Course Overview and Basic C

CSCI 237: Computer Organization
1st Lecture, Friday, September 6

Kelly Shaw

Please write on the index cards

- Your preferred name
- Your preferred pronoun
- A couple of interesting things about you

Today’s Plan

- Welcome
- Course objectives
- How the course fits into the CS curriculum
- Course logistics and academic integrity
- Hello world in C

Why is Computer Organization Required for the CS Major?

Why is Computer Organization Required for the CS Major?

- Abstraction is great but
  - Correctness of software depends on it adhering to hardware realities

Correctness Problem #1: Sign

$$u_0*g^{n-1} + u_1*g^{n-2} + \ldots + u_{n-2}*g^1 + u_{n-1}*g^0$$

```
public int getHashCode(String s){
  int hash = 0;
  int n = s.length();
  for(int i = 0; i < n; i++){
        hash = g*hash + s.charAt(i);
  }
}
```

- When converting a String to a hash code (using Horner’s method), you end up with a negative number.

Correctness Problem #2: Precision

```
double sumDime = 0.0;
sumDime += 0.10;
sumDime += 0.10;
sumDime += 0.10;

System.out.println(“Sum:  $“ + sumDime);
```

Sum: $0.30000000000000004 ?

- When adding two doubles together, you get a result with a very large number of significant digits

Correctness Problem #3: Memory Misuse

```
int array[12];
int val = 0;

array[12] = 1000;
```

Either Exception or val == 1000

- When writing beyond the end of an array, you get an ArrayOutOfBoundsException OR you change a different variable’s value (depending on the programming language).

Correctness Problem #4: Out of Memory

```
int fib(int n)
{
   return fib(n-1)+fib(n-2);
}
```

- When writing a recursive function, you get a StackOverflowException or your program quits unexpectedly.

Correctness Problem #5: Name conflicts

```
public class Widget {
    public int value;
    public void setValue(int val){
        int value = val;
    }
    public static void main(String [] args){
        Widget w = new Widget();
        w.setValue(4);
        System.out.println(“Val:“ + w.value);
    }
}
```

Val: 0 ?

- When you set the wrong variable’s value due to name conflicts.

Correctness Problem #6: I/O

```
public static void main(String [] args)
{
   try{
      File file = new File(“output.txt”);
      PrintWriter pw = new PrintWriter(file);
      pw.println(“Hello World!”);
      pw.close();
   }
   catch(Exception e){
      System.out.println(e);
    }
}
```

- When you can’t save a file because the disk is full.

Why is Computer Organization Required for the CS Major?

- Abstraction is great but
  - Correctness of software depends on it adhering to hardware realities
  - Performance depends on software exploiting what the hardware is fast at and avoiding what its slow at

Performance Problem #1

```
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];
}
```

