伯克利CS61B

What is 61B about?

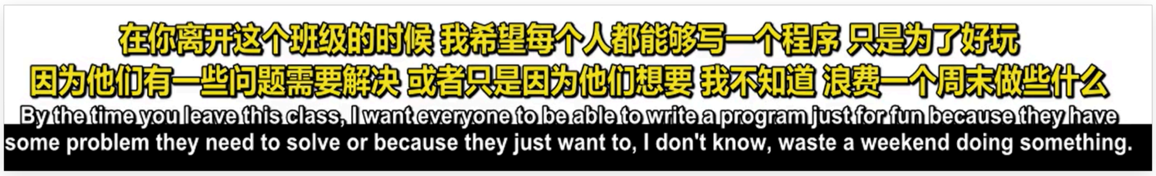
- writing code that runs effciently
  - good algorithms
  - good data structures
- writing code effciently
  - designing,building,tresting,and debugging large programs.
  - Use of programming tools.
    - git,intelli,JUnit,and various command line tools.
  - Java(not the focus of the course)

Assumes solid foundation in programming fundamentals,including:

- **Object oriented programming,recursion,lists,and trees.**

Other great features of 61B:

- The most popular topics for job inte
  rview questions in software engineering
  - Examples: Hash tables, binary
    search trees, quick sort, graphs, Dijkstra's algorithm.
- Some really cool math. Examples:
  - Asymptotic analysis.
  - Resizing arrays
  - The isometry between self-balancing 2-3 trees and self-balancing red black trees
  - Graph theory.
  - P=NP
- Once you're done: **the confident sen
  se that you can build any software.**



在你离开这个班级的时候 我希望每个人都能够写一个程序 只是为了好玩
因为他们有一些问题需要解决 或者只是因为他们想要 我不知道 浪费一个周末做些什么
By the time you leave this class, I want everyone to be able to write a program just for fun because they have some problem they need to solve or because they just want to, I don't know, waste a weekend doing something.

Question for You

What do you hope / expect to learn from this class? Why are you taking it?

- Job.$$$$$$$$
- I want to be able to run my code efficiently (finally)

- I want an A.
- Coding from scratch.
- Greater Grasp of data structures and algorithms

Who are you?

- Freshman? Sophomore? Junior? Senior? Grad student? None of the above?
- CS Major? Intending to be a CS Major? Something else?
- CS 61A? Java experience

Course Components

Lectures provide you with an introduction and a foundation.

You'll learn most of what you learn in the class by:

- Programming(labs, hws, projects, discussion section).
- Solving interesting problems (study guides, HW3, HW4 old exam problems, discussionsection).

> Generally speaking,you'll learn a lot through doing. **Programming is not something you can learn just from theory.** You have to actually practice it. So you'll be getting a lot of experience with programming Java through your homeworks,your labs, your projects. **Discussion won't do direct programming, but you'll be discussing a lot about how to code effectively.** You'll get to solve a lot of interesting problems throughout the semester.

Evaluation

Four types of points in this class:

- Low effort, everyone should get them: Weekly Surveys, Course Evaluations
  - Median score is 100%
- High effort, everyone should get them: HW, Project, Lab
  - Median score is 100%
- High effort, not everyone gets them: Exams
  - Mean score is 65%
  - Final exam score can replace midterms if you have a bad midterm (or two)
- Pacing points: Attending Discussion, Lab, and keeping up with Lecture
  - Small amount of extra credit for keeping up with class.
  - Will not increase your score beyond 75% (B-).
    - Example: You have 740 points and earn 20 pacing points, you get 750 points.
- B to B+ threshold is 65% on exams, 95% on everything else.

