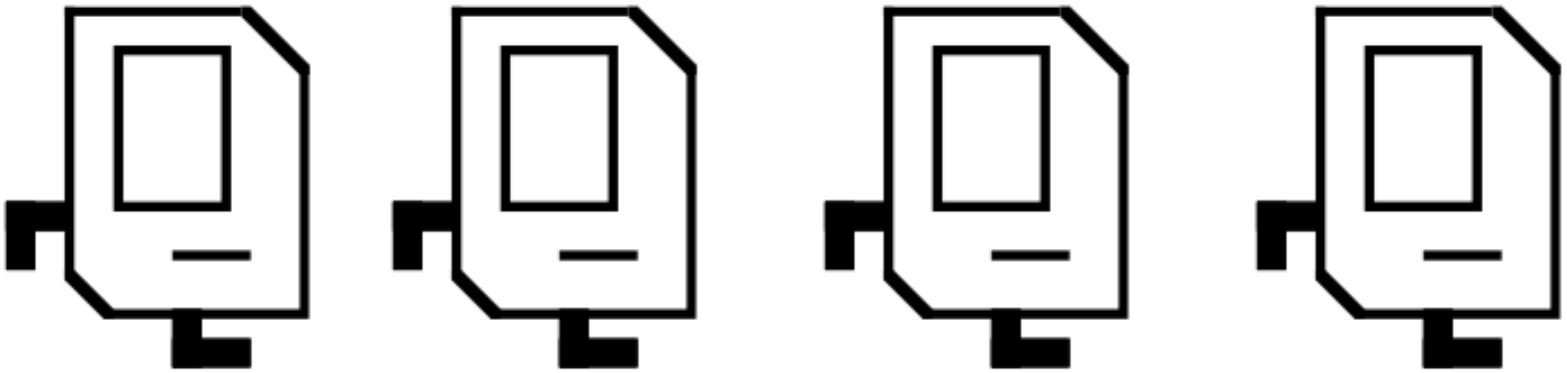
Introduction to Java

Announcements

- Programming Assignment #1 Out:
  - Karel the Robot: Due Friday, January 20 at 3:15 PM.
  - Email: Due Sunday, January 22 at 11:59PM.
- Section Assignments Posted
  - Check online at **http://cs198.stanford.edu**.
  - Sections and LaIR hours start this week.
  - Feel free to switch sections to an open time if it works better for you; your fellow CS106Aers are counting on you!
- Submitting Assignments:
  - Submitter now open.
  - Verify submissions at

http://paperless.stanford.edu

A Farewell to Karel



Welcome to Java

But First...

