Review of Java

… *or maybe not*

Complete this class

```
public _____ Foo _____ Bar _____ Comparable {
    /** name of this Foo */
    private String name;
    /** initialize a new Foo */
    public ____ Foo(String name) {
       _______________ =  ________
    }
    public String toString() { return name; }

    // what other method is required?
}
```

Make the name *immutable*

```
public _____ Foo _____ Bar _____ Comparable {
    /** name of this Foo */
    private _________  String name;
    ...
```

immutable means you cannot change the value after it is set the first time.

Write an access & mutator for `size`

```
public class TShirt {
    /** the size of the t-shirt (inches) */
    private int size;
    /** Get the size of t-shirt (in inches) */
    public ______ ____________________ {

    }

    /** Set the size of this t-shirt. */
    public ______ ________________ {

    }
}
```

What is the difference between left & right?

```
boolean
int
double
char
```

```
String
Integer
Date
Scanner
System
```

Name these Primitives

_____ 32-bit whole numbers -9999 …. 0, 1, 0x10

_____ 64-bit whole numbers, written like 4L

_____ 'ก', 'ด', \u0420

_____ 2.98E+08

_____ 8-bit values 0, 1, …, 255 (not int values)

What are the other 3 primitive types?

_____ _____ _____

Wrapper Classes

Primitive	Wrapper
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

```
double root = Math.sqrt( 2.0 );
Double d1 = new Double( root );
// same thing: automatic boxing
Double d2 = root;


// print as a string
out.println(  d2.toString( ) );
// static method to make a string
out.println( Integer.toString( 2 ) );
```

Numeric Constants

1. What is the largest possible "int" value?

```
int max = _____;
```

2. What happens if you go beyond that?

```
int tooBig = max + 1;
```

a) exception thrown

b) error

c) converts to "long"

d) wrap around to negative value

3. What are the largest and smallest (magnitude) double values?

What is the result?

```
> char c = 'A';
> c+1
> ++c
> (short) c
> (byte) c
> (char) 66
```

(int) c means "convert the value to an int".
This is called a *cast*.

You can use class name in a cast, too:

(Character) c

What is the result?

> c = 'ก' ; (*gau gai)*

> (int) c

> ++c

> (char) c

Java uses Unicode for char and Strings,
but the output may not be readable if the output device doesn't use Unicode

How are these different?

```
// 1 Billion + 2 Billion
> 1000000000 + 2000000000
> 1000000000L + 2000000000L
> 1E9 + 2E9


// in Java 7 you can write _ in numbers
> 1_000_000_000L + 2_000_000_000L
```

Bizarre Numbers

```
System.out.println( 12 );
```

```
System.out.println( 012 );
```

```
System.out.println( 0x12 );
```

```
System.out.println( 012 + 0x12 );
```

Which data type should you use for ...

```
// the day of the month
________  day = 13;  // 13th of Jan
// population of the world
________  worldPop = (7 billion) ;
// Bank account number
// example: 001230055555
________ accountNumber = . . .
```

How to Convert Primitive to Object?

A List (like ArrayList) can only contain *objects*.

How can we add *primitive values* (like int) to a List?

```
List mylist = new ArrayList( );
int n = 51651111;
mylist.add( n ); // Wait? How is this possible?
```

Try in BlueJ Codepad:

```
> List list = new ArrayList( );
> list.add( "hello" )
> list.add( 10 )   // what is being put in the List?
> list.get( 0 )
"apple"   (String)
> list.get( 1 )
<object reference>  (Integer)
```

Methods to convert to/from String

```
int n = 29*31;

// convert n to a String

String product = Integer.toString(n);

// parse integer value of a String

String s = "123";

int m = Integer._____ (s);

// parse double value of String s

double d = _____;
```

parseInt(string) versus valueOf(string)

What is the difference?

```
String s = "123";

Integer.parseInt( s )


Integer.valueOf( s )
```

Useful Constants in Numeric Wrapper classes

1. What is the largest "int" value?
2. What is the smallest "long" value?
3. What is the range (smallest, biggest) of double?

```
int maximum =

long minimum =

double minsize =

double maxsize =
```

What value is after the biggest value?

```
int n = Integer.MAX_VALUE;
n = n + 1;
System.out.println( n );
double d = Double.MAX_VALUE;
d = d + 1;
System.out.println( d );
d = d * 1.000001;
System.out.println( d );
```

Packages

- Java uses packages to organize classes.
- Packages reduce size of *name space* and avoid *name collisions* (like `Date` in `java.util` and `java.sql`).

Q: Which package contain these classes?

Java language core classes (Object, String, System, ...).
You never have to import this.

Classes for input and output, like InputStream, FileReader

Date classes and collections (List, ArrayList)

Utilities Scanner, Arrays,

Java Graphics frameworks (2 packages)

Name these Packages

	Java language core classes `Object, String, System, Integer, Double, Math, Thread.`
	Classes for input and output `InputStream, BufferedReader, File`
	Date/time classes, collections, utilities `Calendar, Date, List, ArrayList, Set, Arrays, Formatter, Scanner`
	java compiler always "import"s this package, so you don't need to.

Useful Packages added in Java 8

`java.datetime`	New date and time classes.
`java.util.stream`	Classes and interfaces for "streams" style programming. A very cool and handy feature, found in other languages.

Packages

What package are these classes in?

String …

System ...

Scanner …

Date …

InputStream and FileReader …

What is the output?

```
System.out.println( 3 + 4 );
```

```
System.out.println( "3" + 4 );
```

```
System.out.println( '3' + 4 );
```

```
System.out.println( 3 + "4" );
```

Bit Operations

```
> int a = 7;
> int b = 10;
> a & b
> a | b
> a ^ b
> a == b
> a = b
> a && b
```

Arguments are passed by value

```
public static void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
// elsewhere in the code...
int m = 10;
int n = 20;
swap( m, n );
What is m?
```

Person class

Person
- name: String
<<constructor>> Person(name: String) getName( ): String setName(newname: String): void toString(): String

```
Person p = new Person( "Pee" );
p.setName( "Nong" );
System.out.println( p.toString() );  // prints "Nong"
```

Passing arguments, again

```
public void swap(Person a, Person b) {
    Person temp = a;
    a = b;
    b = temp;
}
// elsewhere in the code...
Person m = new Person( "Meaw" );
Person n = new Person( "Nok" );
swap( m, n );
What is m.toString() ?
```

How about this?

```
public void swapName(Person a, Person b) {
    String temp = a.getName();
    a.setName( b.getName() );
    b.setName( temp );
}
// elsewhere in the code...
Person m = new Person( "Meaw" );
Person n = new Person( "Nok" );
swapName( m, n );
What is m.toString() ?
```

Difference between "==" and .equals?

```
> Double x = new Double(10);
> Double y = new Double(10);
> x == y
> String s = "yes";
> String t = "yes";
> s == t
> String u = new String("yes");
> s == u
> s.equals(u)
```

How to write equals( )

You should usually define **`equals( )`** like this:

```
public class Person {
     public boolean equals( Object other ) { … }
```

Not like this:

```
    public boolean equals( Person other ) { ... }
```

JAR files

What is a JAR file?

Why use them?

How to create one?

WHERE ARE THE JDK CLASSES?

Classes in the Java SE API:

- 4,024 in java 7
- 3,793 in java 6
- 3,279 in java 5.0

Actually there are MORE classes that this – some classes are not documented in the API. And this number does not include *interfaces*.

These classes are on your computer (in the JDK). Where are they?

BlueJ IDE Layout