

What is the difference between a JDK and a JRE?

"JRE" – for **users**. Java Runtime Environment. An implementation of the Java Virtual Machine which **executes** Java programs.

"JDK"– for **developers**. Java Development Kit, used to **develop** Java based software. Contains JRE(s), compiler, debuggers, dev libraries etc. **You need a JDK to compile**

Where are they located? (windows command line demo)

```
C:\Users\Perkins>java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)

C:\Users\Perkins>where java
C:\Windows\System32\java.exe
C:\Program Files\Java\jdk1.7.0_51\bin\java.exe
```

Interfaces

Used to define classes of behavior

Define a type of behavior

- Abstract (methods empty,
- derived classes fill in)
- *You cannot instantiate an interface.*
- *An interface does not contain any constructors.*
- *All of the methods in an interface are abstract.*
- *An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final.*
- *An interface is not extended by a class; it is implemented by a class.*

```
interface animal {  
    public void eat();  
    public void travel();  
}
```

```
public class Mammal  
implements animal{  
}
```

Show how you are forced to
override methods or make
class abstract

Demo using Animation.AnimationListener in a class

Don't use magic numbers

```
@Override
public void resume() {
    //problem here, 10 is a magic number,
    //10 what? what does it mean?
    initDeals(10);
}
```



Bad

```
//better idea, define a constant
//make it static so it is only allocated once
//not every time enclosing object allocated
//make it final so it cannot be changed
public static final int NUMBER_DEFAULT_DEALS = 10;
```

```
@Override
public void resume() {
    initDeals(NUMBER_DEFAULT_DEALS);
}
```



Good

Also note that the name is all caps, convention states that a variable in all caps is a constant

Consider defining constants in 1 place

```
public final class Constants {  
  
    public static final boolean PASSES = true;  
    public static final boolean FAILS = false;  
  
    //static helper class do not  
    //need to be constructed  
    private Constants(){ }  
}  
  
//usage  
boolean myGrade = Constants.PASSES;
```