A Brief History of Computing

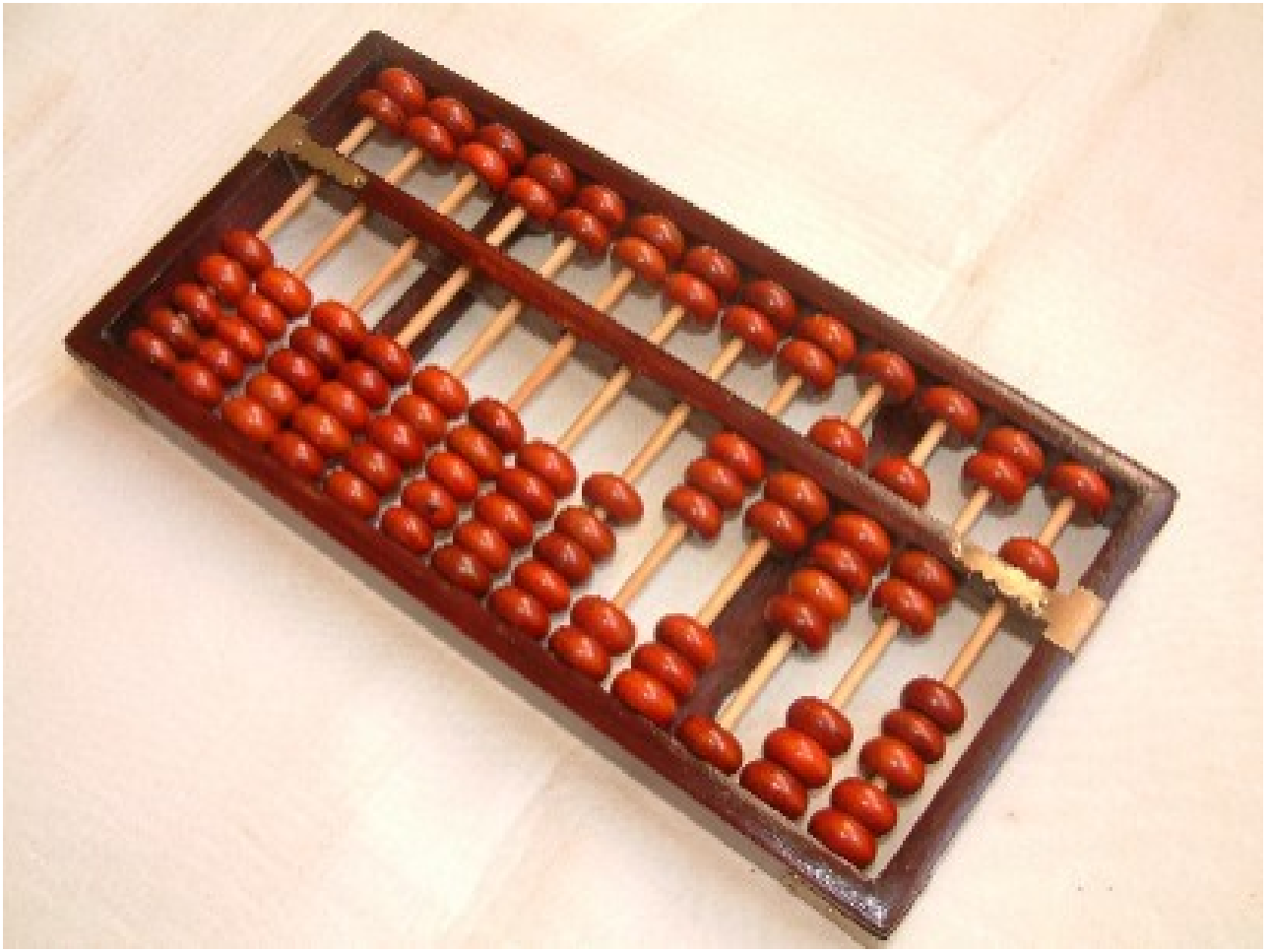


Image credit: http://en.wikipedia.org/wiki/File:Boulier1.JPG

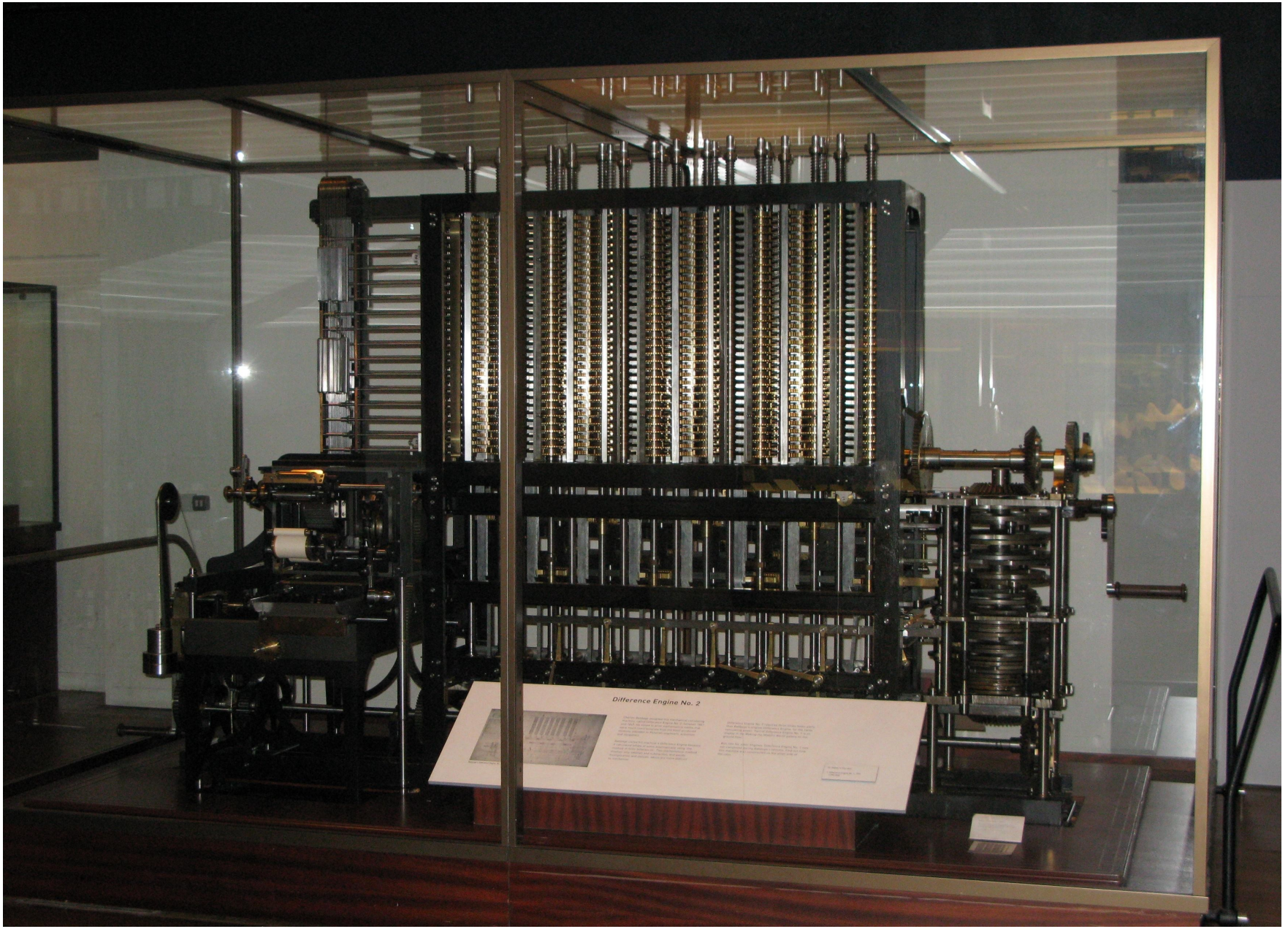


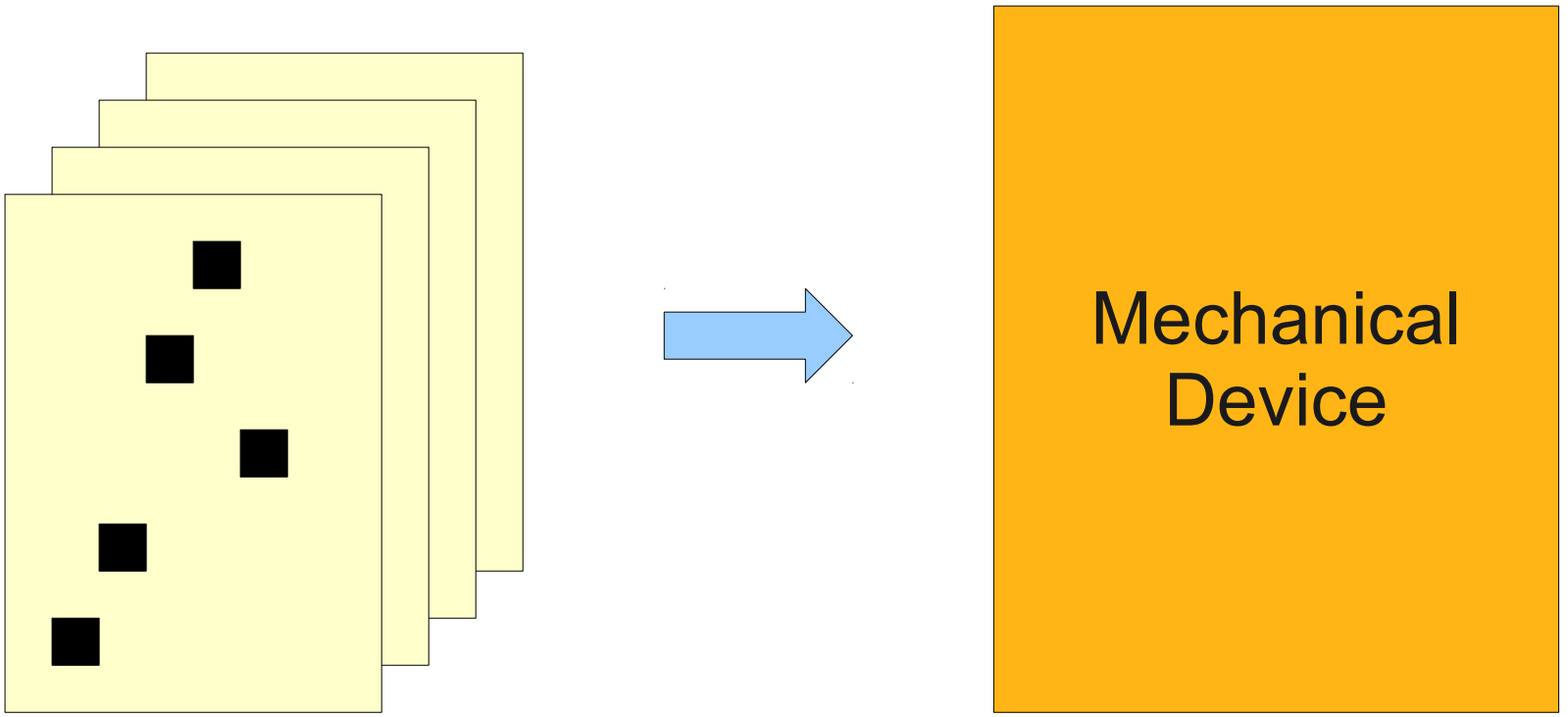
Image: http://upload.wikimedia.org/wikipedia/commons/8/8b/Babbage_Difference_Engine.jpg

http://upload.wikimedia.org/wikipedia/commons/thumb/a/a4/Analytical_Engine_(2290032530).jpg/200px-Analytical_Engine_(2290032530).jpg

Augusta Ada Byron: The First Programmer

Image Credit: http://upload.wikimedia.org/wikipedia/commons/0/0f/Ada_lovelace.jpg