Full details around point distributions, letter grade assignments, grade

Class Phase

This class is divided into three phases:

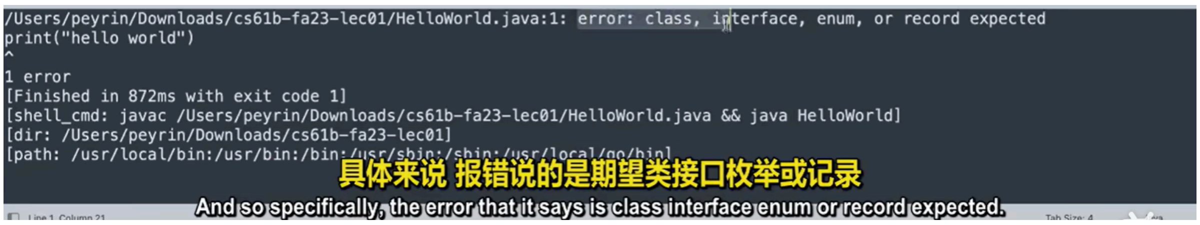
- Phase 1(weeks 1 -4): Intro to Java and Data Structures.
    - All coding work is solo.
    - Moves VERY fast.
    - HW0 (intro to Java) due Friday (in two days!)
- Phase 2(weeks 5-10): Data Structures:
    - All coding work is solo.
    - Moves moderately fast.
- Phase 3 (weeks 12-14): Algorithms.
    - Coding work is entirely dedicated to final project, done in pairs.0
    - Slower pace.

Intro to Java

```
print("hello world");
```

所以我要尝试的第一件事是将这个程序复制粘贴到Java解释器中

So the first thing I'll try is I'll take this program and copy-paste it into the Java interpreter.



```
/Users/peyrin/Downloads/cs61b-fa23-lec01/HelloWorld.java:1: error: class, interface, enum, or record expected
print("hello world")
^
1 error
[Finished in 872ms with exit code 1]
[shell_cmd: javac /Users/peyrin/Downloads/cs61b-fa23-lec01/HelloWorld.java && java HelloWorld]
[dir: /Users/peyrin/Downloads/cs61b-fa23-lec01]
[path: /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/go/bin]
```

具体来说 报错说的是期望类接口枚举或记录

And so specifically, the error that it says is class interface enum or record expected.

Java is a language that cares very much about object-oriented programming.

It really cares that all the code you write is in a class. let's make a class. So the *magic words* I'm going to use to define a class are public class.

```
public class HelloWorld{
    public static void main(String[] args){
        System.out.println("hello world");//print not include "\n"
    }
}
```

Java and Object Orientation

Reflections on Hello World:

- In Java, all code must be part of a class.
- Classes are defined with public class CLASSNAME
- We use { } to delineate the beginning and ending of things.
- We must end lines with a semicolon.
- The code we want to run must be inside public static void main(String[] args)
    - We'll learn what this means later.

Java is an object oriented language with strict requirements:

- Every Java file must contain `a class declaration*`.
- **All code** lives inside a `class*`, even helper functions, global constants, etc.
- To run a Java program, you typically define a main method usingpublic static void main(Stringl args)

*: This is not completely true, e.g. we can also declare "interfaces" in .java files that maycontain code. We'll cover these soon.

Lager:

```
x=0
while(x<10):
    print(x)
    x=x+1
```

Java:

```
public class HelloNumbers{
    public static void main(String[] args){
    int x=0;
        while(x<10){
            System.out.println(x);
            x=x+1;
        }
    }
}
```

> So the problem here is that in Java,before you can use a variable, you have to declare that it exists first.

So what java is expecting is first you have to say, I declare there is a variable called x, and it is of type integer.

And this seems like a really minor change. But as you start using Java for more complicated programs, it turns out that just this little tiny change can have huge implications on how you write your program.

Java and Static Typing

Reflections on Hello Numbers.

- Before Java variables can be used, they must be declared.
- Java variables must have a specific type.
- Java variable types can never change.
- Types are verified before the code even runs!

Java is statically typed!

- All variables, parameters, and methods must have a declared type.
- That type can never change.
- Expressions also have a type,e.g. “larger(5, 1)+ 3" has type int.
- The compiler checks that all the types in your program are compatible before theprogram ever runs!
    - e.g.String x=larger(5,10)+ 3 will fail to compile.
    - This is unlike a language like Python, where type checks are performed DURING execution.