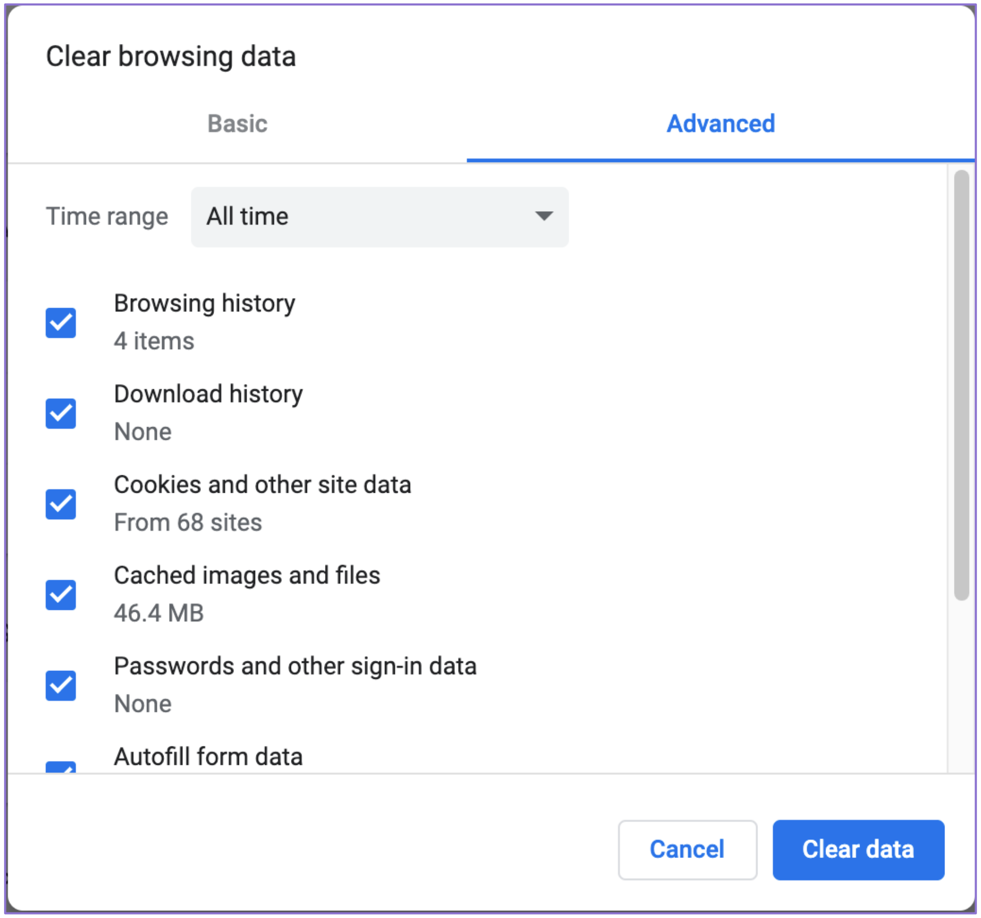
48ms

2.9 GHz Intel Core i5

167ms

- Hierarchical memory organization
- Performance depends on access patterns
  - Including how to step through a multi-dimensional array (row or column?)

Performance Problem #2



Clear browsing data

Time range: All time

☑ Browsing history
4 items

☑ Download history
None

☑ Cookies and other site data
From 68 sites

☑ Cached images and files
46.4 MB

☑ Passwords and other sign-in data
None

☑ Autofill form data

- Without web caches, web browsers would require all data displayed to be transmitted on access of every webpage

Why is Computer Organization Required for the CS Major?