Programming in the 1800s



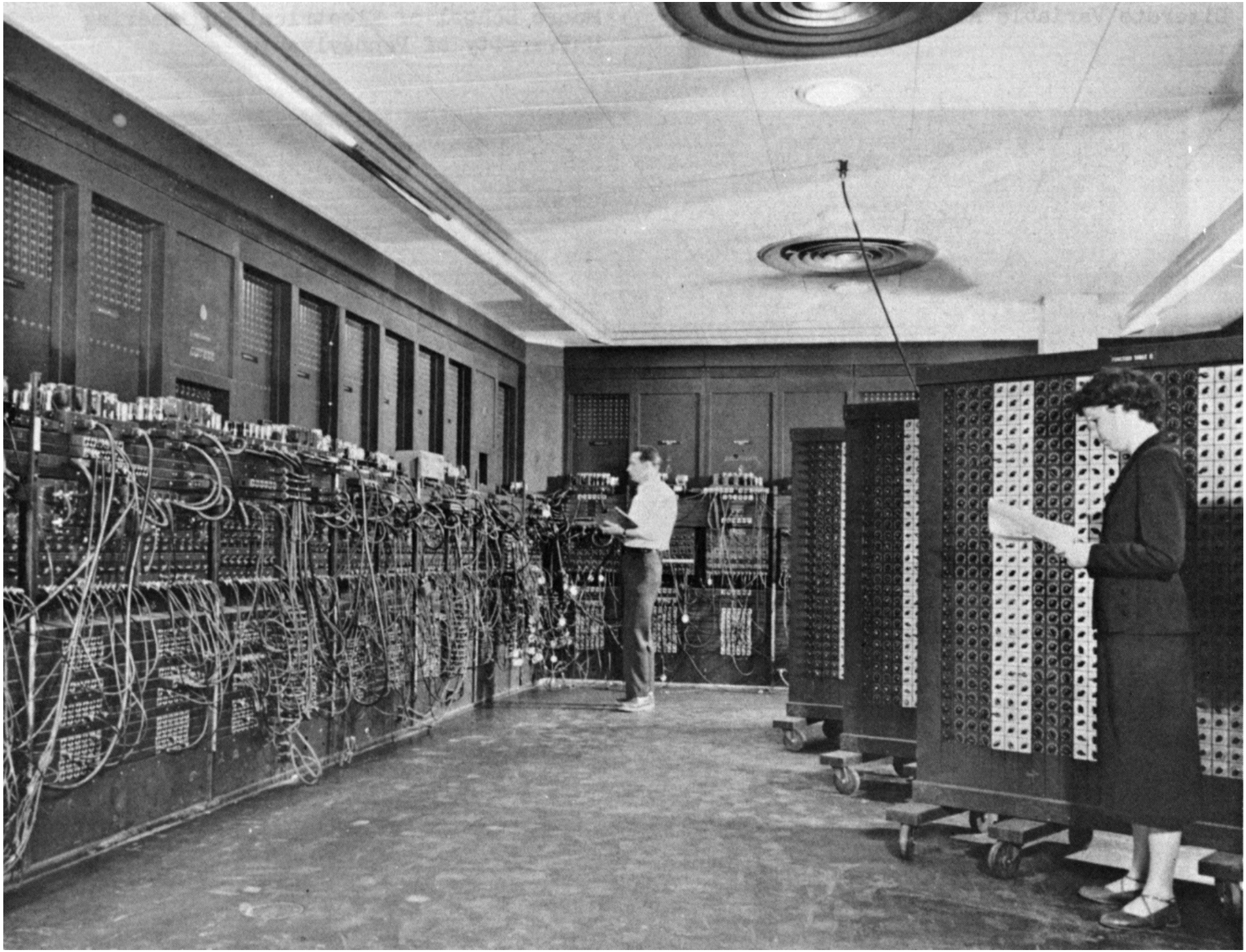
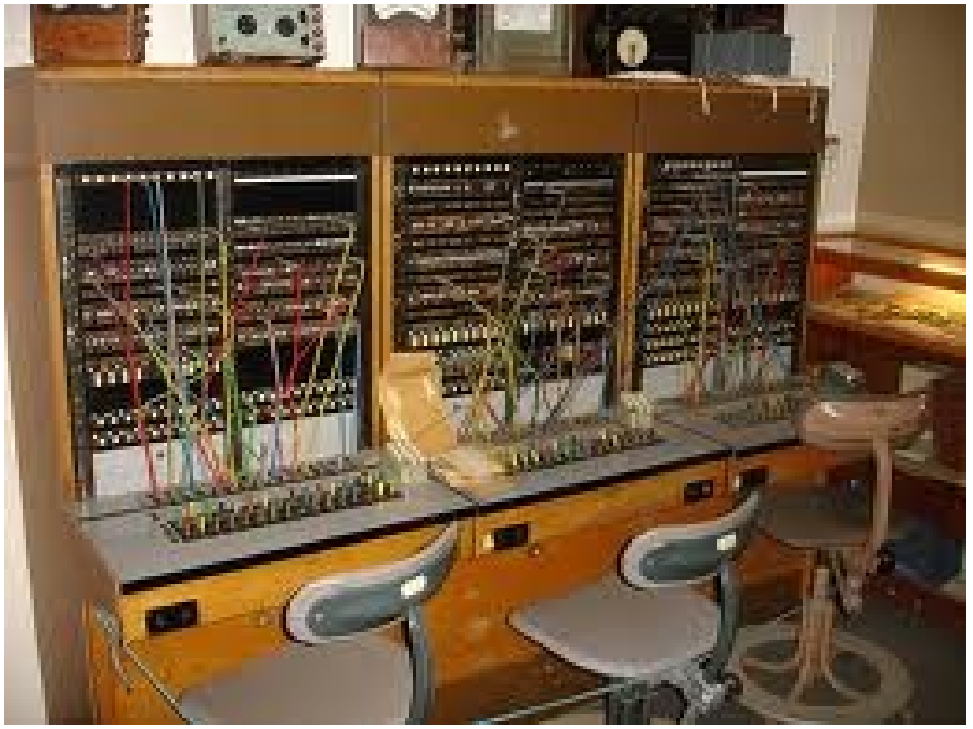
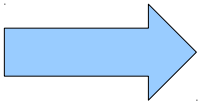


Image credit: http://upload.wikimedia.org/wikipedia/commons/4/4e/Eniac.jpg

Programming in the 1940s





Electrical Device

High-Level Languages

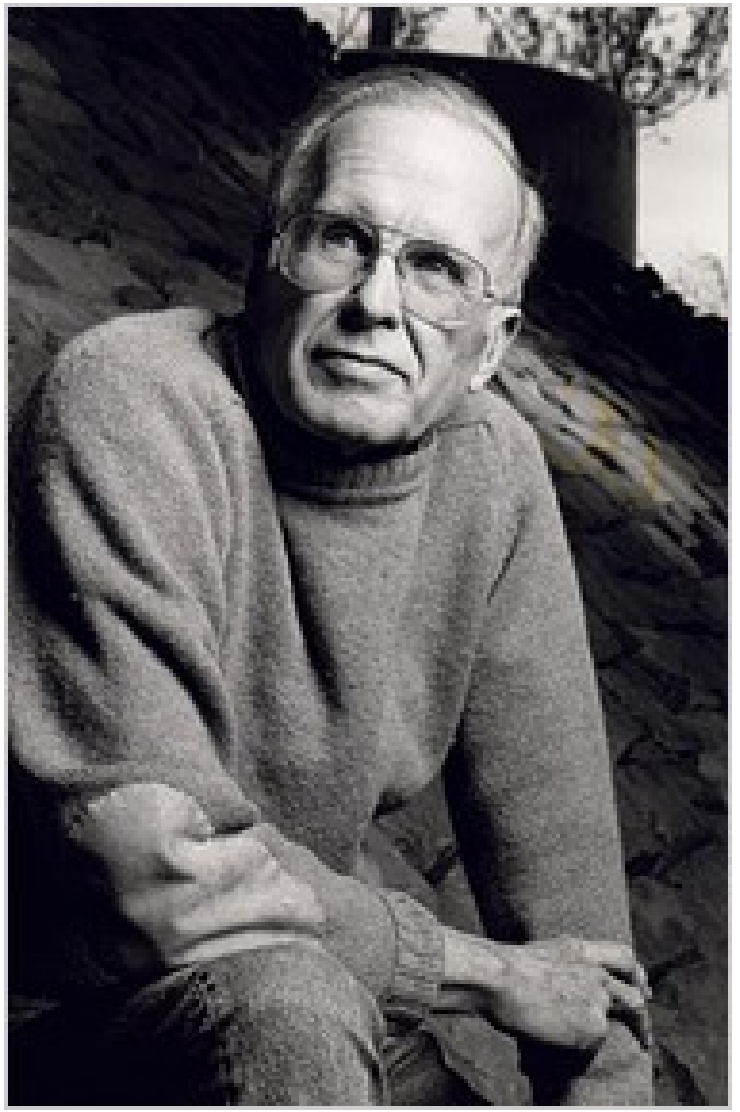
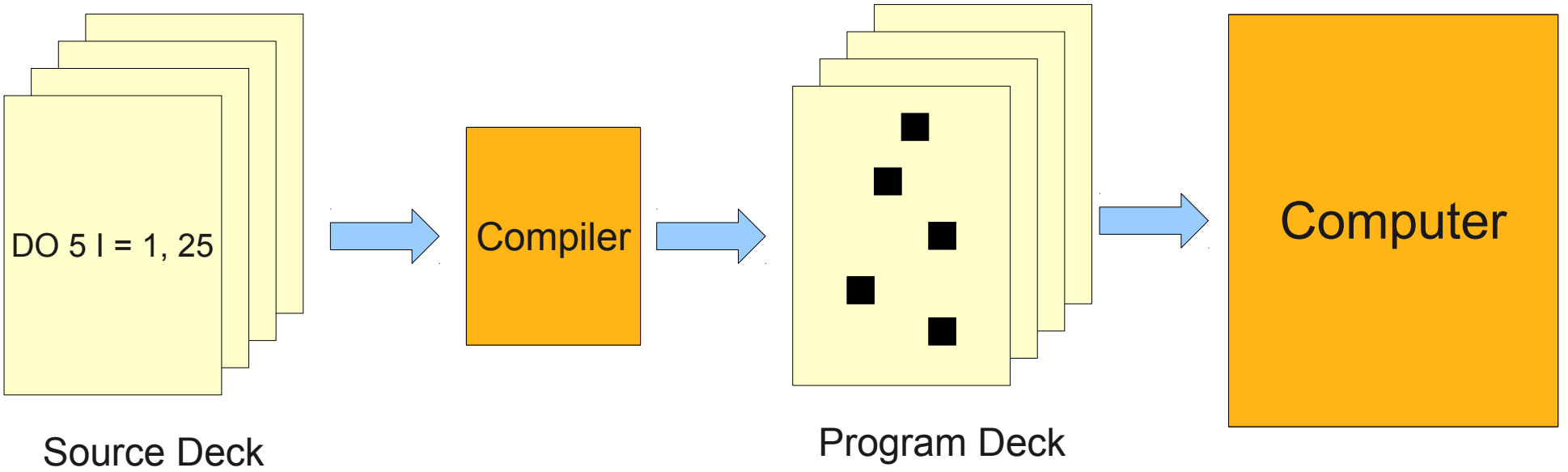
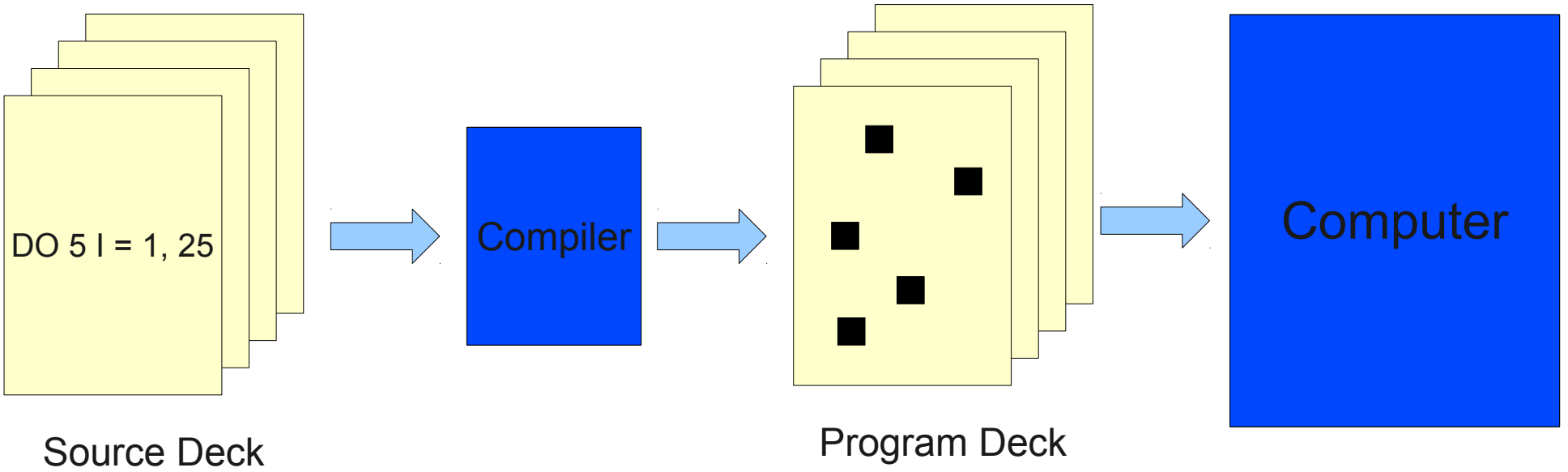


Image: http://upload.wikimedia.org/wikipedia/commons/thumb/5/55/Grace_Hopper.jpg/300px-Grace_Hopper.jpg
http://www.nytimes.com/2007/03/20/business/20backus.html

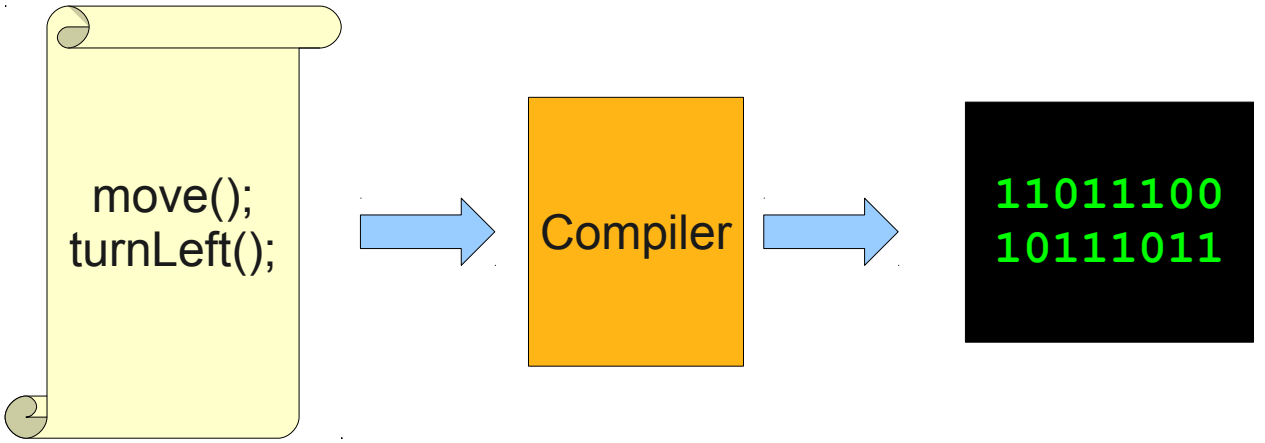
Programming in the 1950s



Programming in the 1950s



Programming Now(ish)

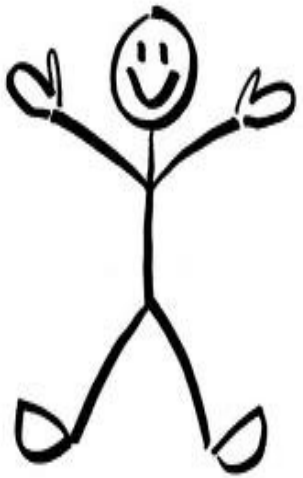


Source Code

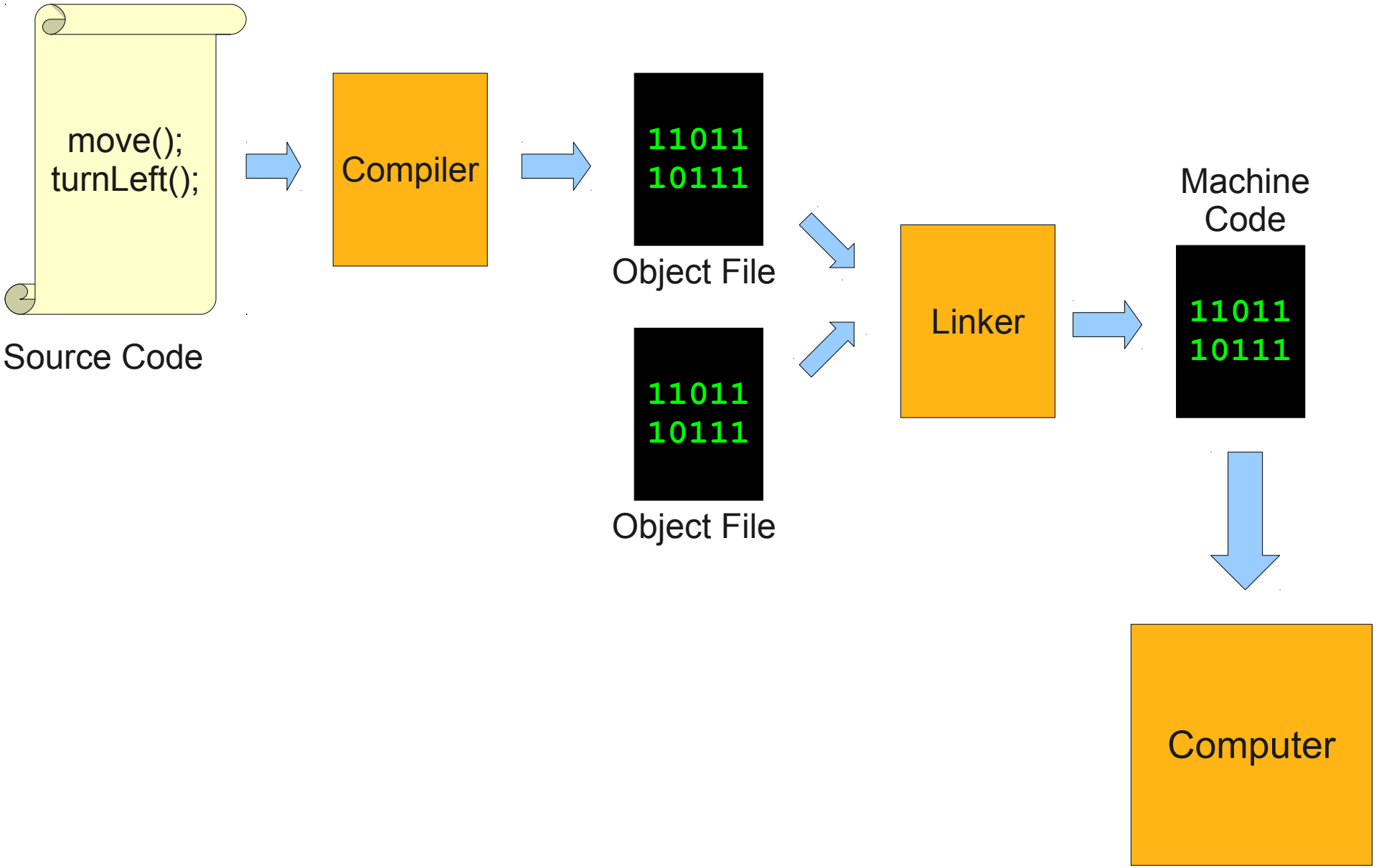
Machine Code

Hey! I wrote a program that can draw stick figures!

