Foundations of Computer Systems

CSCI 2330

Stephen Houser

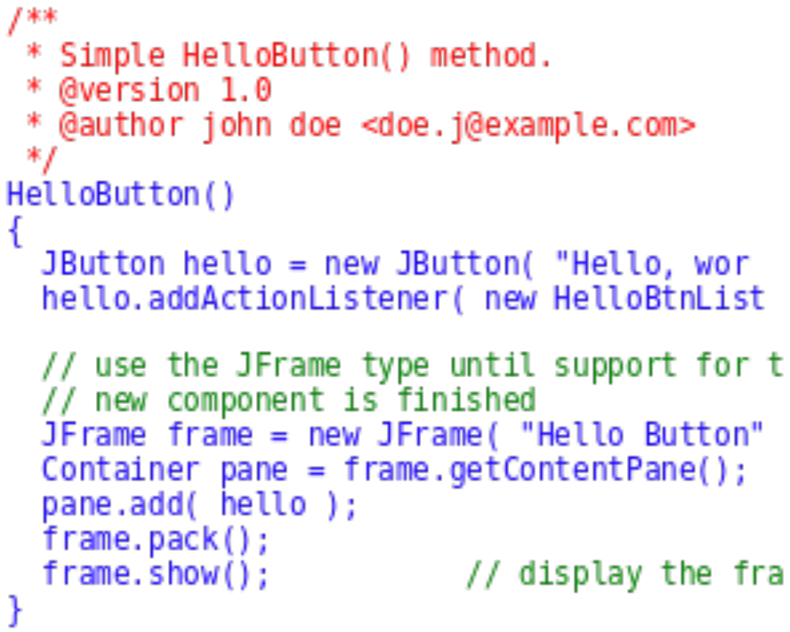
Who, What, Where…

Stephen Houser
<houser@bowdoin.edu>
112 H-L, Mon & Wed 11-12, or by appointment

—

(teaching assistants)
TBA
224 Searles

What we are about…



```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author john doe <doe.j@example.com>
 */
HelloButton()
{
  JButton hello = new JButton( "Hello, wor
  hello.addActionListener( new HelloBtnList

  // use the JFrame type until support for t
  // new component is finished
  JFrame frame = new JFrame( "Hello Button"
  Container pane = frame.getContentPane();
  pane.add( hello );
  frame.pack();
  frame.show();            // display the fra
}
```

How does a program run on a computer?

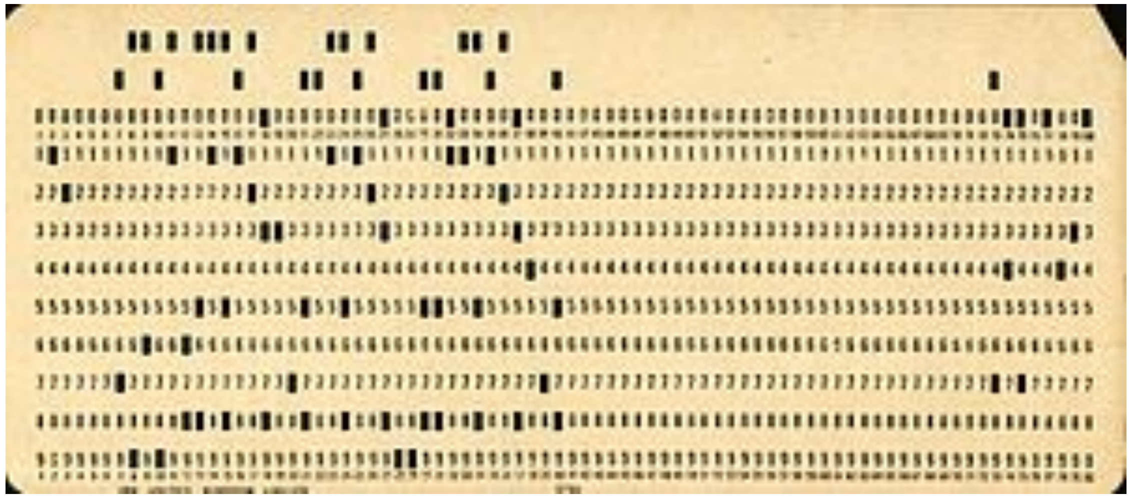
Layers of Abstraction

Layers of abstraction are used to reduce complexity and allow efficient design and implementation of complex systems.