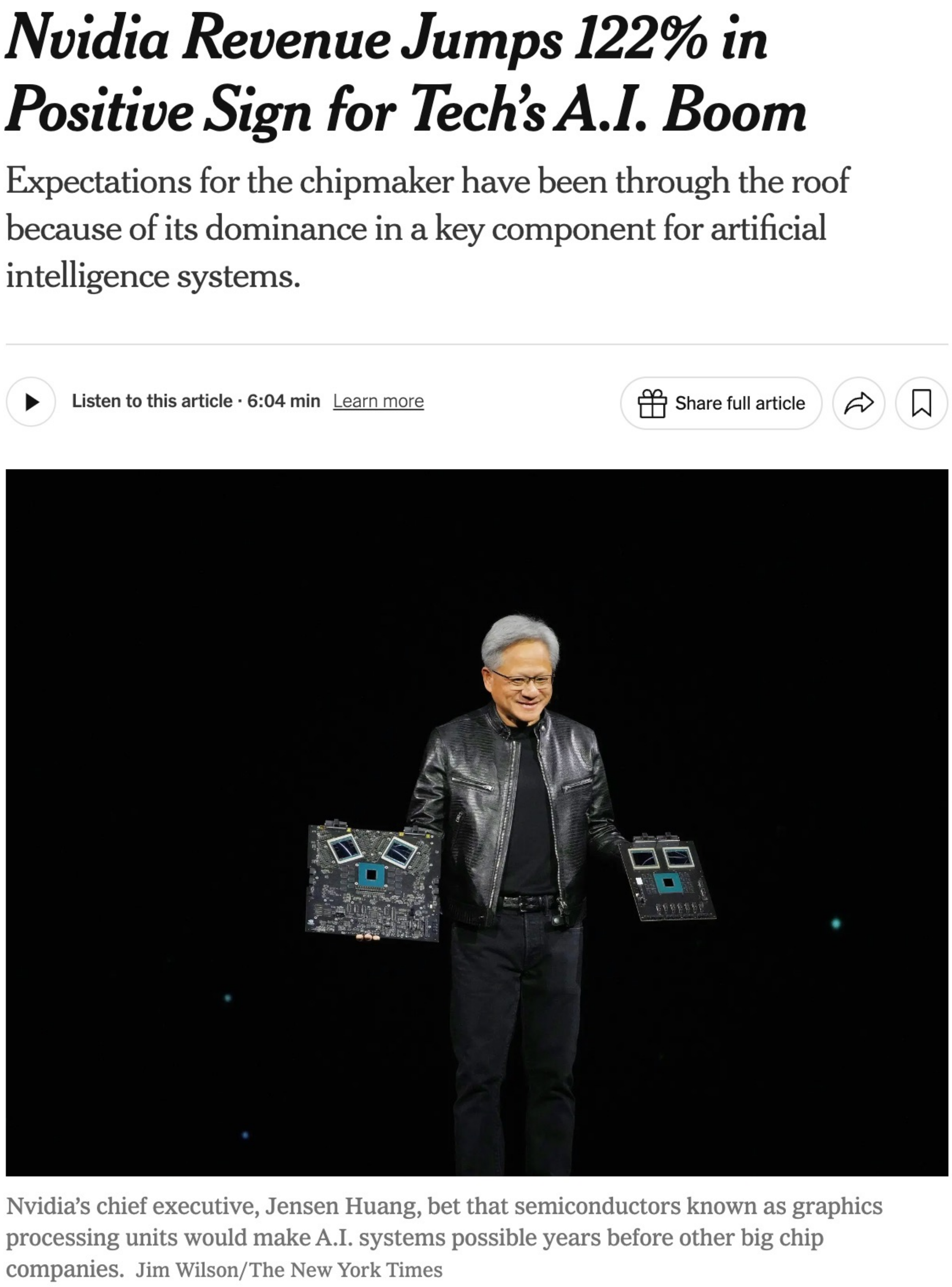
- Abstraction is great but
  - Correctness of software depends on it adhering to hardware realities
  - Performance depends on software exploiting what the hardware is fast at and avoiding what hardware is slow at
  - Hardware constructs may introduce security flaws

Security Problem

- Hardware features introduced for performance become security flaws in environments with multiple users (esp. data centers).

Why is Computer Organization Required for the CS Major?

- Abstraction is great but
  - Correctness of software depends on it adhering to hardware realities
  - Performance depends on software exploiting what the hardware is fast at and avoiding what hardware is slow at
  - Hardware constructs may introduce security flaws
  - Hardware enables AI/ML/NLP/Graphics/etc.



Nvidia Revenue Jumps 122% in Positive Sign for Tech's A.I. Boom

Expectations for the chipmaker have been through the roof because of its dominance in a key component for artificial intelligence systems.

Nvidia's chief executive, Jensen Huang, bet that semiconductors known as graphics processing units would make A.I. systems possible years before other big chip companies. Jim Wilson/The New York Times