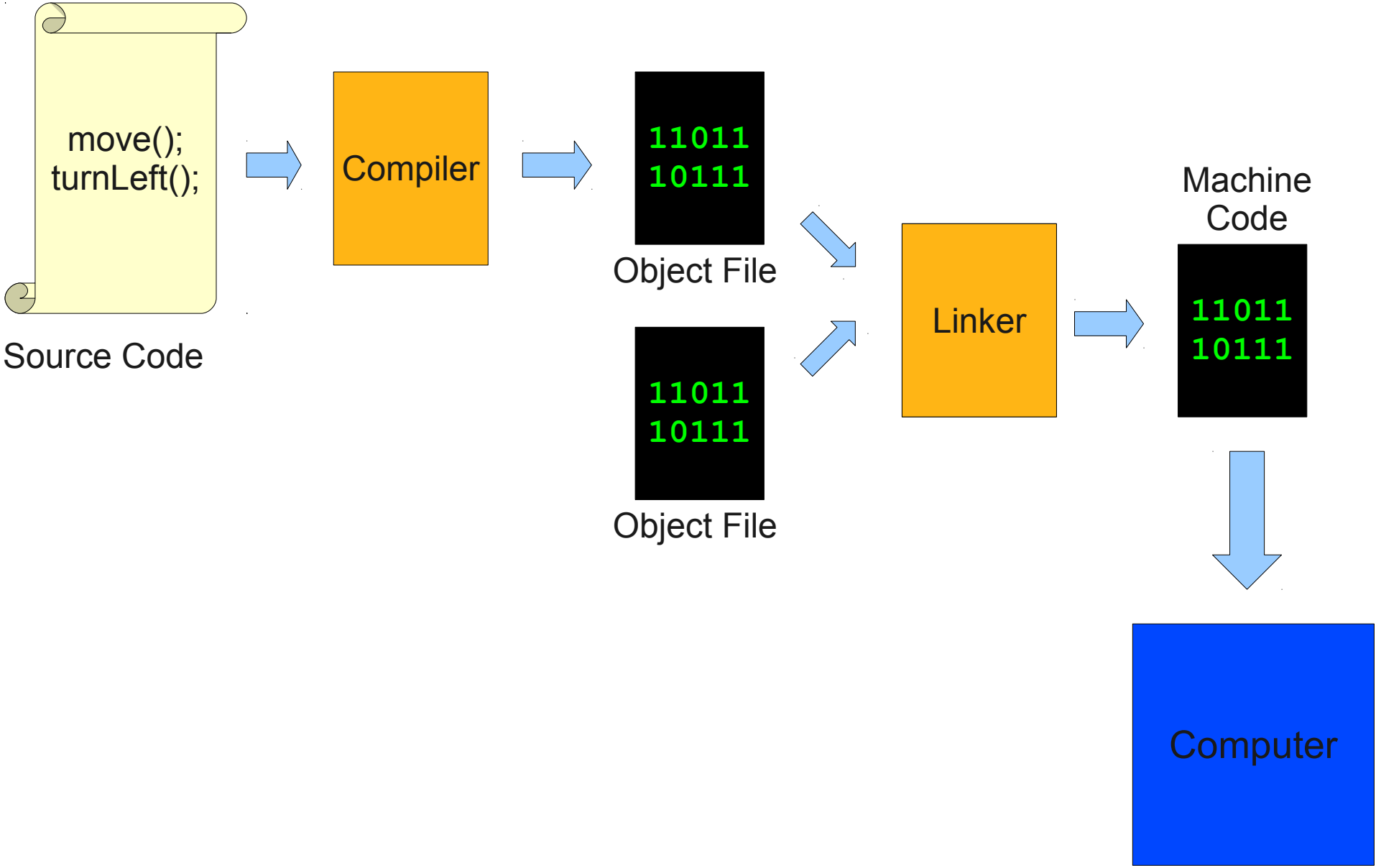
That's great! I wrote a program that makes speech bubbles!



