

# HW2

1. What is the difference between a class and an object? Provide a real-world example.

A class is a definition, it's like a blueprints; an object is the exact item constructed based on that blueprints.

2. If you create a parameterized constructor in a class, what happens to the default constructor? What must you do if you still need it?

default constructor would be invalid. You need to explicitly create a default constructor to overload it.

3. What are the four access modifiers in Java? List them from most restrictive to least restrictive.

private < default < protected < public

4. Explain the difference between method overloading and method overriding.

overloading: same method name, different parameters, in a same class.  
static polymorphism.

overriding: different implementation of parent class method in children class.  
dynamic polymorphism.

5. Can you override a final method? Can you override a private method? Explain why or why not.

No: final means it cannot be changed. Overriding relies on runtime polymorphism and would break the contract enforced by final

No: private means methods are not visible to subclasses. Subclasses cannot override a method they cannot access.

6. What is the difference between static polymorphism and dynamic polymorphism? When does each occur?

static polymorphism: determined in compile-time. based on reference type.

dynamic polymorphism: determined in runtime. based on exact object type.

7. Why does Java not support multiple inheritance with classes? How can you achieve multiple inheritance in Java?

to avoid the diamond problem

use interface.

8. Consider the following code:

```
List<Integer> lst = new ArrayList<>();
```

Which principle of OOP does this demonstrate? Explain.

Polymorphism: the code uses the `List` interface, allowing different list implementations like `ArrayList` or `LinkedList` to be substituted without changing the code.