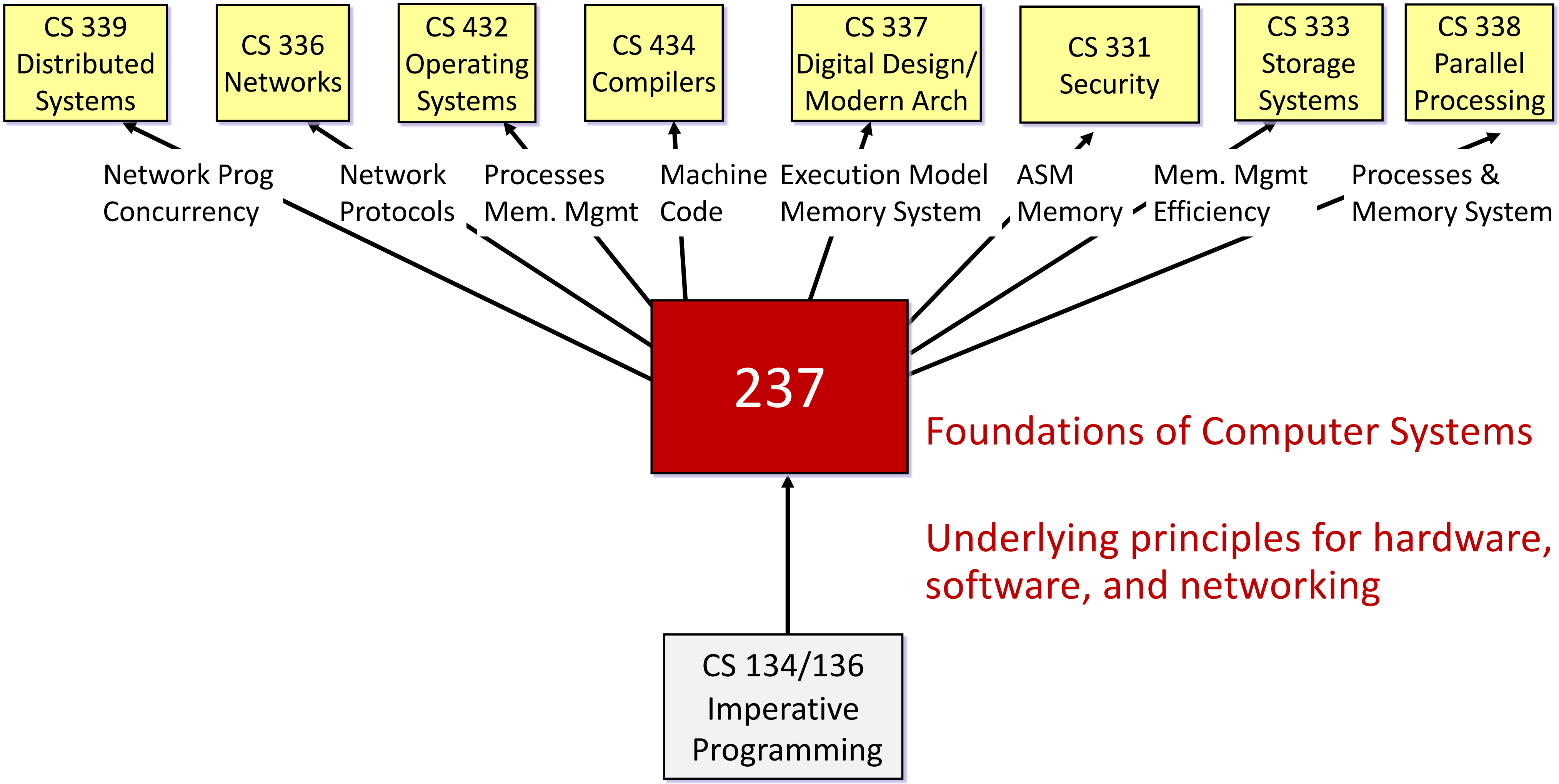
Course Objectives

- Conceptual
  - Learn hardware fundamentals
  - Learn how hardware impacts correctness and performance of software

More effective and efficient programmer

- Practical
  - Learn C
  - Learn software development skills to become a more effective and efficient programmer

Role within CS Curriculum



Foundations of Computer Systems

Underlying principles for hardware, software, and networking

Today’s Plan

- Welcome
- Course objectives
- How the course fits into the CS curriculum
- Course logistics and academic integrity
- Hello world in C

I’m Kelly

kshaw@cs.williams.edu
TCL 309
Office hours/drop by times:

- M 1-5:30
- Th 2-3:30
- By appt.