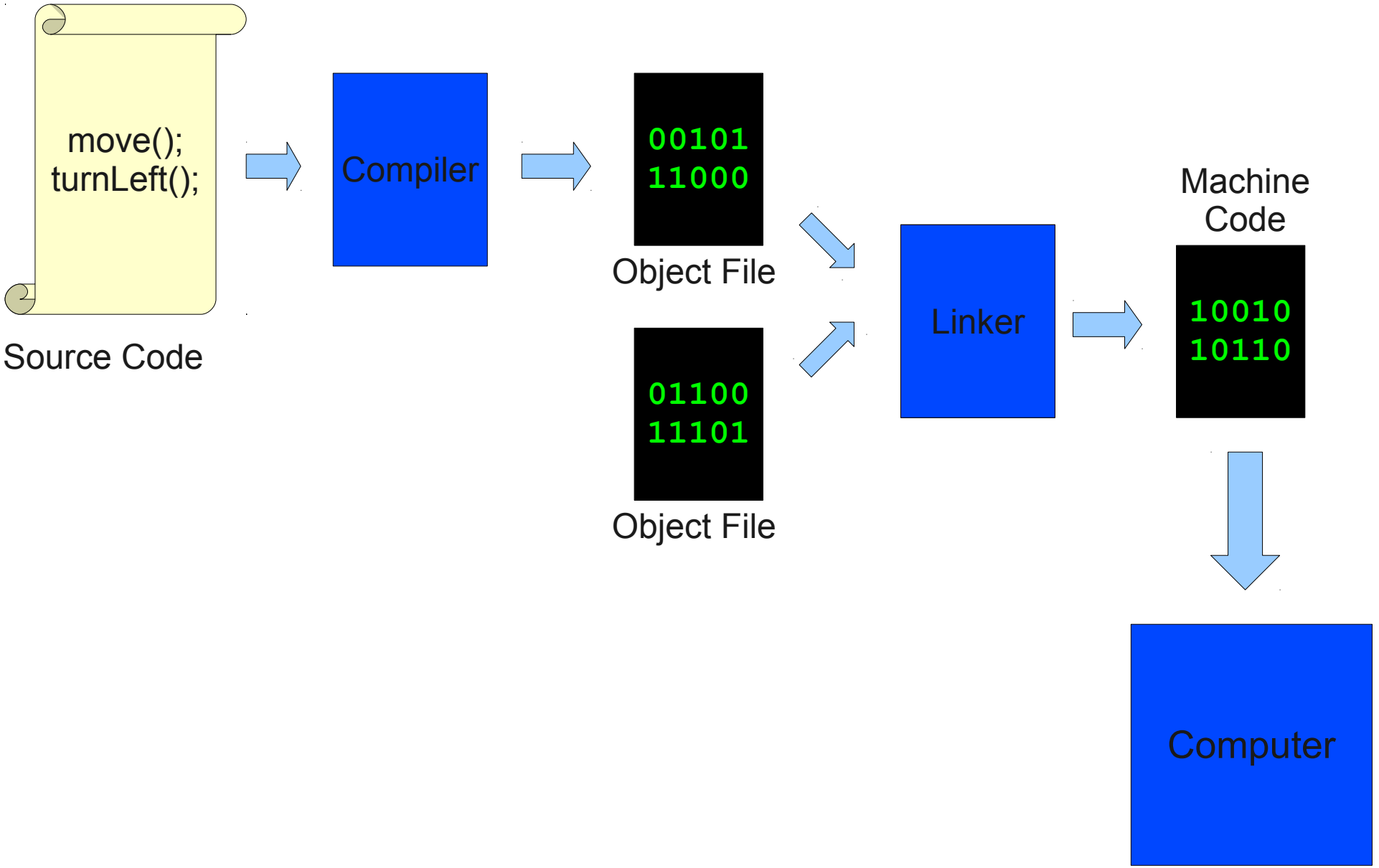
Programming Now



Programming Now



Programming Now



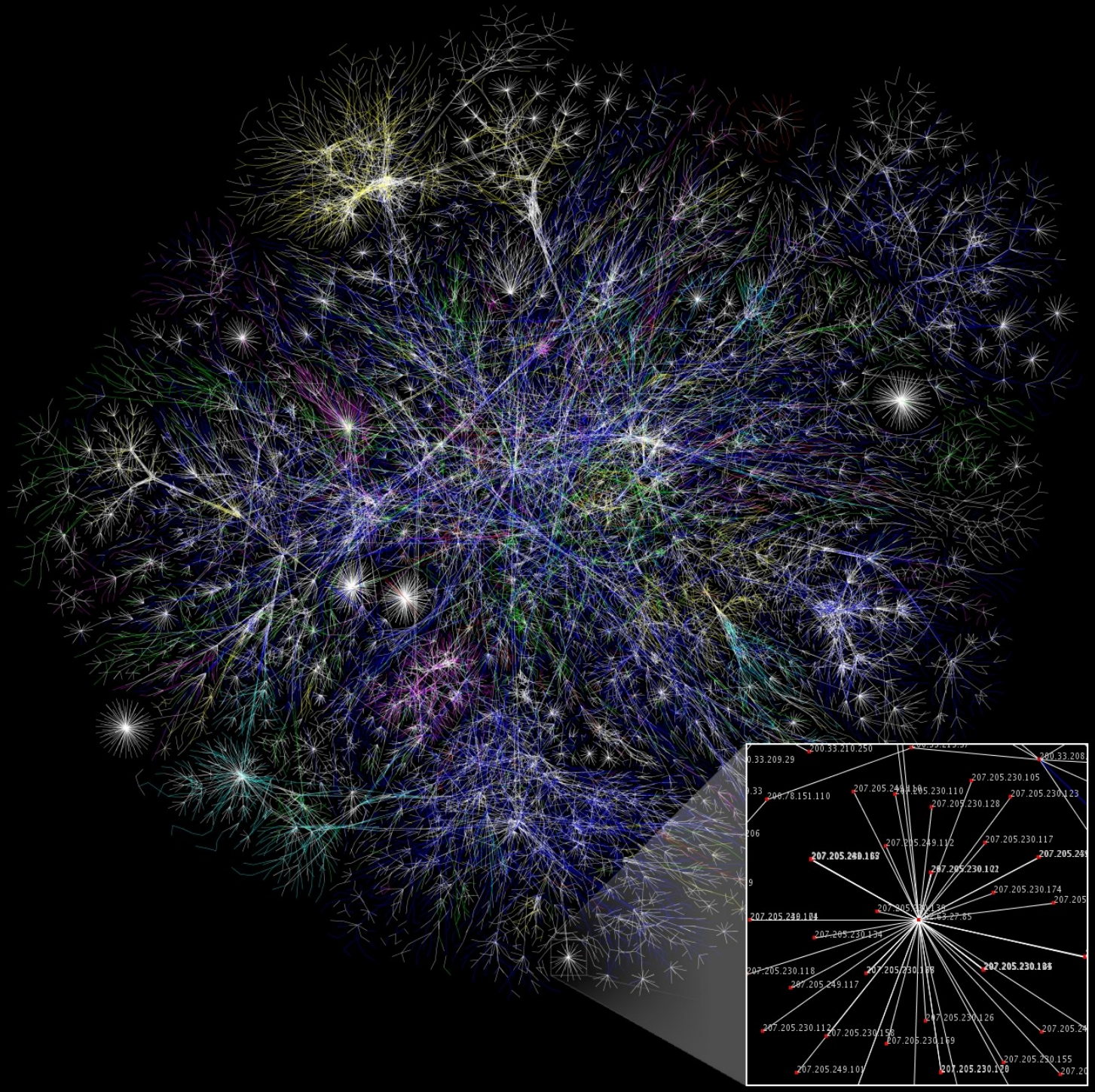
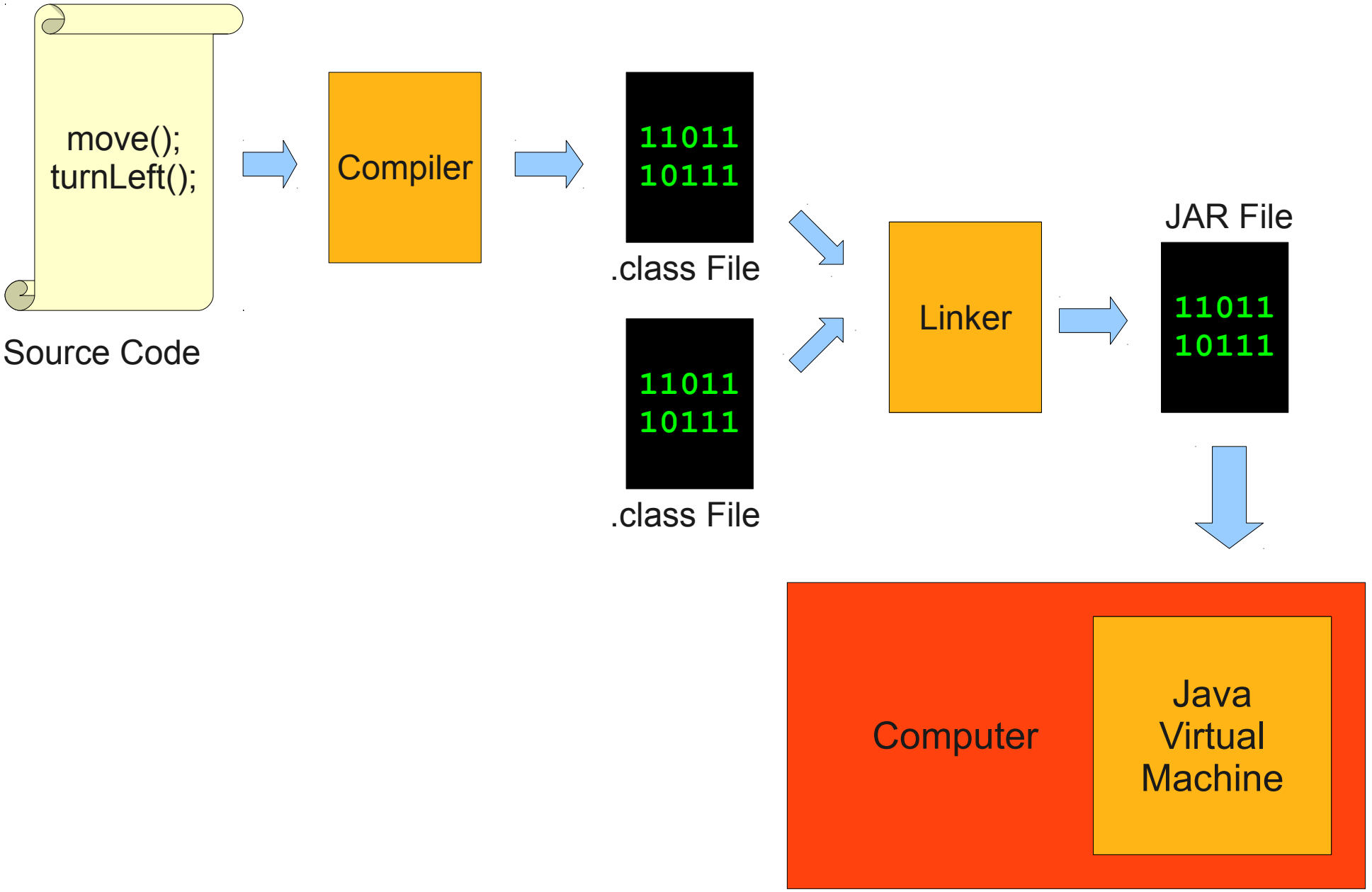


Image credit: http://upload.wikimedia.org/wikipedia/commons/d/d2/Internet_map_1024.jpg

The Java Model



Object-Oriented Programming

An **object** is an entity
that has state and behavior.

Has a fur color.
Has an energy level.
Has a level of cuteness.
Can be your friend.
Can sit.
Can stay.
Can bark.

A **class** is a set of features and behavior common to a group of objects.

Class Dog

Has a fur color.
Has an energy level.
Has a level of cuteness.
Can sit.
Can stay.
Can be your friend.

Instances of **Dog**

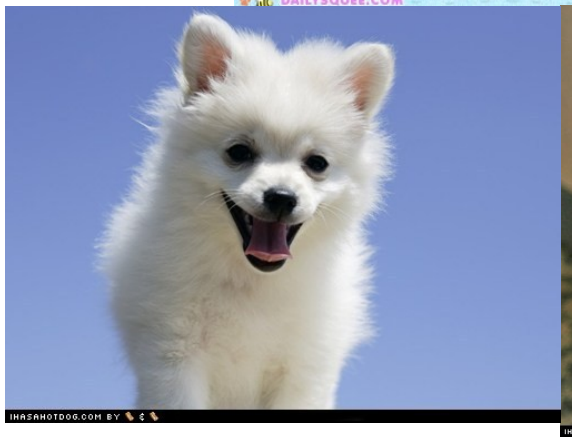
An **object** is an entity
that has state and behavior.

