

Please note that the slides published AFTER the lectures and workshops are the official slides and are the ones that should be used for revision.



Add WeChat edu\_assist\_pro

Workshop 01 (with answers)

OO and Java Refresher (1/2)

Peer-Olaf Siebers



- Lecture 1
  - What is Software Maintenance?
  - Information about module organisation
  - Some examples of software
- Lab 1
  - Eclipse and IntelliJ
  - Practicing Java basics
  - Working with existing code
- Workshop 1
  - OO and Java Programming Refresher

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Case Study: Zoo Management

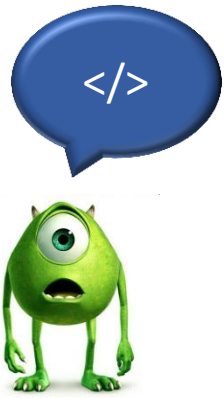


Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Case Study: Zoo Management



- Come up with a draft class diagram
  - Note that this is only a small choice of relevant classes!

**Assignment Project Exam Help**

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

Add WeChat edu\_assist\_pro



## Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Case Study: Zoo Management



As we focus on Java basics today we want to keep it simple ...

**Assignment Project Exam Help**

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

**Add WeChat edu\_assist\_pro**



# Basic OO Concepts



- Object-oriented programming is founded on these ideas:
  - **Abstraction**: Simple things like objects represent more complex underlying code and data
    - A class is a blueprint for a
    - An object is an entity that focuses on that data
  - **Encapsulation** (information hiding): The ability to hide the components of the object from external access
    - e.g. keeping fields within a class private, then providing access to them via public methods
  - **Inheritance**: The ability for a class ("subclass") to extend or override functionality of another class ("superclass")



- Object-oriented programming is founded on these ideas:
  - **Polymorphism**: The provision of a single interface to entities of different types
    - Compile time (static) pol
      - Method overloading: C++ but different signatures
    - Run time polymorphism through...
      - Method overriding: Create method in derived class and signature than in base class
      - Sub classing: reference of base class is able to create and destroy objects of derived class
  - **Interface**: A specification of method signatures (without implementations) as a mechanism for enabling polymorphism in a declarative way.



# What's coming up ...



- Public vs. Private
- Accessors and Modifiers
- Encapsulation
- The "this" keyword
- Constructors
- Passing parameters
- Static fields and methods

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Public vs. Private



- What are the general rules for constructors, methods, helper methods, fields, and static constants?

## Assignment Project Exam Help

- Constructors and methods
  - Usually declared public (to be accessible from outside the class)
- Helper methods that are needed only inside the class
  - Usually declared private
- Fields
  - Usually declared private (to support encapsulation)
- Static constants
  - Usually declared public

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

Add WeChat edu\_assist\_pro

# Accessors and Modifiers



- Accessors (also called Getters):

- Methods that return values of private fields
- Name often starts with get

Assignment Project Exam Help

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

- Modifiers (also called Mut

- Methods that set values of private fields
- Name often starts with set

Add WeChat edu\_assist\_pro



# Encapsulation



- Hiding the implementation details of a class (making all fields and helper methods private) is called encapsulation

Assignment Project Exam Help

- Encapsulation helps in protecting the implementation details of a class so that a change in one class does not affect other classes

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

Add WeChat edu\_assist\_pro

- A client of a class interacts with the class only through well-documented public constructors and methods; this facilitates team development

# The Keyword "this"



- "this" refers to the implicit parameter inside your class
  - A variable that stores the object on which a method is called

**Assignment Project Exam Help**

- Refer to a field
  - `this.field`

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

- Call a method
  - `this.method(parameters);`

**Add WeChat edu\_assist\_pro**

- One constructor can call another
  - `this(parameters);`



# Constructors



- What are constructors used for?

