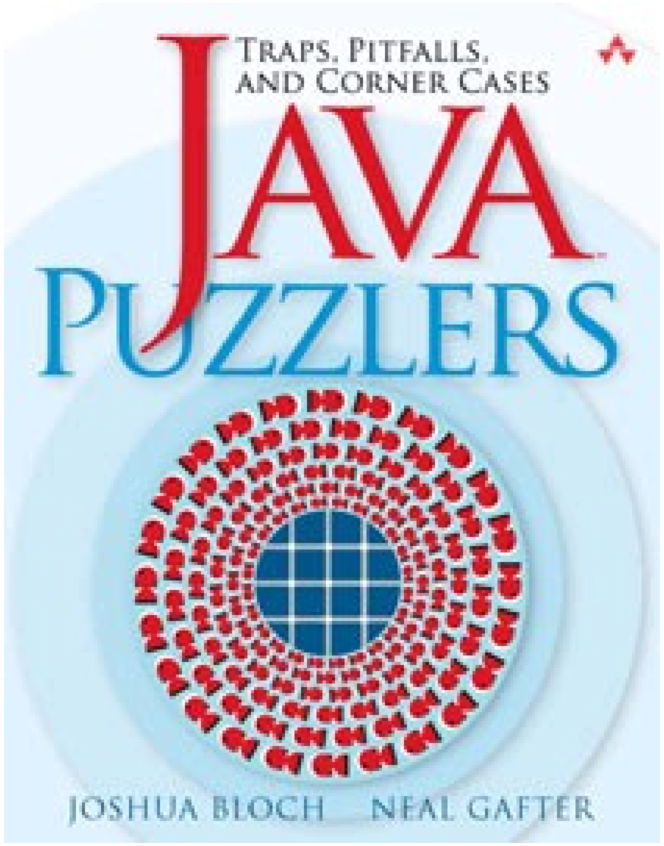
How good are your Java skills?

Today

- Java traps, pitfalls, puzzles and problems.
- A tour of things that go wrong



- Some examples are taken from *Java Puzzlers* by Bloch and Gafter
- Definitely worth a read, no matter what level of programmer you are
- **Paperback:** 312 pages
- **Publisher:** Addison Wesley (21 Jul 2005)
- **ISBN-10:** 032133678X
- **ISBN-13:** 978-0321336781

PolyPain

```
public class PolyPain {
    public String name = "Parent";
    public void Print() { System.out.println("Parent"); }
    public static void Print2() { System.out.println("Parent"); }
}

public class PolyPainChild extends PolyPain {
    public String name = "Child";
    public void Print() { System.out.println("Child"); }
    public static void Print2() { System.out.println("Child"); }
}

public static void main(String[] args) {
    PolyPainChild c = new PolyPainChild();
    PolyPain p = (PolyPain)c;
->  p.Print();    — Child
    p.Print2();   — Parent
    System.out.println(p.name);   — Child
}
```

A. "Parent"
B. "Child"

PolyPain

- Overridden methods exhibit dynamic polymorphism
- Overridden static methods do *<u>not</u>*
- Overridden (“shadowed”) fields do *<u>not</u>*

Even or odd?

```
public static boolean isOdd (int x) {
    return (x % 2 == 1);
}
```

A. Works just fine
B. Works for negative x only
C. [✓ Works for positive x only]
D. Fails all the time
E. I don’t care

Even or Odd?

- Java defines % as: (a / b) * b + (a % b) = a
- So if a<0, b>0 then (a % b) < 0.
- i.e (-7 % 2) = -1 and not 1!
- Fixes:

```
public static boolean isOdd (int x) {
   return (x % 2 != 0);
}


public static boolean isOdd (int x) {
   return (x & 1 != 0);
}
```

Can Java do simple maths?

```
public static void main (String[] args) {
    System.out.println(12+21);
}
```

A. 32
B. 31
C. 14
D. 7
E. Still don’t care

Can Java do simple maths?

```
public static main (String[] args) {
    System.out.println(12+2l);
}
```

Meant to be (int)12 + (long)2. The problem is that Java doesn’t enforce the use of capital “L”, which causes confusion in certain fonts!

Moral: Always use “L”

Can Java do simple maths 2?

```
public class CanJavaDoMaths2 {

    public static void main(String[] args)
     {
        long prod = 1000000*5000;
        System.out.println(prod);
     }
}
```

A. 5000000000
B. 5000
C. 23560043
D. 705032704
E. Something else
F. Seriously, I don’t care

Can Java do simple maths 2?

```
public class CanJavaDoMaths2 {

     public static void main(String[] args)
{

     long prod = 1000000*5000;
     System.out.println(prod);

    }
}
```

Same as:

```
int x = 1000000;
int y = 5000;
int xy = x*y;    // This overflows!
long prod = xy;
```

Can Java do simple maths 2?

```
public class CanJavaDoMaths2 {

    public static void main(String[] args)
{

    long prod = 1000000L*5000;
    System.out.println(prod);

   }
}
```

Can Java do simple maths 3?

```
public class CanJavaDoMaths3 {
    public static void main(String[]
args) {
        double x = 2.0;
        double y = 1.1;
        System.out.println( x - y );
    }
}
```

A. 0
B. 0.9
C. Something else
D. Are you not hearing me? I’m not interested

Can Java do simple maths 3?

- The problem here is that powers of ten (which we use so often for currency etc) can’t be represented exactly using binary point
  - 1.1 = 11x10^(-1) [decimal]
  - 11d = 1011b
  - We want **e** s.t. 10^(-1) = 2^(e)
  - log(10^(-1)) = e log(2)
  - e = -1/(log 2), but (log 2) is irrational!!
- In this case, 1.1 gets represented as the nearest double
  - Then we subtract
  - The answer is rounded to the nearest double, which happens to be the ugly thing you’ve just seen.
- Moral: Use integers where you can!!!

Can Java do simple maths 4?

```
public class CanJavaDoMaths4 {
    public static void main(String[] args) {
        int x = 10 + 010;
        System.out.println(x);
    }
}
```

A. 20
B. 18
C. 11
D. Something else.
E. Hmmm.. I wonder how rude I can be on the feedback form?

Can Java do simple maths 4?

```
public class CanJavaDoMaths4 {
   public static void main(String[]
args) {
      int x = 10 + 010;
      System.out.println(x);
   }
}
```

If you prefix an integer with a zero, Java interprets it as being in octal (base-8) rather than in decimal!

010o = 8d

Can Java do simple maths 5?

```
public class CanJavaDoMaths5 {
    public static void main(String[] args) {
        double x = 1.0 / 2L;
        System.out.println(x);
    }
}
```

A. 0.5
B. 0.0
C. 1.0
D. Something else.
E. Where did I put that copy of Varsity?

Can Java do simple maths 5?

```
public class CanJavaDoMaths5 {
    public static void main(String[] args) {
        double x = 1.0 / 2L;
        System.out.println(x);
    }
}
```

Just testing – there’s nothing unexpected going on here!

Java's Gone Loopy

```
for (long i = Long.MAX_VALUE-5;
        i<=Long.MAX_VALUE;
        i++) {
    System.out.println("Hello");
}
```

A. 4x
B. 5x
C. 6x
D. 100x
E. Never stops
F. At least this is the last of these silly lectures

Loopy

```
for (long i = Long.MAX_VALUE-5;
        i<=Long.MAX_VALUE;
        i++) {
    System.out.println(“Hello”);
}
```

Always true.
Should have used
< and not <=

Mad Modulo

```
public class MadModulo {
    public static void main(String[] args) {

        int x = 11 % 2*5;
        System.out.println(x);

    }
}
```

A. 1
B. 0
C. 10
D. 5
E. Something else
F. Is it lunch time yet?

Mad Modulo

```
public class MadModulo {
    public static void main(String[] args) {

        int x = 11 % 2*5;
        System.out.println(x);

    }
}
```

Operator precedence is the same for % and *

So Java does (11 % 2)*5 = 5

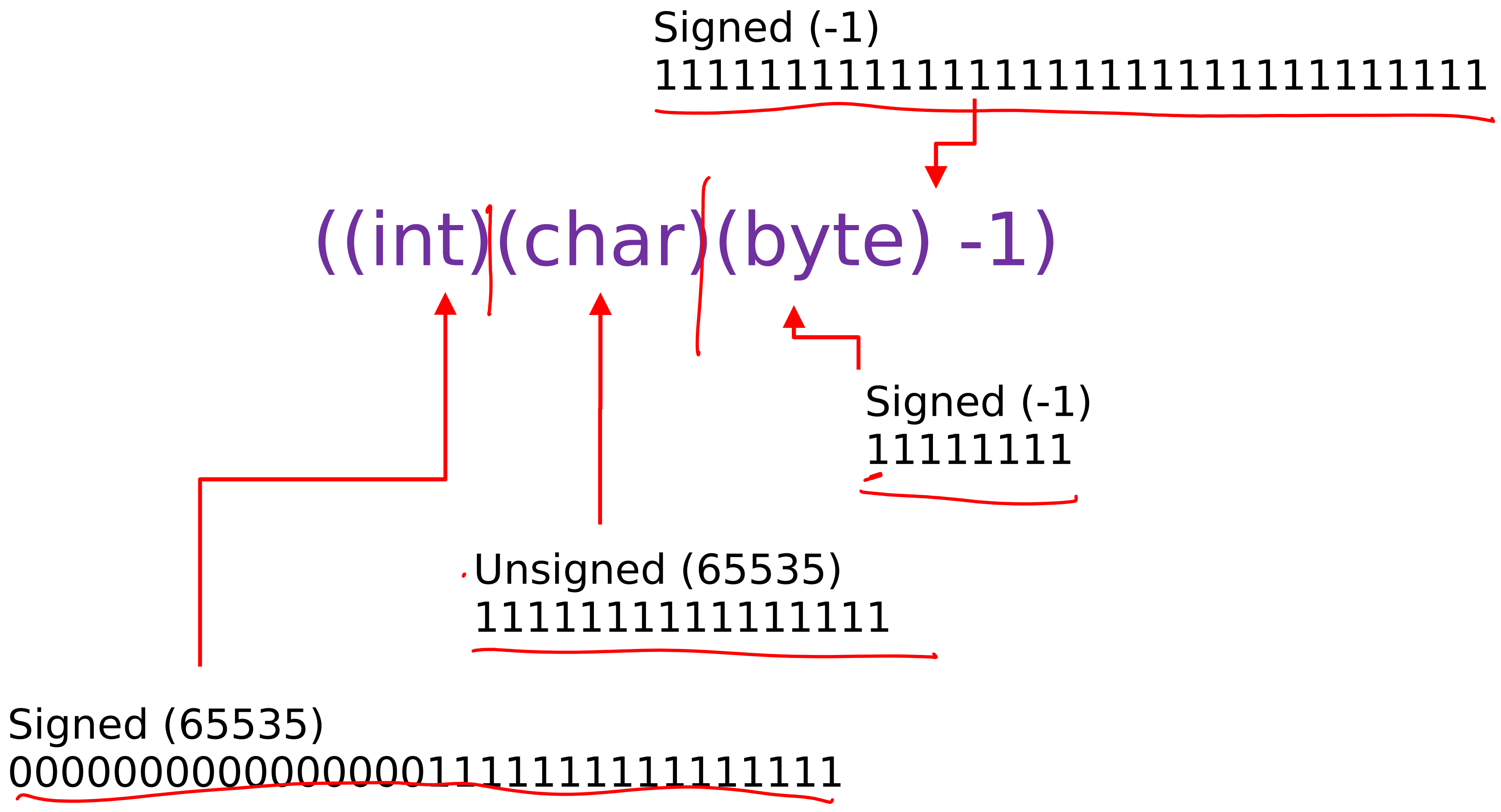
We should have been explicit: 11%(2*5);

Cast-igation

```
((int)(char)(byte) -1)
```

A. 0
B. -1
C. 255
D. 65535
E. Something else
F. ZZZzzz...

Cast-igation



Rule: Sign extension is performed if the original value is signed. Otherwise zero extension.

Cast-igation Part II

```
Student p = new Student();
Object o = (Object)p;
Sausage b = (Sausage)o;
```

A. Won’t compile
B. Gives ClassCastException
C. Runs fine
D. Something else
E. Aaaarrrggghhh…

Cast-igation Part III

```
Student p = null;
Object o = (Object);p;
Sausage b = (Sausage)o;
```

A. Won’t compile
B. Gives ClassCastException
C. Runs fine
D. Something else
E. Aaarrggghhh...

Cast-igation Part III

```
Student p = null;
Object o = (Object);
Sausage b = (Sausage)o;
```

It turns out that Java lets us cast null to anything we like without throwing an error!!!

TryHarder

```
public class TryHarder {
    public static boolean test() {
        try {
            return true;
        }
        finally {
            return false;
        }
    }

    public static void main(String[] args) {
        System.out.println(test());
    }
}
```

A. True
B. False
C. 42
D. Would he notice if I snuck out?

TryHarder

```
public class TryHarder {
    public static boolean test() {
        try {
            return true;
        }
        finally {
            return false;
        }
    }

    public static void main(String[] args) {
        System.out.println(test());
    }
}
```

finally will *always* run when the function surrenders control so the “return true” is overridden

PlusPlusPain

```
int j=0;
for (int i=0; i<100; i++)  j=j++;
System.out.println(j);
```

A. 100
B. 99
C. 0
D. 1
E. Something else
F. Arrggghhh.. Let me out of here!

PlusPlusPain

```
int j=0;
for (int i=0; i<100; i++)  j=j++;
System.out.println(j);
```

This is a *postfix* operator. That means this is the same as:

```
int j2 = j;
j = j + 1;
j = j2;
```

Lesson: Don’t assign the same variable more than once per line

PlusPlusPain

```
int j=0;
int i = (j++) + (j++);
System.out.println(i+" "+j);
```

A. 0 1
B. 1 2
C. 0 2
D. 0 0
E. Something else
F. This *is* the last lecture, right?

Strung out

```
public class StrungOut {
    public static void main(String[] args) {
        System.out.print(“R”+“2”);
        System.out.print(‘D’+‘2’);
    }
}
```

A. R2D2
B. R2
C. Something else which I could figure out if I wanted to
D. No idea, but you’re probably trying to trick us
E. *< head-butting table repeatedly >*

Strung out

‘A’ is a char (an unsigned 16-bit number)
“A” is a string

char is treated as a number for the + operator. So when two chars are added, we get a numerical result.

Shift Shame

```
public class ShiftShame {
    public static void main (String[] args) {

        long x = 1 << 32;
        System.out.println(x);
    }
}
```

A. 0
B. 255
C. 1
D. Something else
E. I could be home in bed right now

Shift Shame

```
public class ShiftShame {
    public static void main (String[] args) {

        long x = 1 << 32;
        System.out.println(x);
    }
}
```

LHS treated as an int

But isn’t (1<<32) all zeroes, so the answer should have been 0?

The java shift operator performs shifts by modulo-32 (int) or modulo-64 (long) amounts.

(1<<32) is therefore (1<<0)

ClassyConundrum

```
public class ClassyConundrum {

    public String mString = "CS is fun";

    public void ClassyConundrum() {
        mString = "CS is dull";
    }

    public static void main(String[] args) {
        ClassyConundrum cc = new ClassyConundrum();
        System.out.println(cc.mString);
    }
}
```

A. CS is fun
B. CS is dull
C. Something else
D. Surely there can’t be any more… can there?

ClassyConundrum

```
public class ClassyConundrum {

	public String mString = "CS is fun";

	public void ClassyConundrum() {
		mString = "CS is dull";
	}

	public static void main(String[] args) {
		ClassyConundrum cc = new 
ClassyConundrum();
		System.out.println(c.mString);
	}
}
```

Moral: don’t name your methods after your class (unless they are constructors!)

Enough Already!

Course Review

Computer Basics
Virtual Machines
Compilation
Pointers
References

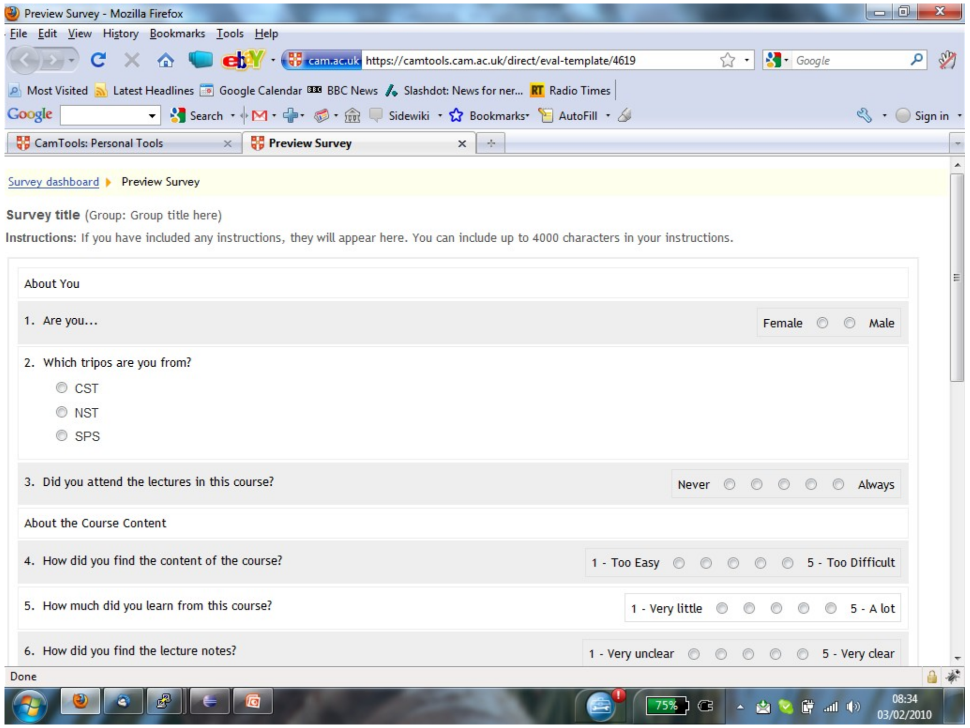
OOP Fundamentals
Modularity
Encapsulation
Inheritance
Polymorphism
UML Class Diagrams
Abstract Classes
Interfaces [Java]
Constructors
Destructors
Static Data and Methods

Java
Cloning Objects
Comparing Objects
Packages
Collections
Generics

Design Patterns
Decorator
State
Strategy
Composite
Singleton
Proxy
Observer
Abstract factory

Course Feedback

- You will get an email request to fill in a feedback questionnaire on this course.
- PLEASE fill it in


Preview Survey - Mozilla Firefox
File Edit View History Bookmarks Tools Help
cam.ac.uk https://camtools.cam.ac.uk/direct/eval-template/4619
Google
Most Visited Latest Headlines Google Calendar BBC News Slashdot: News for ner... Radio Times
Search Sidewiki Bookmarks AutoFill Sign in
CamTools: Personal Tools
Preview Survey
Done
75%
08:34
03/02/2010

Survey dashboard ▸ Preview Survey

Survey title (Group: Group title here)

Instructions: If you have included any instructions, they will appear here. You can include up to 4000 characters in your instructions.

About You

1. Are you... Female ○ ○ Male

2. Which tripos are you from?
   - ○ CST
   - ○ NST
   - ○ SPS

3. Did you attend the lectures in this course? Never ○ ○ ○ ○ ○ Always

About the Course Content

4. How did you find the content of the course? 1 - Too Easy ○ ○ ○ ○ ○ 5 - Too Difficult

5. How much did you learn from this course? 1 - Very little ○ ○ ○ ○ ○ 5 - A lot

6. How did you find the lecture notes? 1 - Very unclear ○ ○ ○ ○ ○ 5 - Very clear

Thank you for your attention
(or for not snoring at least)