A **class** is a set of features and behavior
common to a group of objects.

An **instance of a class** is an
object that belongs to that class.

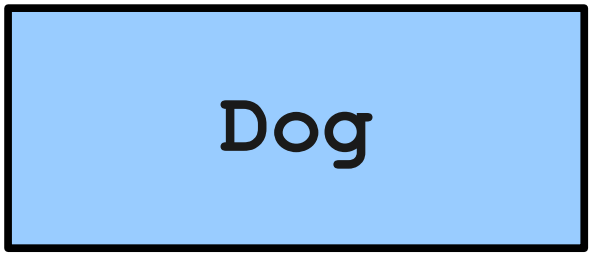
Class Dog

Has a fur color.
Has an energy level.
Has a level of cuteness.
Can sit.
Can stay.
Can be your friend.

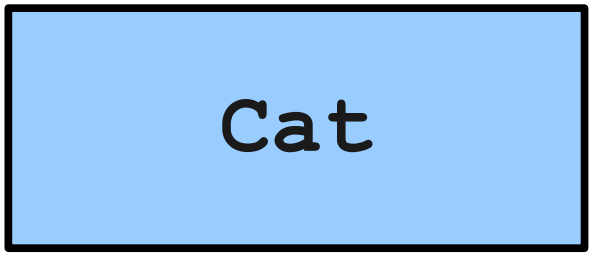


Class Cat

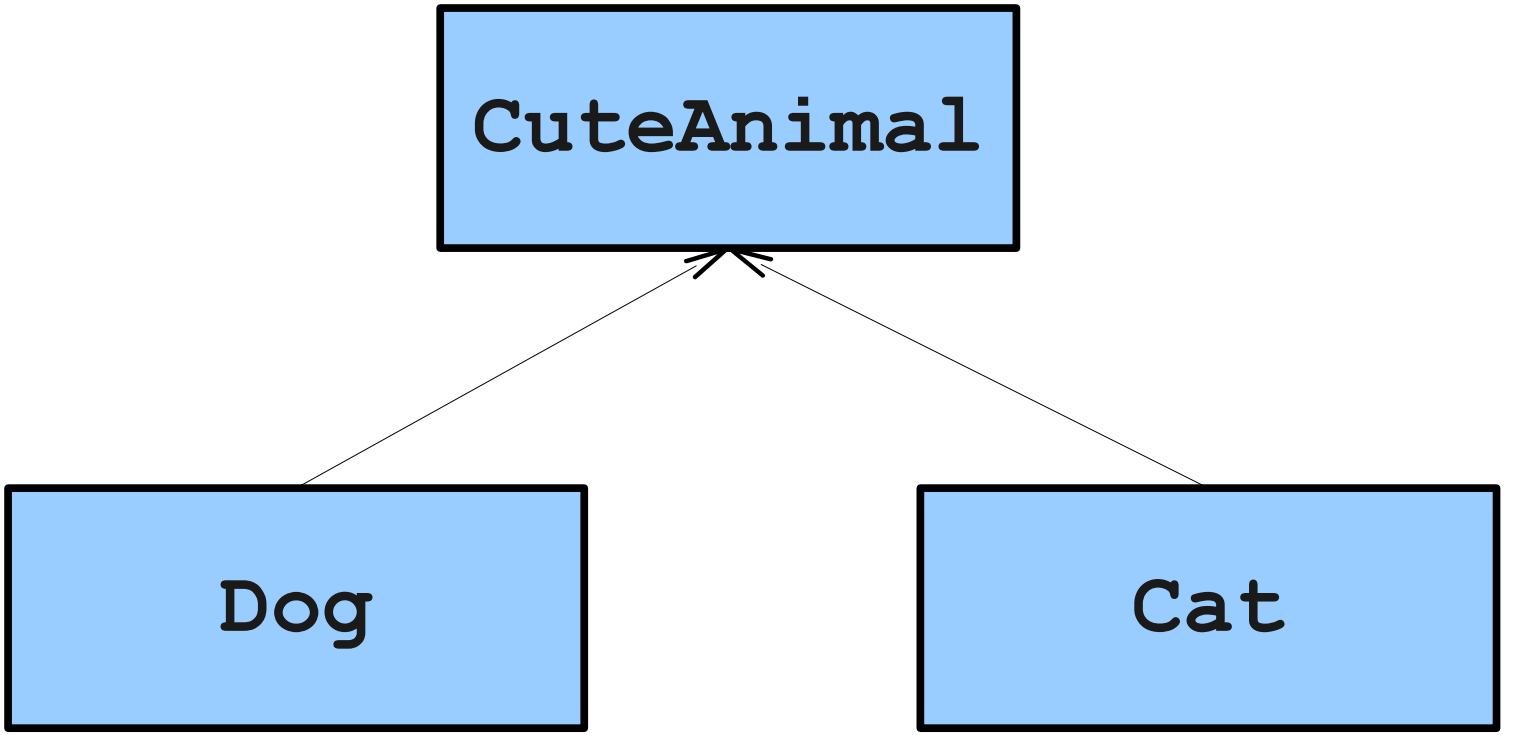
Has a fur color.
Has an energy level.
Has a level of cuteness.
Can purr.
Can haz cheezburger?



Has a fur color.
Has an energy level.
Has a level of cuteness.
Can sit.
Can stay.
Can be your friend.