如果我们把x="horse"呢?

```
x=0
while(x<10):
    print(x)
    x=x+1
x="horse"
print(x)
```

python是可以运行的，但是Java却不能，他的执行情况是：*我们可以看到0 1 2 3 4 5 6 7 8 9的情况，但我们不能得到horse的字样*

```
public class HelloNumbers{
    public static void main(String[] args){
    int x=0;
        while(x<10){
            System.out.println(x);
            x=x+1;
        }
        x="horse";
        System.out.println(x);
    }
}
```

如果此时我们添加`x="horse"+5`的字样，python只能出现

```
0 1 2 3 4 5 6 7 8 9
can only concatenate str(not 'int') to str
```

Java则是：

```
You can't add horse and 5.
```

but this time, the code didn't even run.

这是由于编译器的原因，java的编译器分析完后告诉说，我拒绝运行这段代码，因为这段代码有一个明显的错误。

> You have a tight bear.

Java和Python都是高级编程语言，它们都需要编译过程，但它们的编译方式和运行机制有所不同。

1. **Java**：
   - Java代码首先被编译成字节码（`.class`文件），这是一种中间状态的代码，它不是直接运行在硬件上的机器码，而是运行在Java虚拟机（JVM）上的。
   - JVM会将字节码进一步"即时编译"（JIT编译）成特定平台的机器码，这个过程是在程序运行时进行的。
   - 因此，Java实际上是"编译一次，运行时编译多次"。
2. **Python**：
   - Python代码在运行时会被解释器逐行编译成字节码，然后由Python虚拟机（PVM）执行。
   - Python的字节码不是跨平台的，它是为特定的Python解释器版本和操作系统平台编译的。
   - 因此，Python可以被看作是"一边编译一边运行"，但它也有一个中间的字节码阶段。

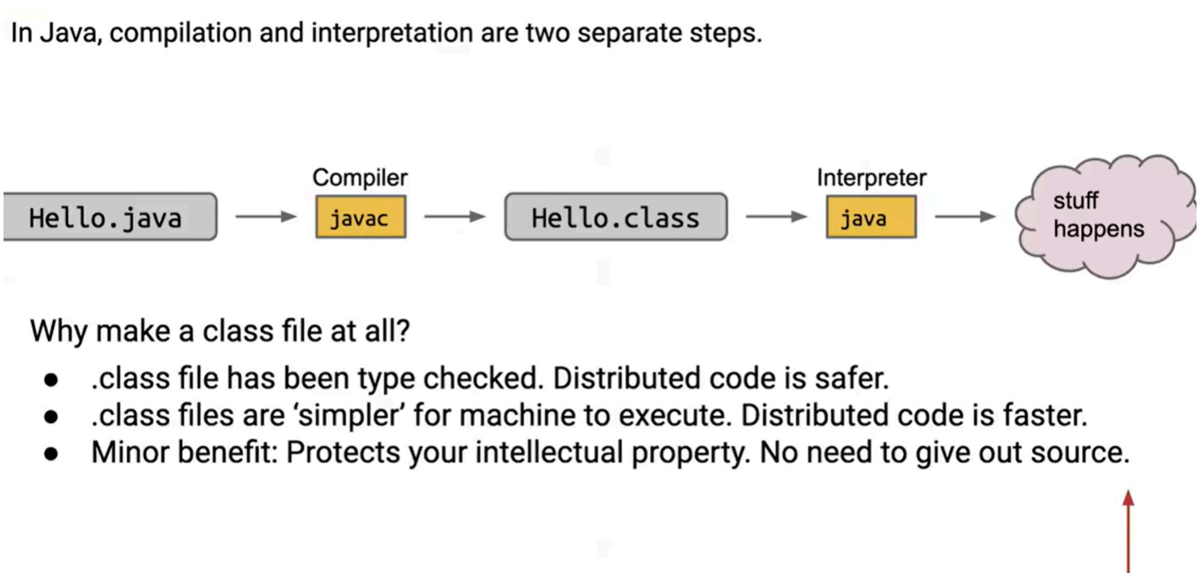
Writing a function.

```python
def larger(x,y):
    if(x>y):
        return x
    else:
        return y

print(large(-5,10))
```

```java
public calss LargeDemo{
    public static int larger(int x,int y){
        if(x>y){
            return x;}
        else{
            return y;
        }
    }
    public static void main(String[] args){
        System.out.println(larger(5,4))
    }
}
```

