

Week00

- introduce class
- introduce self
- take roll
- syllabus
- type
- whitespace
- scope
- success tips
- fix Eclipse installs

NOTE: lecture is recorded and posted ~noon same day

NOTE: Homework00 is due on Monday

NOTE: Help Hours start on Monday



introduce class

what is CS 136?

- a computer science major requirement 🎓
- a bridge between coding and theory
- about how you store data (**HINT:** in an array)
- harder problems
- bigger problems
- good prep for coding interviews 💎
- going beyond "it works" to different measures of "good"
 - speed 🏎️
 - space 🗑️
 - elegance? ✨

introduce self

[other slide deck]

take roll

take roll

- this is the only time i will ever take roll
 - this is also the only time i will wear a coat 🙄
- please correct me if i mispronounce your name
- if you are not yet enrolled, please fill out the other sheet now

syllabus

syllabus

sylla 🚌



the syllabus is online

<https://github.com/james-bern/CS136/wiki>

grading

final grade

- **final grade** is 1/3 homework, 1/3 midterm exam, 1/3 final exam
- **homework** is on an A- (90), A (95), A+ (100) scale
 - A- is meant to be doable
 - A is meant to be challenging
 - A+ is meant to be a *growth opportunity* 🌱
 - i will go over some of your homework with you during lab and ask you questions about it 😊👍
- **exams** are in-person, on-paper, no-notes
 - questions are either similar to homework, or similar to PracticeExam
 - "if you understand the homework, the exam should be unsurprising and approachable"

ChatGPT

ChatGPT can do every homework
in this class up to an A level



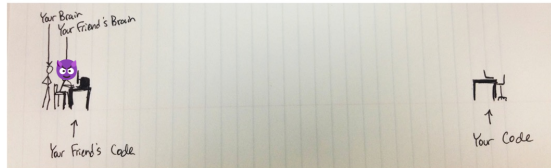
it is impossible for me to prove
whether or not you used ChatGPT
(people who say otherwise are wronggg)

so, ChatGPT, etc. are "allowed," but...

- you are not allowed copy and paste code from anywhere
 - bad bad very bad don't do it
 - (unless i gave you the code then it's fine 😊👍)
- you can't use ChatGPT, etc. on exams
- 🦉 "if your coding job can be done by an AI, then it will be done by an AI"

collaboration

When you help another student with their code, your own code should remain 50 feet away. Help them with your brain, not your own solution.



questions?

Java

Java



this class is in Java

this class is in Cow.Java

which is basically C

by which I mean Java

type

type

a variable in Java has a type

```
# okay in Python
foo = 7 # foo is an int
foo = False # now, foo is a boolean
```

```
// NOT okay in Java
int foo = 7; // foo is an int
foo = false; // Error: incompatible types:
// boolean cannot be converted to int
```

declaring and initializing are separate things

- // Option A: one line
`int foo = 7; // declare int foo and initialize it to 7`
- // Option B: two lines
`int foo; // declare a variable foo of type int`
`foo = 7; // initialize foo to 7`

primitive types

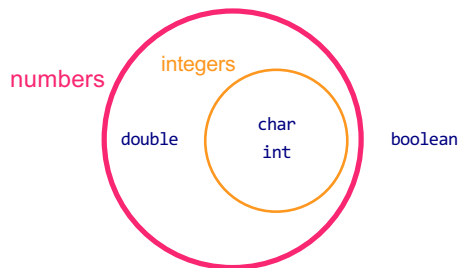
primitive types

- in this class, "a variable being a **primitive**" means that the variable is a `boolean`, `char`, `double`, or `int`
- primitive types are simple
- primitive types are small
- primitive types are NOT Objects
 - we will talk about Objects later
 - **examples of Objects:** `String`, `MyCoolClass`, `int[]` (array of ints)