We are going to peel away those layers

Before we start

Take a card and write down one thing you hope to learn in this course.



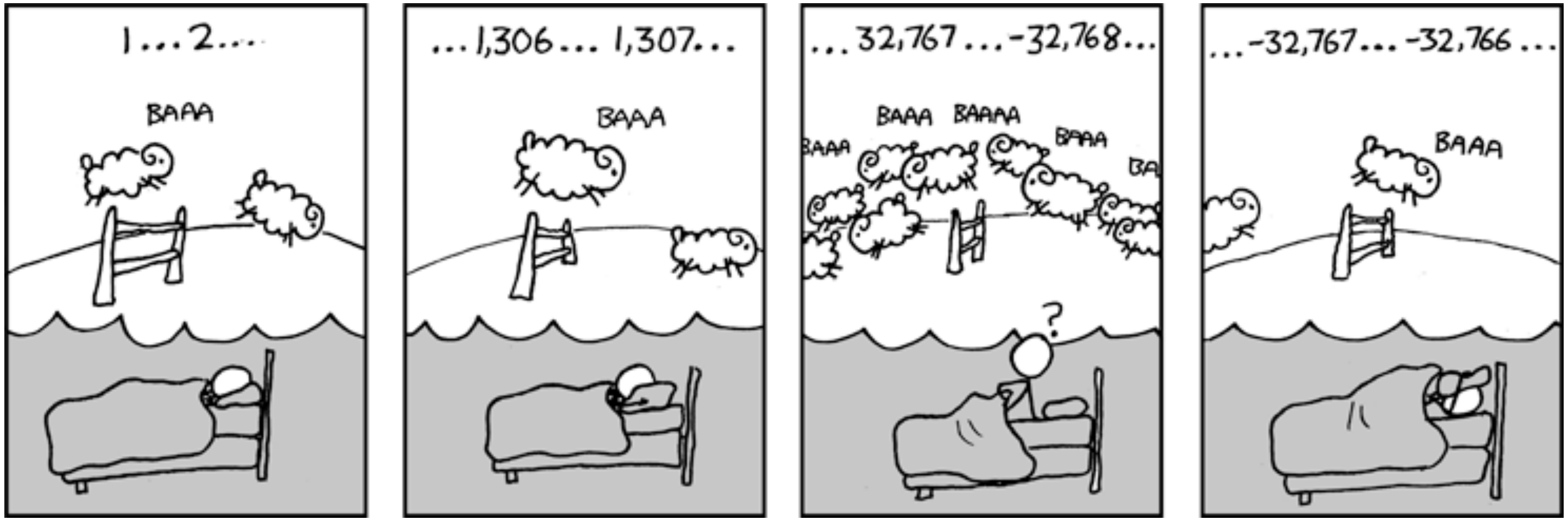
Pass your card forward when done.

Abstractions

- Abstractions have limits
- Useful Outcomes
  - Become more effective programmer
  - Able to find and fix bugs efficiently
  - Understand program performance
  - Prepare for later more detailed classes

Reality #1

Math on computers is broken*



<u>int</u>s are not Integers and <u>float</u>s are not Real

Reality #2

You need to know assembly language

Likely you will never write assembly

Understanding is key to…

- how the processor works
- how to tune performance
- how to find and fix bugs
- how to implement system software

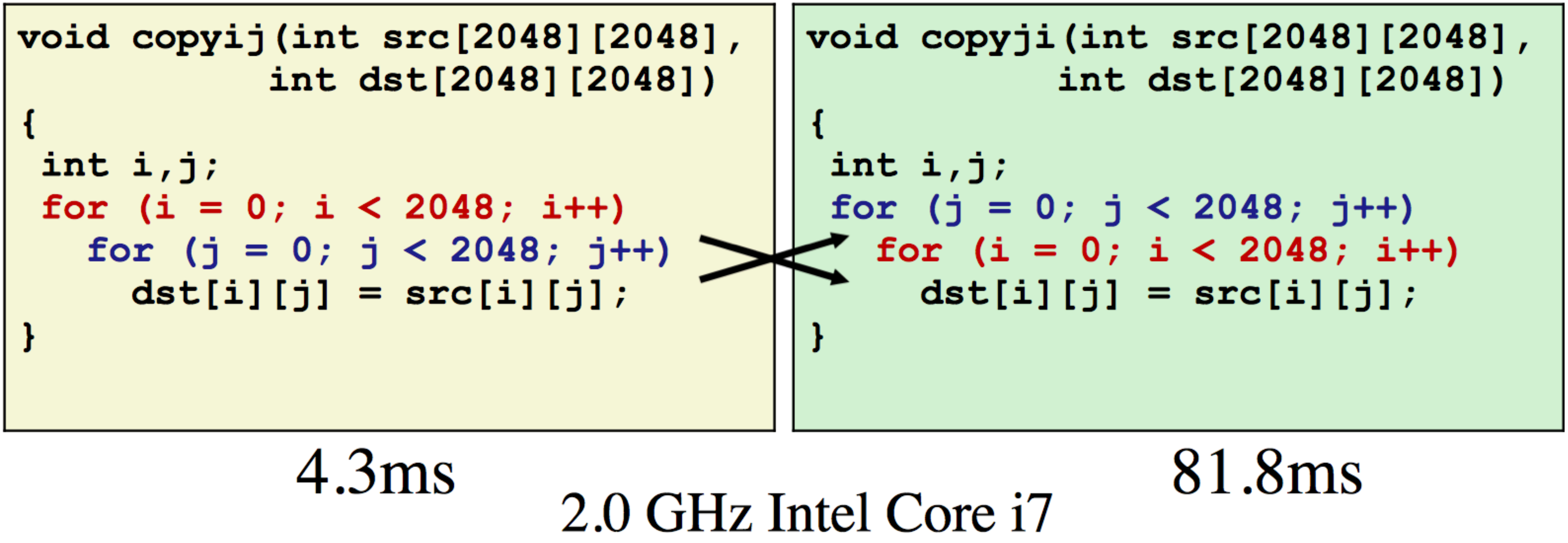
Reality #3

Memory Matters

- Memory is not unlimited
- Memory reference bugs are hard
- Memory performance is not uniform
- Caching correctly increases performance!

Reality #4

Performance is more than complexity



```
void copyij(int src[2048][2048],
            int dst[2048][2048])
{
 int i,j;
 for (i = 0; i < 2048; i++)
   for (j = 0; j < 2048; j++)
     dst[i][j] = src[i][j];
}
```

```
void copyji(int src[2048][2048],
            int dst[2048][2048])
{
 int i,j;
 for (j = 0; j < 2048; j++)
   for (i = 0; i < 2048; i++)
     dst[i][j] = src[i][j];
}
```

4.3ms

81.8ms

2.0 GHz Intel Core i7

The Schedule

From the bottom up:

1. Information Representation
2. x86_64 Processor and Machine Code
3. Memory and Cache
4. Processes and Threads

The Course

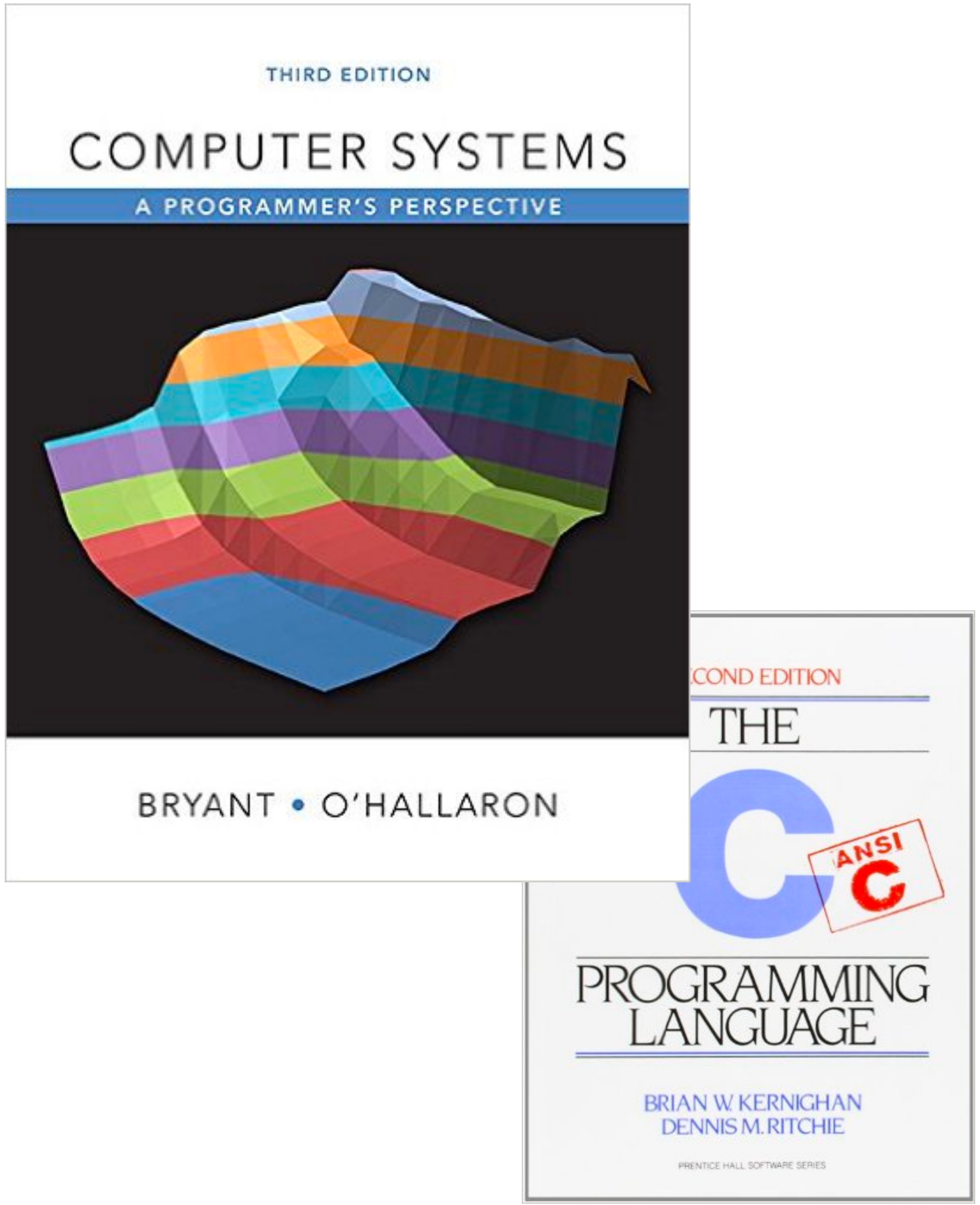
The work ahead:

- 2 Exams (midterm and final)
- 7 Labs (only 6 are graded)
- Regular in-class exercises
- Attendance and engagement
- Collaboration policy and Honor Code

The Course

Resources and such:

- Class meetings
- Lab meetings
- Laptops / Phones
- Textbook(s)



The Course

Where’s the stuff?

- Blackboard = Grades, Assignments, Links

Always start in Blackboard

- GitHub = Assignments (your work)
- Website = Schedule, Resources, etc.

Who, What, Where…

Stephen Houser
<houser@bowdoin.edu>
112 H-L, Mon & Wed 11-12, or by appointment

—

(teaching assistants)
TBA
224 Searles

fin