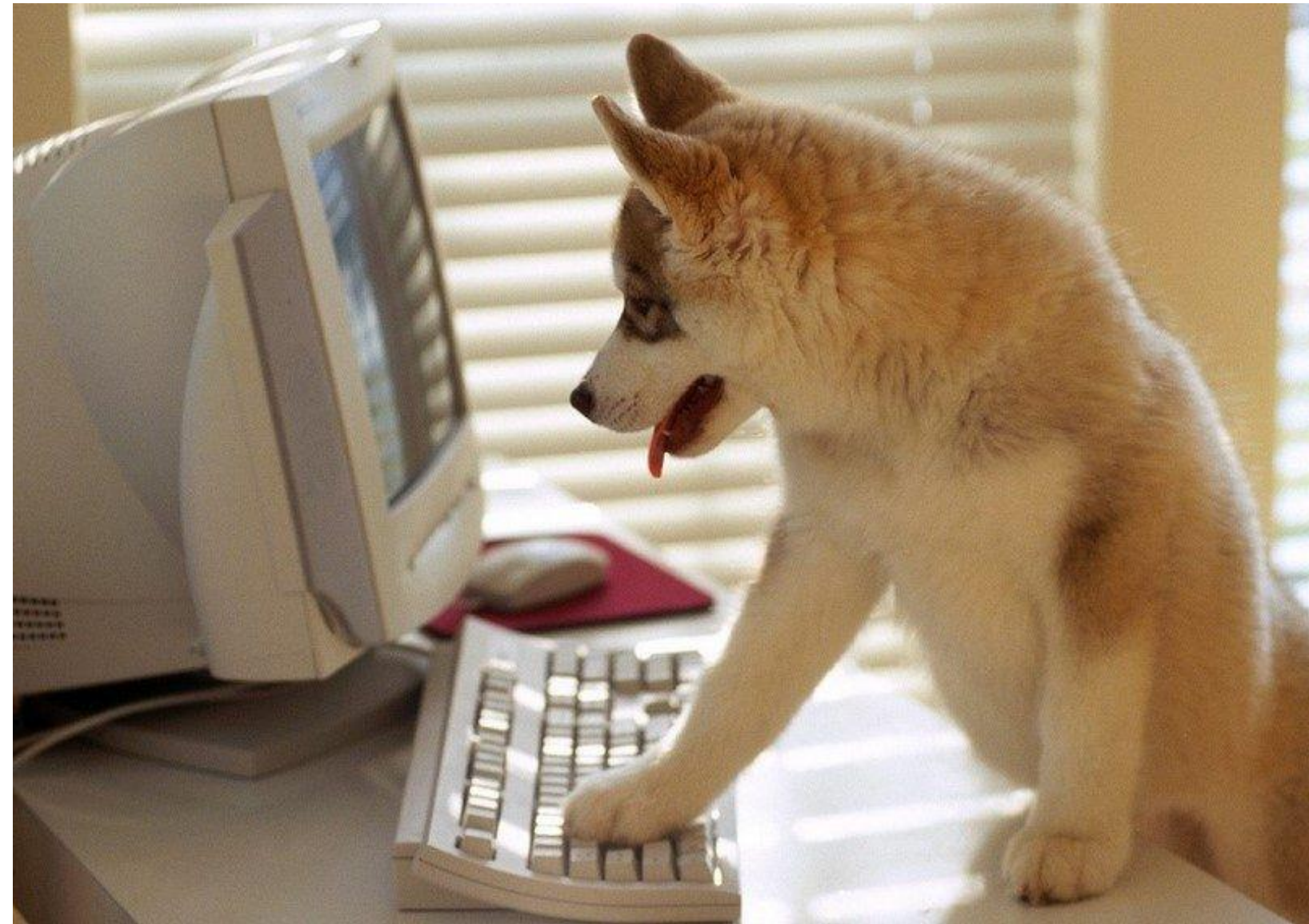


# Java Programming



# Organizational Stuff

23.09.: No Students :(

24.09.: Structures

25.09.: Methods & Recursion

26.09.: Arrays

27.09.: Strings

-----

30.09.: OOP

01.10.: Generics & Linked Lists

02.10.: Exceptions & Testing

03.10.: Holiday

04.10.: GUI

# Exceptions

```
String s = "eleven";  
int i = Integer.parseInt(s);
```

# Exceptions

```
String s = "eleven";  
try{  
    int i = Integer.parseInt(s);  
} catch (NumberFormatException e) {  
    e.printStackTrace();  
}
```

# Exceptions

```
public class SomeAwesomeException extends Exception{  
    public SomeAwesomeException(String message){  
        super(message);  
    }  
}
```

# Exceptions

```
public void doSomething(int x) throws SomeAwesomeException{  
    if(x<0){  
        throw new SomeAwesomeException("<0");  
    }  
}
```

# Testing

```
/*Initialize*/  
//Test owner  
@Test public static void InitTest1() {  
    //define  
    Person owner = new Person("John Doe");  
    BankAccount account = new BankAccount(owner, 1000.00);  
    //test  
    assertEquals(owner.getName(), account.getOwner().getName(), 0.001);  
}
```

# Testing

```
/*Initialize*/  
//Test owner  
@Test public static void InitTest1() {  
    //define  
    Person owner = new Person("John Doe");  
    BankAccount account = new BankAccount(owner, 1000.00);  
    //test  
    assertEquals(owner.getName(), account.getOwner().getName(), 0.001);  
}
```



# Testing

```
/*Initialize*/  
//Test owner  
@Test public static void InitTest1() {  
    //define  
    Person owner = new Person("John Doe");  
    BankAccount account = new BankAccount(owner, 1000.00);  
    //test  
    assertEquals(owner.getName(), account.getOwner().getName(), 0.001);  
}
```

# Testing

Test for:

- Normal Cases
- Edge Cases
  - Null
  - Double.MAX\_VALUE
  - Empty Array
  - ...

# Exceptions and Testing

**Today's Assignment:**

<https://classroom.github.com/a/cohxNMBm>

