



# Java Puzzlers

---

From the book **Java Puzzlers**  
by Joshua Bloch and Neal Gafter





# Oddity

---

- The following method tests whether its argument is odd:
- ```
public static boolean isOdd(int i) {  
    return i % 2 == 1;  
}
```
- Does it work?
- It gives the correct answer for  $\frac{3}{4}$  of the integers



# Making change

---

- ```
public class Change {  
    public static void main(String[] args) {  
        System.out.println(2.00 - 1.10);  
    }  
}
```

- 0.89999999999999999999



# Long Division

---

- ```
public class LongDivision {  
    public static void main(String[] args) {  
        final long MICROS_PER_DAY = 24 * 60 * 60 * 1000 * 1000;  
        final long MILLIS_PER_DAY = 24 * 60 * 60 * 1000;  
        System.out.println(MICROS_PER_DAY / MILLIS_PER_DAY);  
    }  
}
```

- 5



# Addition

---

- ```
public class Addition {  
    public static void main(String[] args) {  
        System.out.println(12345 + 54321);  
    }  
}
```

- **17777**



# Tweedledum

---

- Declare variables `x` and `i` such that

`x += i;`

is legal, but

`x = x + i;`

is not legal

- `short x = 0;`  
`int i = 123456;`



# Tweedledee

---

- Declare variables `x` and `i` such that

`x = x + i;`

is legal, but

`x += i;`

is not legal

- `Object x = "Hello ";`  
`String i = "world!";`



+ =

---

- ```
public class PlusEquals {  
    public static void main(String[] args) {  
        int i = 2;  
        i += 3.75;  
        System.out.println(i);  
    }  
}
```

- 5





# Last Laugh

---

- ```
public class LastLaugh {  
    public static void main(String[] args) {  
        System.out.print("H" + "a");  
        System.out.print('H' + 'a');  
    }  
}
```

- Ha169



# Indecision

---

- ```
public class Indecisive {  
    public static void main(String[] args) {  
        System.out.println(decision());  
    }  
}
```

```
    static boolean decision() {  
        try {  
            return true;  
        }  
        finally {  
            return false;  
        }  
    }  
}
```

- false



# HelloGoodbye

---

- ```
public class HelloGoodbye {  
    public static void main(String[] args) {  
        try {  
            System.out.println("Hello world!");  
            System.exit(0);  
        }  
        finally {  
            System.out.println("Goodbye world!");  
        }  
    }  
}
```

- Hello world!



# The reluctant constructor

---

- ```
public class Reluctant {
    private Reluctant internalInstance = new Reluctant();

    public Reluctant() throws Exception {
        throw new Exception("I'm not coming out!");
    }
    public static void main(String[] args) {
        try {
            Reluctant b = new Reluctant();
            System.out.println("Surprise!");
        }
        catch (Exception e) {
            System.out.println("I told you so.");
        }
    }
}
```
- Exception in thread "main" java.lang.StackOverflowError



# Hello again

---

- ```
public class Null {  
    public static void main(String[] args) {  
        ((Null)null).greet();  
    }  
  
    public static void greet() {  
        System.out.println("Hello world!");  
    }  
}
```
- Hello world!



# The End

---

