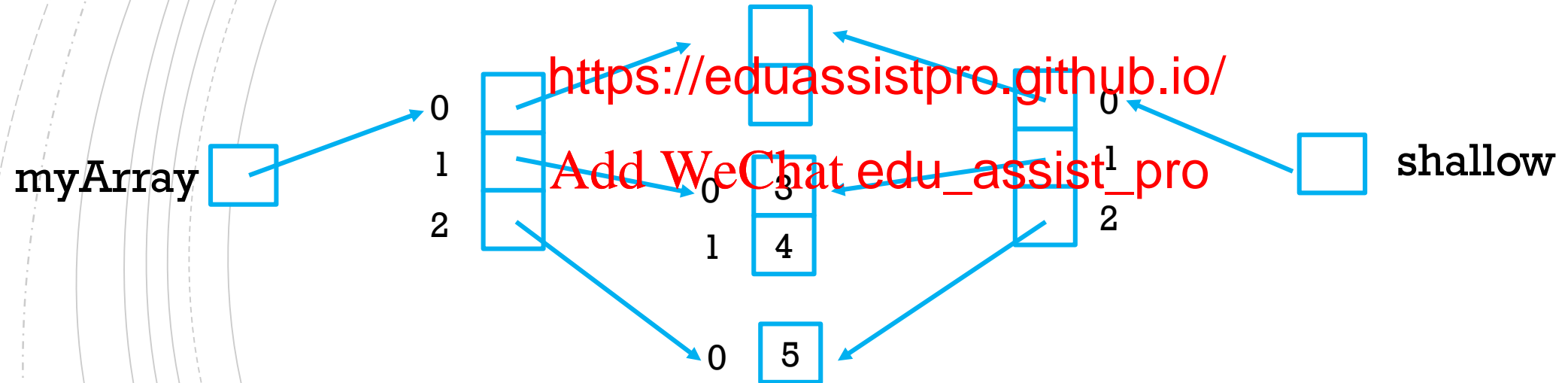
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

SHALLOW VS DEEP COPY

Consider a 2-dimensional array. A shallow copy is obtained by creating a new array and copying to it the references stored in the original array one by one.

Assignment Project Exam Help

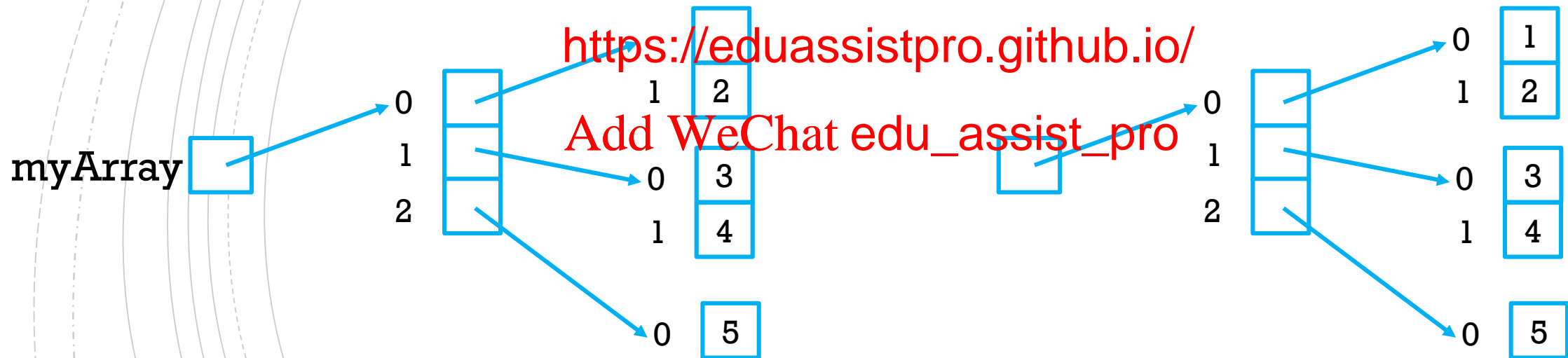


```
shallow[0][0] = 5; # affects also myArray
shallow[0] = new int[1]; # does not affect myArray
shallow = new int[5][1]; # does not affect myArray
```

SHALLOW VS DEEP COPY

Consider a 2-dimensional array. A deep copy is obtained by creating a new array and copying to it a (deep) copy of the objects in the original array one by one.

Assignment Project Exam Help



Nothing we can do through deep can affect myArray (and vice versa)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

FINAL VARIABLES

If a variable is declared to be final, its value can **never** be changed after it has been initialized.

Assignment Project Exam Help

```
final https://eduassistpro.github.io/  
x = 10; // compile-time error!
```

Add WeChat edu_assist_pro

```
final Cat myCat = new Cat("Small cat");  
myCat = new Cat("Spritz"); // compile-time error!
```


MUTABLE REFERENCE TYPES AND FINAL

```
final Cat myCat = new Cat("Small cat");  
myCat = new Cat("Spritz"); // compile-time error!
```

<https://eduassistpro.github.io/>

However, you can still change the object it points at, without changing myCat's value.

```
myCat.setName("Spritz"); // no problem!
```

FINAL FIELDS

- Final fields must be initialized!

(Otherwise **compile-time error**)

- If the class has a final **final non-static field**, you must initialize it in every instance of the class.

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- If the class has a final class variable (i.e. a final static field), you should initialize it in place (on the same line of the declaration) or in a Static Initializer Block.

An orange paint roller with a red handle, positioned horizontally. The roller is partially filled with orange paint, and there are orange paint splatters and drips around it. The text "Coming Soon" is written in white on the orange surface of the roller.

Coming Soon

Assignment Project Exam Help

In the next

- UML Dia

- Inheritance

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro