On course materials
a workshop for L&S TA training

William C. Benton
University of Wisconsin

willb@cs.wisc.edu
http://pages.cs.wisc.edu/~willb/
About me
This talk: “why” and “how”
This talk: “why,” “what,” and “how”
Why make course materials?
Forecast

• When and how to use slides
• Why and how to make handouts
• How to make your life easier
• Teaching the Facebook generation
Forecast

• When and how to use slides
• Why and how to make handouts
• How to make your life easier
• Teaching the Facebook generation
What makes for good slides?
What makes for bad slides?

- One characteristic of bad slides is that they often have too much text or serve as an outline for a lazy speaker who has failed to adequately prepare, meaning that the audience will be reading instead of listening.
- Also feature many sentence fragments.
- Because font size is too small, readability suffers, key points not reinforced, audience asleep.
- Fortunately, you can print this out and read it later, gathering much of benefit of attending talk.
- Unfortunately, slides like this make you wonder why you are bothering to attend the presentation in the first place.
Other kinds of bad slides?

• Sure!
  • Many ways
  • Too terse perhaps as bad as too verbose
• Kind of too bad that “verbose” is on a line by itself up there; is it lonely?
Other kinds of bad slides? (ctd.)

- Does this help my presentation?
  - Not really.
- Inconsistent punctuation is sort of fun
- What information do these slides convey?
  - Why aren’t these bullets in my notes?
Trying to hide bad slides (2007)

- hidden via nonsense
- left sadly unadorned

- unbearable
- terrible
- tedious
- tiresome
Yikes!
Bad slides are ubiquitous

Engineering and Work Practice Controls (con’t)

- The employer must:
  - Train employees to use new devices and/or procedures
  - Document in ECP

Results

- **Attitudes**
  - People choosing to live near forests accept risks: Agree (%) 91
  - Homeowners should follow gov’t guidelines to manage for wildfire risk: Agree (%) 80

(“credits”: OSHA, USDA, DOL)
A better model
A better model
A better model
A better model
A better model
A better model
A better model
What should slides do?
Case study: figures
DA-DA-DA
DAAAH!
DA-DÁ-DA-DAAH!

BASSOONISTS
WAKE UP HERE

DUH-DÚH-DÚH
DUUH!
Case study: animated processes
public class Foo {
    private Foo f;
    private int k;

    public Foo(int k) {
        this.k = k;
        this.f = this;
    }

    public void sF(Foo f) {
        this.f = f;
    }

    public static void main(String args) {
        Foo f1, f2, f3, f4;
        f1 = new Foo(1);
        f2 = new Foo(2);
        f3 = new Foo(3);
        f4 = new Foo(4);
        f2.sF(f4);
        f4 = f1;
        // BANG
    }
}
public class Foo {
    private Foo f;
    private int k;

    public Foo(int k) {
        this.k = k;
        this.f = this;
    }

    public void sF(Foo f) {
        this.f = f;
    }

    public static void main(String args) {
        Foo f1, f2, f3, f4;
        f1 = new Foo(1);
        f2 = new Foo(2);
        f3 = new Foo(3);
        f4 = new Foo(4);

        f2.sF(f4);

        f4 = f1;
        // BANG
    }
}
What else are slides good for?
Forecast

• When and how to use slides
• Why and how to make handouts
• How to make your life easier
• Teaching the Facebook generation
One problem with slides

I can't see how you expect me to learn if you don't give me the slides in advance.
(Yes, this actually happened.)
The dirty little secret...

...is that I don’t want to distribute my slides at all!
The dirty little secret...

...is that I don’t want to distribute my slides at all!
The dirty little secret...
...is that I don’t want to distribute my slides at all!

```java
public class Foo {
    private Foo f;
    private int k;

    public Foo(int k) {
        this.k = k;
        this.f = this;
    }

    public void sF(Foo f) {
        this.f = f;
    }

    public static void main(String args) {
        Foo f1, f2, f3, f4;
        f1 = new Foo(1);
        f2 = new Foo(2);
        f3 = new Foo(3);
        f4 = new Foo(4);
        f2.sF(f4);
        f4 = f1;
        // BANG
    }
}
```
The dirty little secret...

...is that I don’t want to distribute my slides at all!
The quandary
Handouts are the solution
Handouts are the solution

- The employer must:
  - Train employees to use new devices and/or procedures
  - Document in ECP
Handouts are the solution

Engineering and Work Practice Controls (con’t)

- The employer must:
  - Train employees to use new devices and/or procedures
  - Document in ECP

Handout for CS 302 by William C. Benton (willb@cs)

Why integer overflow "wraps around"

The whole number types in Java (e.g., long, etc.) have limited ranges. Of course, any type representable in a finite computer has a limited range, but you’re likely to actually run into the limits of the data types you have to deal with in a manner that you can’t represent in computer memory. Manipulating a variable so that its value would exceed the range of its type results in integer overflow. When this happens, the variables "wrap around" to the other end of the range, just as a classic video game character might wrap around to the other side of the screen upon crossing an edge.

This handout will explain why this happens. To do so, we’ll first need to consider how computers represent numbers. (If you find this interesting, you’ll enjoy a computer organization class!)
Handouts are the solution

Why integer overflow "wraps around"

The whole number types, like long, are limited ranges. Of course, any type representable in a finite computer has a limited range, but you're likely to actually run into the limits of the data types types before you have to deal with a number that you can't represent in your computer.

Manipulating a variable so that its value would exceed the range of its type can lead to integer overflow. When this happens, the number "wraps" around to the other side of the range. This is just as a classic slide game character might wrap around to the other side of the screen upon crossing an edge.

This handout will explain why this happens. To do so, we'll first need to understand how computers represent numbers. If you find this interesting, you may enjoy a computer organization class.

Why integer overflow "wraps around"

Let's examine two different four-digit numbers to see the difference:

Let's represent the number 9999 (decimal) in binary:

\[
\begin{align*}
9999 & = 1 * 2^7 + 9 * 2^6 + 9 * 2^5 + 9 * 2^4 + 9 * 2^3 + 9 * 2^2 + 9 * 2^1 + 9 * 2^0 \\
& = 11111
\end{align*}
\]

The same number in decimal is 10000 in binary. The binary number has the value 11:

\[
\begin{align*}
11 & = 1 * 2^3 + 1 * 2^0 \\
& = 1011
\end{align*}
\]

The decimal number has the value 01:

\[
\begin{align*}
01 & = 0 * 2^3 + 1 * 2^0 \\
& = 0111
\end{align*}
\]

Note that with this, you can represent a number up to $2^n - 1$, where $n$ is the base.
Short but descriptive title

your contact information

copyright notice

identifying information for this file
Short but descriptive title

your contact information

copyright notice

identifying information for this file
Short but descriptive title

your contact information

copyright notice

identifying information for this file
Short but descriptive title

your contact information
See *The anatomy of a simple handout* for more
“I’m already overworked.”
OK, THAT HANDOUT IS FINE. NOW CAN I HAVE THE POWERPOINTS?
Forecast

- When and how to use slides
- Why and how to make handouts
- How to make your life easier
- Teaching the Facebook generation
Learn the software
Build your own templates
Use “variables”
Your favorite tip here....
Forecast

• When and how to use slides
• Why and how to make handouts
• How to make your life easier
• Teaching the Facebook generation
Technology and experience