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Constructors



- A constructor is a procedure for creating objects of the class
  - A constructor often initialises an object's fields
  - Constructors do not have a return type
  - All constructors in a class have the same name as the class (of the class)
  - Constructors may take parameters (https://eduassistpro.github.io/)
  - If a class has more than one constructor, they have different numbers and/or types of parameters (constructor overloading) (Add WeChat edu\_assist\_pro)
- Important!
  - Java provides a default constructor for a specific class
  - If you define a constructor for a class, Java does not provide the default constructor anymore

# Constructors



- Constructors of a class can call each other using the keyword "this" (referred to as constructor chaining) - a good way to avoid duplicating code

**Assignment Project Exam Help**

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

**Add WeChat edu\_assist\_pro**





# Invoking Constructors



- Constructors are invoked using the operator new.
  - Declare a reference variable of the required type and then invoke the constructor method after the "new" keyword.

Assignment Project Exam Help

- Parameters passed to "new" are, types, and order of parameters expected by one of the constructors.

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

Add WeChat edu\_assist\_pro

# Invoking Constructors



- What does the output look like?

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro



# Passing Parameters



- In Java, parameters sent to methods are passed by value
  - Just to clarify some terminology
    - The "type" of data that a method can receive is referred to as a "parameter"
    - What is passed "to" a method
- Meaning of "pass-by-value"
  - In this case actual parameter is evaluated and copied into memory (stack) used by the parameters of the method.
- Common misconception: "In Java primitives are passed by value and objects are passed by reference"
  - Objects are not passed by reference but object references (pointers) are passed by value
  - You can test this by using the "Litmus" test (writing a simple swap() function)

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

<http://www.javadude.com/articles/passbyvalue.htm>



# Passing Parameters



- Inside a method, "this" refers to the object for which the method was called. "this" can be passed to other constructors and methods as a parameter.

**Assignment Project Exam Help**

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

**Add WeChat edu\_assist\_pro**

# Return Statement



- A void method can use a return statement to quit the method early
- There is no need for a return at the end

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Overloaded Methods



- Methods of the same class that have the same name but different numbers or types of parameters are called overloaded methods
- The compiler treats overloaded methods as completely different methods
- The compiler knows which number and the types of the parameters passed to the
- The return type alone is not sufficient for distinction between overloaded methods

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro



# Static Fields



- A static field (class field or class variable) is shared by all objects of the class
- A non-static field (instance field or instance variable) belongs to an individual object

Assignment Project Exam Help

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

- Static fields are stored with the class code from instance variables that describe an individual object

Add WeChat edu\_assist\_pro

# Static Fields



- Public static fields, usually global constants, are referred to in other classes using dot notation
  - `ClassName.constName`
- Usually static fields are **NO** **Constructors** (they are initialized either in declarations or in public static blocks)
- If a class has only static fields, there is **no** need to create objects of that class (all of them would be identical).
  - `Math` and `System` are examples of the above (they have no public constructors and cannot be instantiated)

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro



# Static Methods



- Static methods can access and manipulate class's static fields. They belong to the class - not an instance of it.
- Static methods cannot access instance fields or call instance methods of the class; instance methods can access static and non-static methods of their class - both
- Static methods will usually take input from the user, perform actions on it, then return some result.
- Static methods are called using dot notation
  - `ClassName.statMethod(...)`

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro



# Static Fields and Methods



- What does the output look like?

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro



# Static Fields and Methods

- Does this compile?

```
1 package org.siebers.olaf.peer;  
2  
3 public class ZooApp {  
4  
5     public void test(){
```

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro



And finally ...



Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# References



- Sommerville (1992) 'Software Engineering' 4e, Pearson.

Assignment Project Exam Help

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

Add WeChat edu\_assist\_pro

# Acknowledgement



- Slides based on material from

- Bill Leahy's lecture slides
  - <http://www.cc.gatech.edu> <https://eduassistpro.github.io/> [oly.ppt](#)
- Maria Litvin's & Gary Litvin'
  - <http://skylit.com/javamethods/ppt/Ch10.ppt>
- Marty Stepp's lecture slides
  - <http://www.cs.washington.edu/331/>
- And others ...

But I also contributed some stuff myself :-)