

# CS 302: INTRODUCTION TO PROGRAMMING IN JAVA

---

## Lecture 14

# REVIEW – BASIC OOP

---

- What are some differences between procedural programming versus OOP?
- What is the difference between a class and an object?
- What is a public interface versus a private implementation?
- What is encapsulation?

# REVIEW – CREATING OBJECTS

---

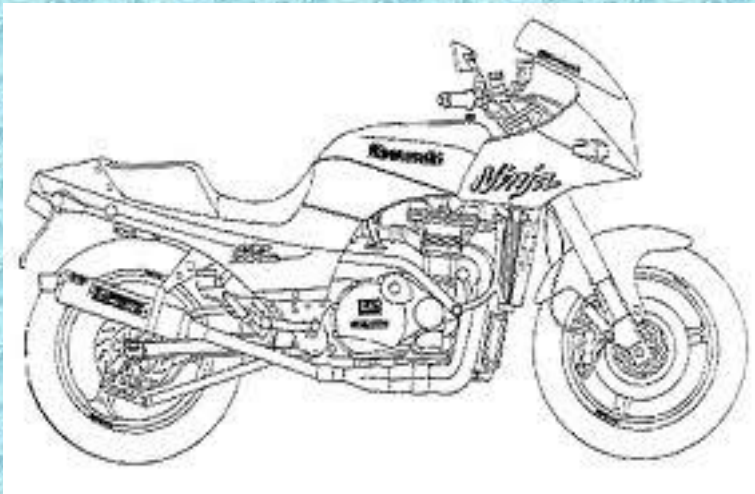
- What is an instance variable?
- What is an instance method and how are they called?
- What are the different types of instance methods?
- What is a constructor?
- How do we create an instance of an object?
- What does the general outline of a class look like?

# PRACTICAL EXAMPLE

---

- How can we implement a bank account object?
- What private instance data will we need?
- What sort of Accessors and Mutators will we need?
- What will the constructor look like?





Class

Instances of the class  
(objects) – only valid at  
runtime



# PRIVATE INSTANCE METHODS

---

- Instance methods usually public – why?
- If we have an internal function that we do not want others to be able to call, make it private
- Ex. a sort method on a phonebook object
  - Phonebook could have public methods to add and remove people from the phonebook but should keep itself sorted no matter what
  - Both the add() and remove() instance methods could call a private sort() method

# ACCESSING INSTANCE VARIABLES

---

- Global scope within the class – why?
- `this.varName` ALWAYS refers to the instance variable
- `varName` will refer to the instance variable if the `varName` is unique



# ACCESSING INSTANCE VARIABLES EXAMPLE

---

```
public class BankAccount
{
    private int balance;
    public BankAccount(int balance)
    {
        //what goes here?
    }
}
```



# CONSTRUCTORS

---

- Form: `public <ClassName> (<param list>)`
- Can have multiple constructors as long as they take in different parameters
  - The appropriate constructor will be called based on what arguments are passed in
- **Method Overloading: having multiple methods with the same name that take in different arguments**

# PRACTICE 1

---

- Create the class for a Motorcycle Object
- Motorcycles have:
  - Color
  - Current Speed
  - Number of Gears
- Motorcycles can:
  - Accelerate
  - Decelerate

# PRACTICE 2

---

- Create the class for a Chess Piece:
- Chess Pieces have:
  - Color
  - Type (pawn, rook, knight, bishop, queen, king)
  - A grid location (a1 – h8)
  - Status (live or dead)
- Chess Pieces can:
  - Move to a new location
  - Die

