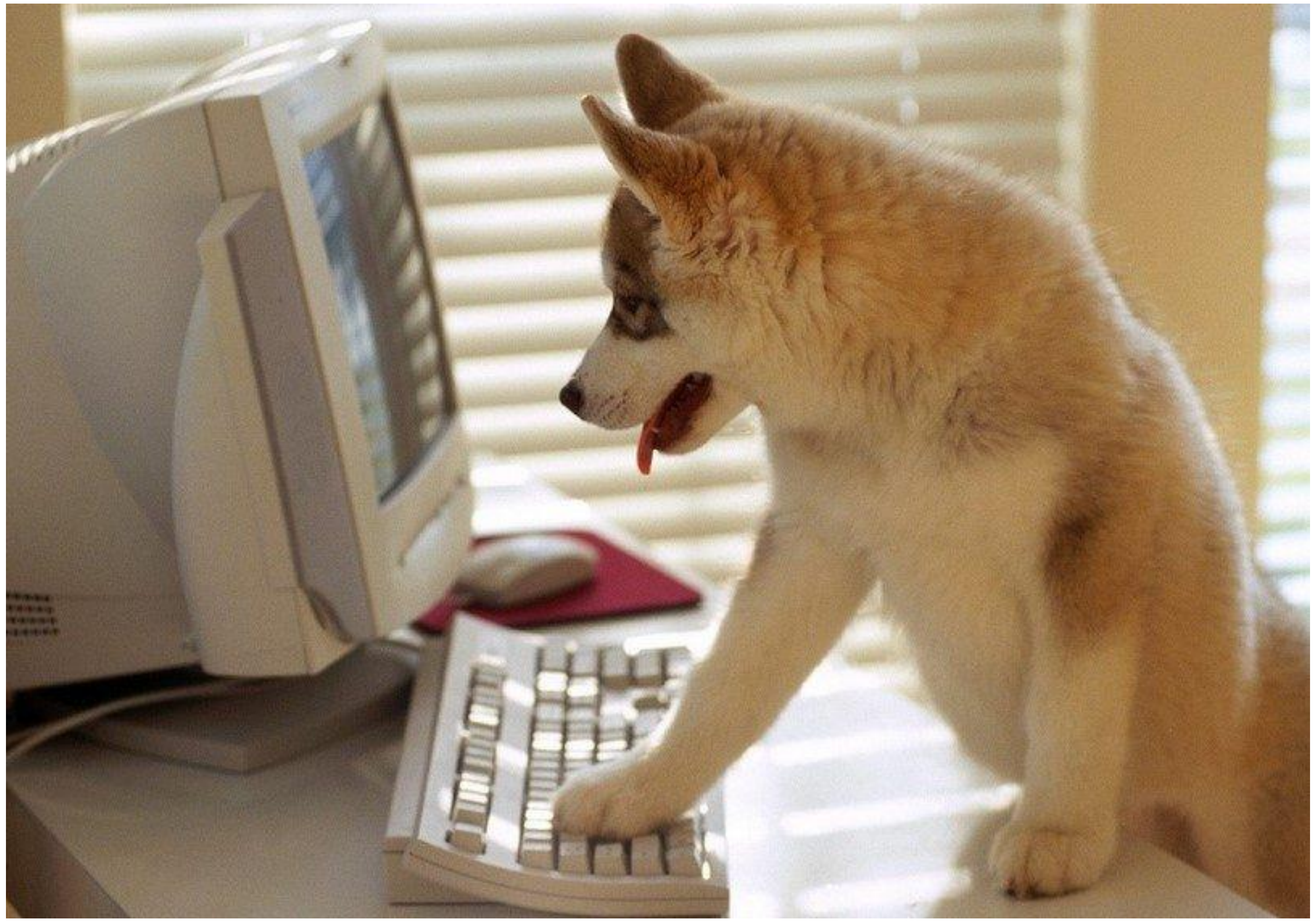
Java Programming



Organizational Stuff

18.03.: Structures
19.03.: Methods
20.03.: Recursion
21.03.: Arrays
22.03.: Strings

25.03.: OOP1
26.03.: OOP2
27.03.: Generics
28.03.: Exceptions & **Testing**
29.03.: GUI

Generics

Java is very strict about data types.
But what if we don’t know the data type or want to keep it flexible?

Generics

```
public class Example<T>{
  private T data;
  public Example(T data){
    this.data=data
  }
  public T getData(){
    return data;
  }
}
```

Generics

```
public class Example<T>{
   private T data;
   public Example(T data){
     this.data=data
   }
   public T getData(){
     return data;
   }
}
```

Generics

```
public class Example<T>{
  private T data;
  public Example(T data){
    this.data=data
  }
  public T getData(){
    return data;
  }
}
```

```
Example<String> ex = new Example<>(“Hello”);
System.out.println(ex.getData());
```

Generics

```
public class Example<T>{
   private T data;
   public Example(T data){
      this.data=data
   }
   public T getData(){
      return data;
   }
}
```

```
Example<String> ex = new Example<>(“Hello”);
System.out.println(ex.getData());
```

Generics

Real World Scenarios?

- ArrayLists
- HashMaps
- Collections
- ...

Generics

Today’s Assignment:
https://classroom.github.com/a/K6ER4dEj