CS 237 TA Team

- Brij Howard-Sarin
- Nick Canora
- Tanatswa Manyakara
- Valeria Starkova
- David Wang

General Categories of Activities

- Lectures
- Readings
  - Do in advance of lecture
- Practice problems
  - Not graded
  - Solutions will be posted on Glow
- Weekly quizzes
  - Available Wednesday at 2:30pm through Friday at 2:30 pm
  - 30 minutes
  - Open book, open notes, but can only talk to Jeannie & Kelly about
- Lab assignments
  - 6-7 throughout semester (will post anticipated due dates soon)
- Midterm (Wednesday, October 23) and final exam
  - Scheduled, closed book, closed notes

Cheating: Description

- **Please pay close attention! We take this VERY seriously.**

- What is cheating?
  - Sharing code: by copying, retyping, **looking at**, or supplying a file
  - Describing: verbal description of code from one person to another
  - Coaching: helping your friend to write a lab, line by line
  - **Searching the Web for solutions or using LLMs (e.g., chatGPT)**
  - Copying code from a previous course or online solution
    - You are only allowed to use code we supply, or from the CS:APP website
- What is NOT cheating?
  - Explaining how to use systems or tools, Unix environment
  - Searching the web for compiler errors or tool/language documentation
  - Discussing the documentation or asking clarifying questions about course materials
- See the course syllabus/assignment description for more details.
  - Ignorance is not an excuse. When in doubt, ask!

Cheating: Consequences

- Penalty for cheating:
  - According to our Honor Code, if we suspect cheating, we must notify the honor committee
  - If found guilty, probably will fail course (less serious outcomes possible, but less common)
  - Permanent mark on your record

- Detection of cheating:
  - We have sophisticated tools for detecting code plagiarism
  - We (unfortunately) catch students every semester

- Just don’t do it!
  - Start early
  - Ask for help when you get stuck

Course Tools

- Class Web page: **http://www.cs.williams.edu/~cs237/**
  - Complete schedule of lectures, exams, and assignments
  - Copies of lecture slides, labs, practice problems
  - Calendar with office hours!!!
- GLOW
  - Practice midterm questions
  - Practice problem solutions
  - Weekly quizzes
- Slack
  - Forum for asking/answering questions
  - Forum for you to work collectively on practice problems
- Zoom
  - TA office hours in person in TBL 301

Facilities

- Kelly’s lectures in Clark 105
- Labs and TA help hours in TBL 301
- Kelly’s help hours in TCL 309
- Use SSH to work remotely
  - Many command line tools, so emacs/vim/nano are great editors
  - Git lets us move around among machines
    - commit and push *often*
  - https://www.cs.williams.edu/systems/cscownames.html

  See webpage and lab1 for info about git, ssh, Unix machines

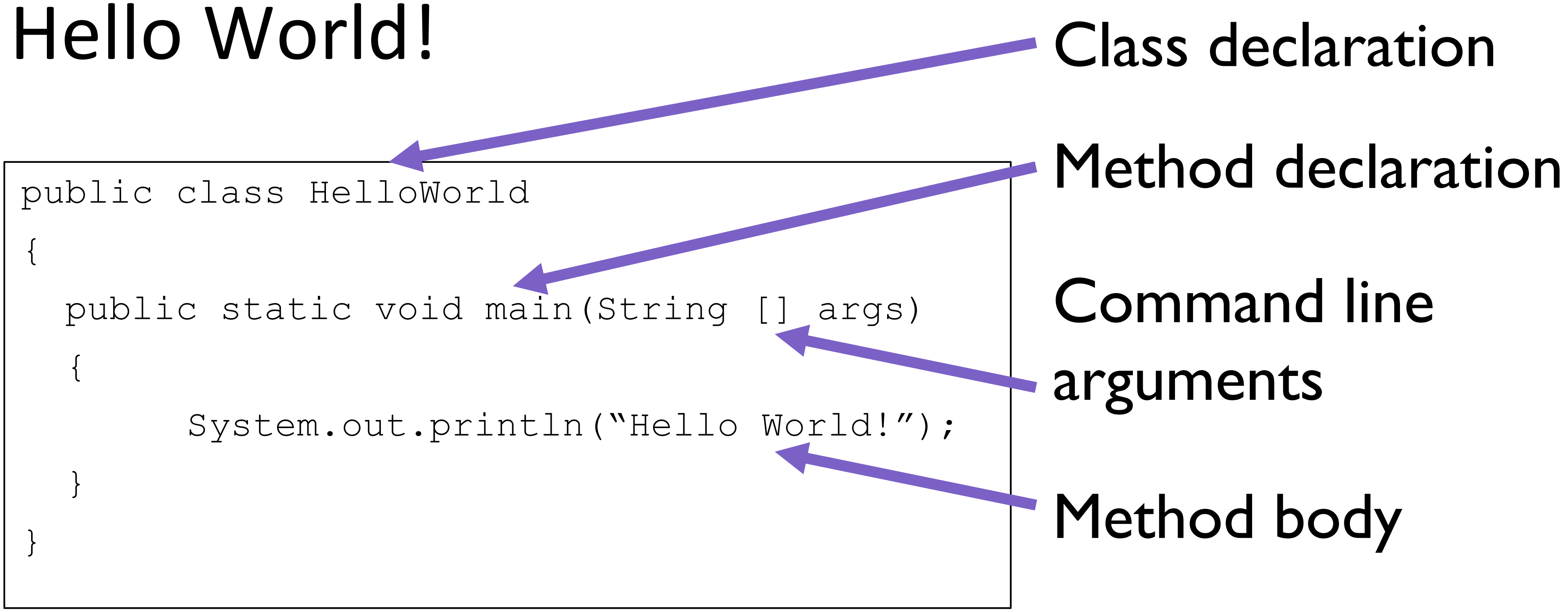
Course To-Do Items

- Please read
  - syllabus, coding guidelines, K&R Ch. 1 (as a reference), CSAPP Ch 1.1-1.4
- Sign up and introduce yourself on Slack workspace
  - #introductions channel
- Read K&R Ch. 2-3
- Weekly quiz
  - Available on Glow from Wednesday at 2:30 to Friday 2:30pm
  - 30 minute window for you to complete it
  - Open book, open notes, but can't discuss with anyone but Kelly
- Lab 1 Checkpoint (parts 0 and 1) due Wednesday at 11pm
  - Submit by commit/pushing your work to evolene
  - Do NOT turn off the Linux machines (only log out)
- Colloquium today in Wege at 2:35pm

Today’s Plan

- Welcome
- Course objectives
- How the course fits into the CS curriculum
- Course logistics and academic integrity
- Hello world in C

Hello World!



```
public class HelloWorld
{
  public static void main(String [] args)
  {
        System.out.println(“Hello World!”);
  }
}
```