Larger: Reflections

- Functions must be declared as part of a class in Java.A function that is part of a class is called a "method"So in Java, all functions are methods.
- To define a function in Java, we use "public static"We will see alternate ways of defining functions later.
- All parameters of a function must have a declared type,and the return value of the function must have a declared type.Functions in Java return only one value!



In Java, compilation and interpretation are two separate steps.

Why make a class file at all?

- .class file has been type checked. Distributed code is safer.
- .class files are 'simpler' for machine to execute. Distributed code is faster.
- Minor benefit: Protects your intellectual property. No need to give out source.

The Good:

- Catches certain types of errors, making it easier on the programmer to debug their code.
- Type errors can (almost) never occur on end user's computer.
- Makes it easier to read and reason about code.
- Code can run more efficiently, e.g. no need to do expensive runtime type checks.

The Bad:

- Code is more verbose.
- Code is less general, e.g. would need a second larger function to comparenon-integers like 5.5.

如果我们搞砸了 我们会一起搞砸的
And if we screw up, we'll screw it up together.

Review:Object-Oriented Programming

- A model for organizing programs
  - Modularity: Define each piece without worrying about other pieces, and they all work together;

    也就是说，我们可以与对象进行交互而不必了解其内部实现
  - Allows for data abstraction:You can interact with an object without knowing how it's implemented ;
- Objects
  - An object bundles together information and related behavior
  - Each object has its own local state
  - Several objects may all be instances of a common type
- Classes
  - A class serves as a template for all of its instances
  - **Each object is an instance of some class**

Class In Python ,Java and Cpp

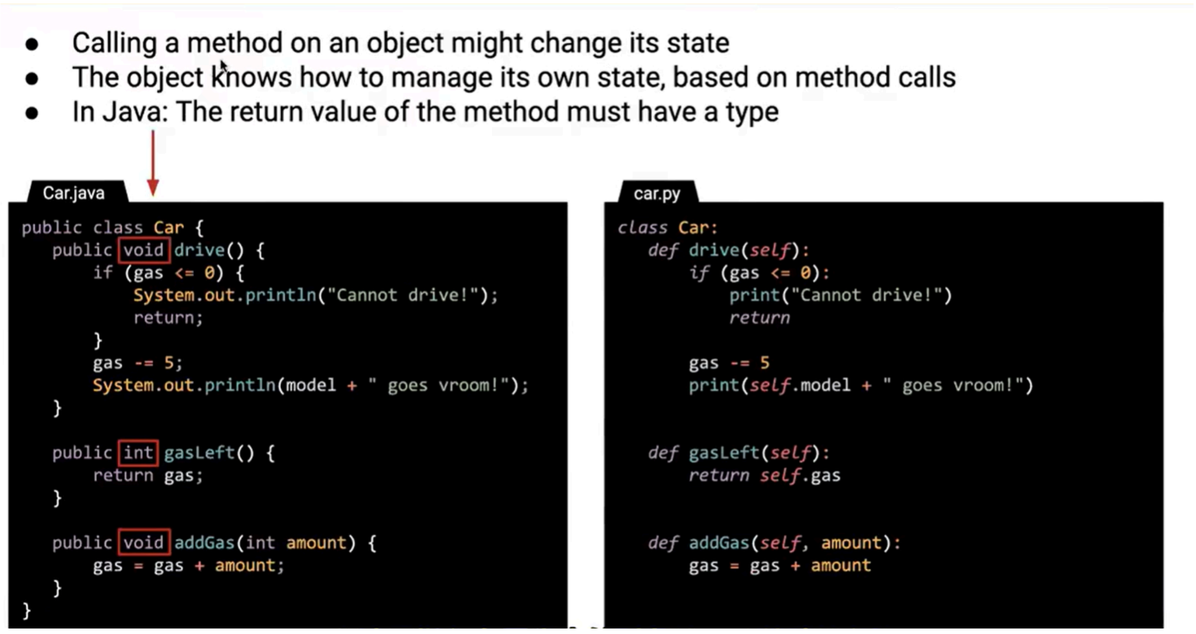
```
class Car:
    def __inti__(self,m):
        self.model=m
        self.wheels=4

    def drive(self):
        if(self.wheels<4):
            print(self.model+" no go vroom")
            return
        print(self.mode+" goes vroom")

        def getNumWheels(self):
            return self.wheels

        def driveIntoDitch(self,wheelsLost):
            self.wheels=self.wheels-whelelsLost

            c1=Car("Civic TyPE R")
            c2=Car("Porsche 911")
            c1.drive()
            c1.driveIntoDitch(2)
            c1.drive()

            print(c2.getNumWheels())
```