`boolean`, `char`, `double`, `int`

- a `boolean` stores a truth value
 - `true`, `false`
- a `char` stores a character
 - `'\0'`, `'a'`, `'z'`, `'!'`, `'\n'`
- a `double` stores a real number
 - `0.0`, `-0.5`, `3.1415926`, `Double.NEGATIVE_INFINITY`
- an `int` stores an integer
 - `0`, `-1`, `4`

primitive type Venn diagram



`char` is an integer type

- a `char` is an integer type
 - each `char` has a corresponding integer, for example (`'a' == 97`)
 - the letters are in order (`'a' == 97`), (`'b' == 98`), (`'c' == 99`)...
 - the numbers are also in order (`'0' == 48`), (`'1' == 49`)...
- you can do math with `char`'s
 - `char foo = 'a' + 2; // foo is 'c'`
 - `char bar = '0' + 7; // bar is '7'`
 - `int baz = '6' - '0'; // baz is 6`

zero

- each primitive data type has its own notion of what it means to "be zero"
- `int` `zero = 0;`
- `double` `zero = 0.0;`
- `boolean` `zero = false;`
- `char` `zero = '\0';` // the "null character"

whitespace

whitespace

whitespace

- **whitespace** includes spaces, tabs, and newlines
- 🐍 Python does care about whitespace (*indentation changes what code does*)
- ☕ Java does NOT care about whitespace (it cares about curly braces)
- 🧑 do you care about whitespace?
 - some guidelines:
 - be consistent
 - make sure your curly braces line up
 - ⚡ use Eclipse's auto-indentation feature (see Tut00)! no excuses!

| sparks joy | NOT equivalent -- doesn't spark joy |
|--|--|
| <pre>for (int i = 0; i < 10; ++i) { if (i % 3 == 0) { System.out.println("fizz"); } }</pre> | <pre>for (int i = 0; i < 10; ++i) { if (i % 3 == 0) { System.out.println("fizz"); } }</pre> |

scope

scope

scope

- a **scope** is a region of code in which variables live
- in Java, a scope is (usually) defined by a pair of curly braces
 - OUTER_SCOPE { INNER_SCOPE } OUTER_SCOPE

```
{
  int i;
  {
    int j;
    // you CAN use i here 🤖
    // you CAN use j here 🤖
  }
  // you CAN use i here 🤖
  // you CANNOT use j here 🙅
}
```

common scope-related errors

cannot be resolved to a variable

Compile Error: foo cannot be resolved to a variable

```
class Main extends Cow {
  public static void main(String[] arguments) {
    if (...) {
      int foo = 0;
    } else {
      int foo = 1;
    }
    PRINT(foo);
  }
}
```

cannot be resolved to a variable

```
class Main extends Cow {
  public static void main(String[] arguments) {
    int foo;
    if (...) {
      foo = 0;
    } else {
      foo = 1;
    }
    PRINT(foo);
  }
}
```

duplicate local variable

Compile Error: duplicate local variable foo

```
class Main extends Cow {
  public static void main(String[] arguments) {
    int foo;
    if (...) {
      int foo = 0;
    } else {
      foo = 1;
    }
    PRINT(foo);
  }
}
```

duplicate local variable

```
class Main extends Cow {
  public static void main(String[] arguments) {
    int foo;
    if (...) {
      foo = 0;
    } else {
      foo = 1;
    }
    PRINT(foo);
  }
}
```


questions?

success tips

success tips (1/3)

- establish basic study habits
(this might not have been necessary in highschool; it most likely is now)
 - start homework early! (very hard to code well while stressed)
- don't work in a vacuum
 - collaborate with your friends / acquaintances / enemies
 - come to help hours (they are there for you!)
- code a lot (ideally, 5+ days per week)
 - if the homeworks are feeling too hard, talk to me; we have **resources**!
 - if the homeworks are too easy, do the A+ / A++; do side projects; just keep coding; code code code code code; COOOOOOOODE

success tips (2/3)

- try to find your own bugs
 - this is a huge skill; it takes time (and pain) to build
- try to answer your own questions (using Google!)
("what does IDE stand for?"; "what does this error message mean?")
 - this is also a huge skill; it also takes time (and pain) to build
- however, if you are stuck (≥ 10 minutes of pain), ASK FOR HELP
 - there is literally no judgement here; we all have different starting points; my only goal is for all students to get stronger and have fun 🤖💡
 - and, for what it's worth, i was a mechanical engineering undergrad and my first comp sci course was "terrifying" (though, eventually, fun)

most importantly though





come see me if you
had trouble with Tut00

otherwise, see you in
class on Monday 🌞