Class declaration

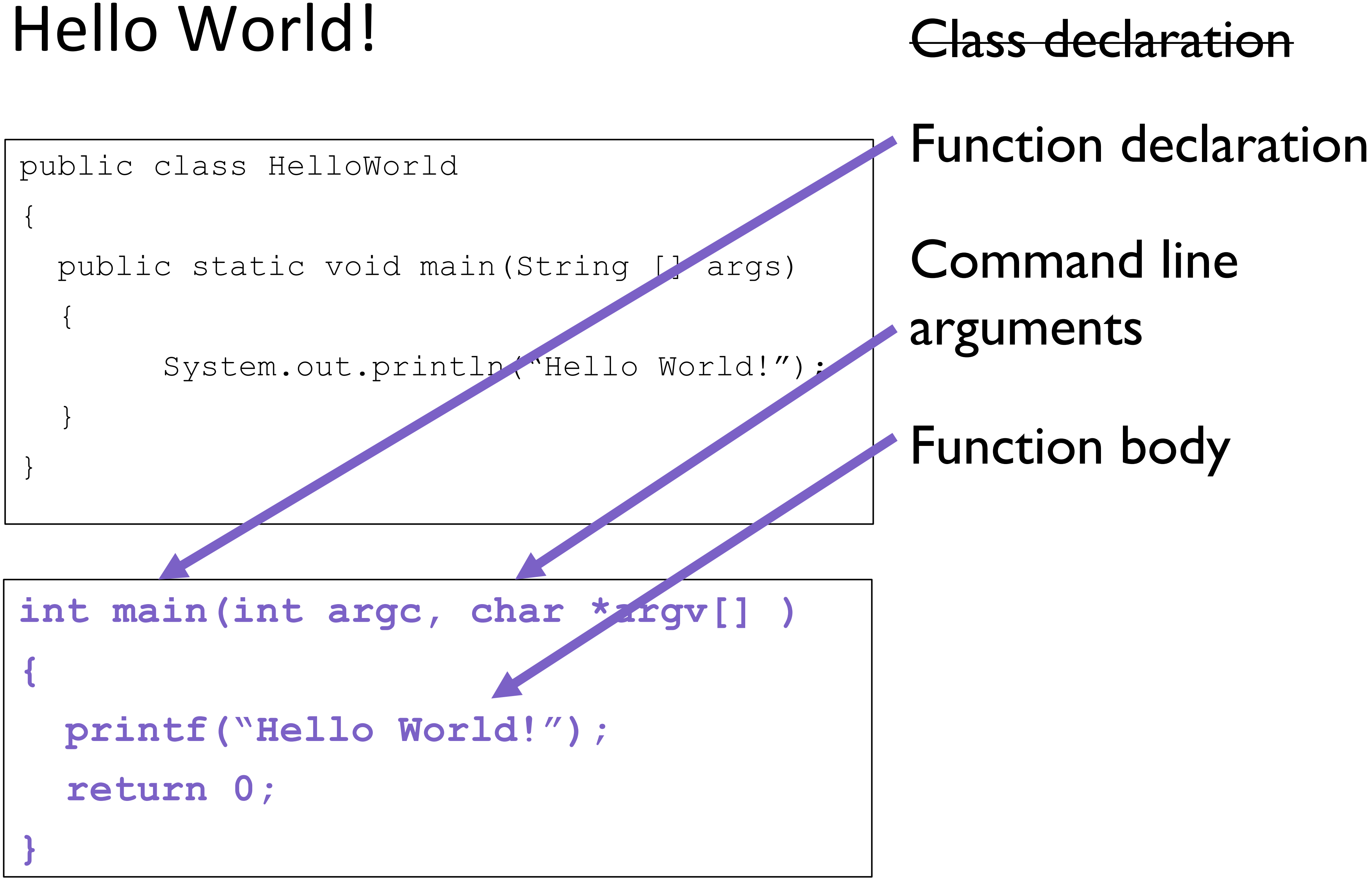
Method declaration

Command line arguments

Method body

```
int main(int argc, char *argv[] )
{
  printf(“Hello World!”);
  return 0;
}
```

Hello World!



~~Class declaration~~

Function declaration

Command line arguments

Function body

```
public class HelloWorld
{
  public static void main(String [] args)
  {
        System.out.println("Hello World!");
  }
}
```

```
int main(int argc, char *argv[] )
{
  printf("Hello World!");
  return 0;
}
```

C Functions

- Global functions
  - ex. *main*
- Return type specified before function name
- Parameters specified w/ type in parentheses

```
#include <stdio.h>

int foo(int x) //definition
{
 return 2*x;
}

int main(int argc, char *argv[])
{
 printf(“Foo %d\n“, foo()); //call
 return 0;
}
```

`printf`

```
printf(<formatted string>, …)
```

- Prints to `stdout`
- Specify formatted string as 1st argument. Use `%` to indicate type for each value to be inserted.
- Remaining arguments are items to be placed in string
  - `printf(“hello world!”);`
  - `printf(“int num %d float num %f %s\n”, 3, 3.14, “done”);`
- Part of the `stdio.h` library
  - Insert `#include <stdio.h>` at top of C file