只是一个蓝图，只是在为未来的汽车类用户编写模板，让他们可以从零开始创建全新的汽车

```java
public class Car{

    public String model;
    public int wheels;
    //如果缺少public有什么影响呢？
    public Car(String m){
        this.model=m;
        this.wheels=4;
    }
    public void drive(){
        if(this.wheels<4){
            System.out.println(this.model+" no go vroom");
            return;
        }
        System.out.println(this.model+" go vroom");
    }

    public int getNumWheels(){
        return this.wheels;
    }

    public void driveIntoDitch(int wheelsLost){
        this.wheels=this.wheels-wheelsLost;
    }

    public static void main(String []args){
        Car c1;
        Car c2;

        c1=new Car("Porsche 911");
        c2=new Car("Toyota Camry");

        c1.drive();
        c1.driveIntoDitch(2);
        c1.drive();

        System.out.println(c2.getNumWheels());
    }
};
```

在Cpp中则是：

```cpp
#include<iostream>
class Car{
public:
    string model;
    int wheels;
    Car(string m){
        this.model=m;
        this.wheels=4;
    }
    void drive(){
        if(this.wheels<4){
            std::cout<<this.model<<" no vroom."<<endl;
            return;
        }
        std::cout<<this.model<<" vroom."<<endl;
    }
    int getNumWheels(){
        return this.wheels;
    }

    void driveIntoDitch(int wheelsLost){
        this.wheels=this.wheels-wheelsLost;
    }
}
int main(){
    Car c1;
    Car c2;

    c1=new Car("Toyata Camry");
    c2=new Car("Porsche 911");//玩过2077的都知道这车多帅

    c1.drive();
    c1.driveIntoDitch(2);
    c1.drive();

    c2.drive();
    std::cout<<c2.getNumberWheels()<<std::endl;

    delete c1;
    delete c2;//avoid(shield/protect) the memory leak

    return 0;
}
```



- Calling a method on an object might change its state
- The object knows how to manage its own state, based on method calls
- In Java: The return value of the method must have a type

Car.java

```java
public class Car {
    public void drive() {
        if (gas <= 0) {
            System.out.println("Cannot drive!");
            return;
        }
        gas -= 5;
        System.out.println(model + " goes vroom!");
    }

    public int gasLeft() {
        return gas;
    }

    public void addGas(int amount) {
        gas = gas + amount;
    }
}
```

car.py

```python
class Car:
    def drive(self):
        if (gas <= 0):
            print("Cannot drive!")
            return

        gas -= 5
        print(self.model + " goes vroom!")


    def gasLeft(self):
        return self.gas


    def addGas(self, amount):
        gas = gas + amount
```