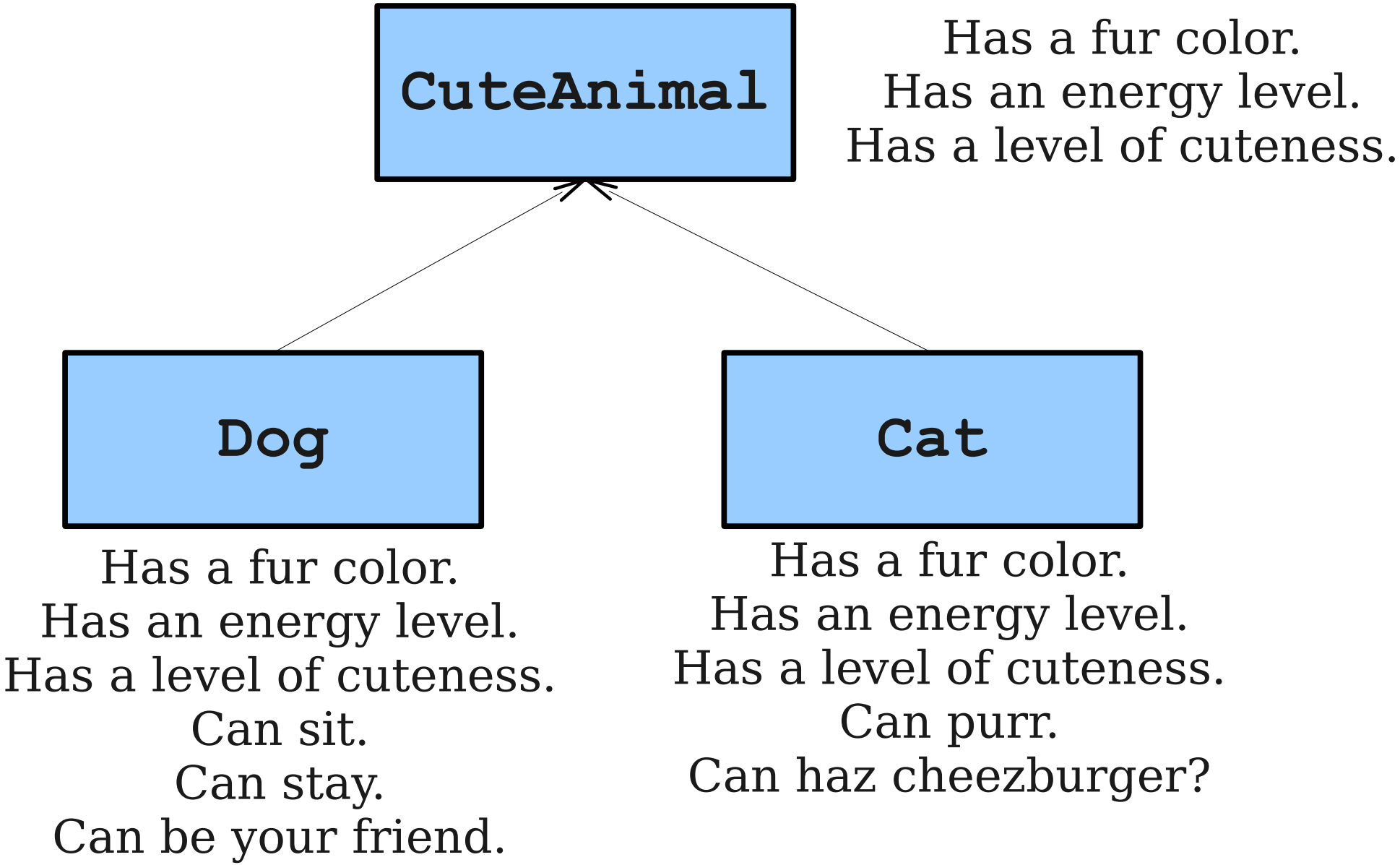


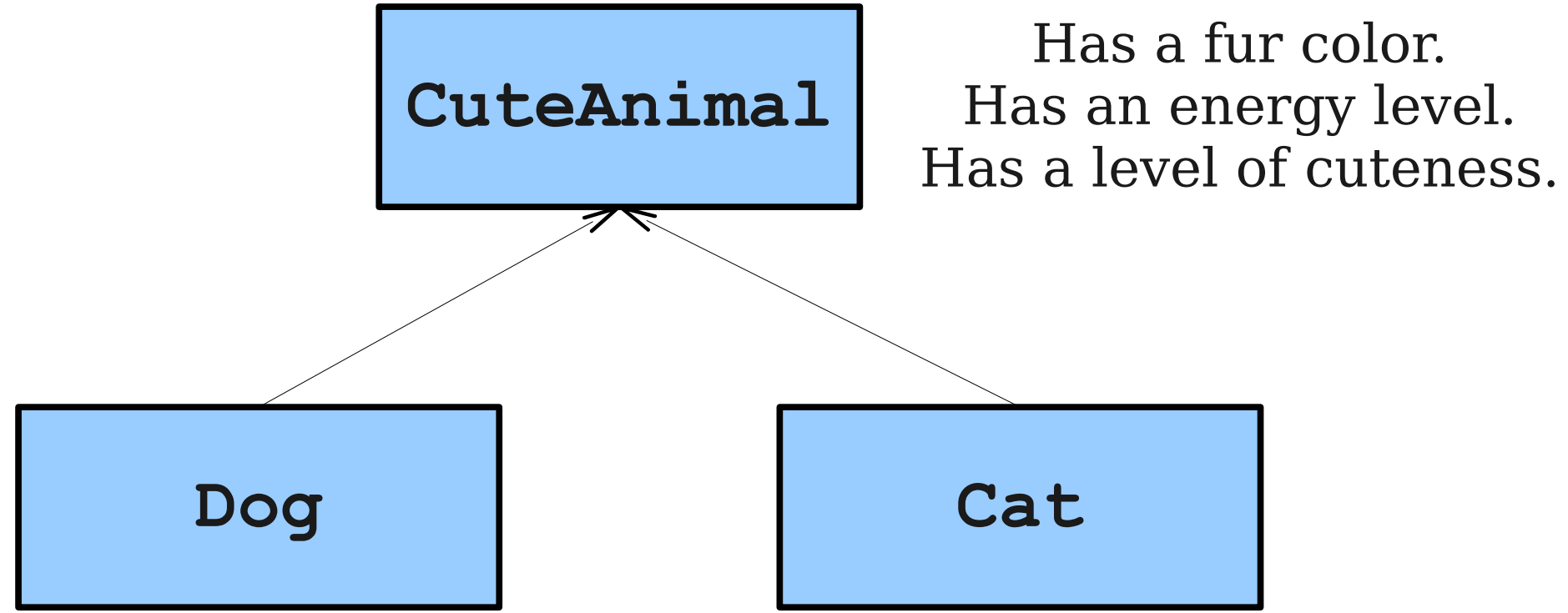
Has a fur color.
Has an energy level.
Has a level of cuteness.
Can purr.
Can haz cheezburger?



Has a fur color.
Has an energy level.
Has a level of cuteness.
Can sit.
Can stay.
Can be your friend.

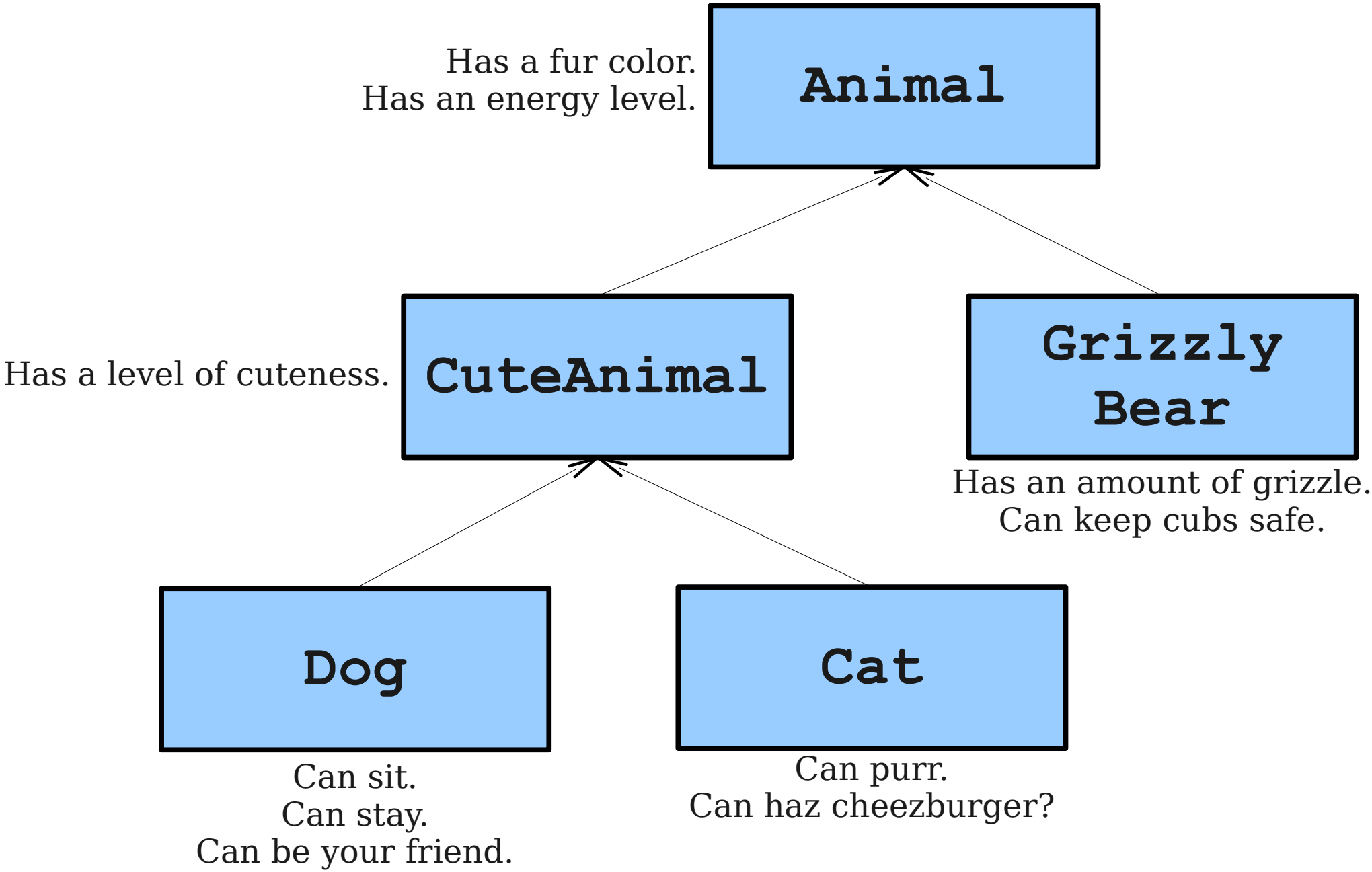
Has a fur color.
Has an energy level.
Has a level of cuteness.
Can purr.
Can haz cheezburger?


CuteAnimal
Has a fur color.
Has an energy level.
Has a level of cuteness.
Dog
Has a fur color.
Has an energy level.
Has a level of cuteness.
Can sit.
Can stay.
Can be your friend.
Cat
Has a fur color.
Has an energy level.
Has a level of cuteness.
Can purr.
Can haz cheezburger?



The class **`Dog`** and **`Cat`** classes are **subclasses** of the **`CuteAnimal`** class.

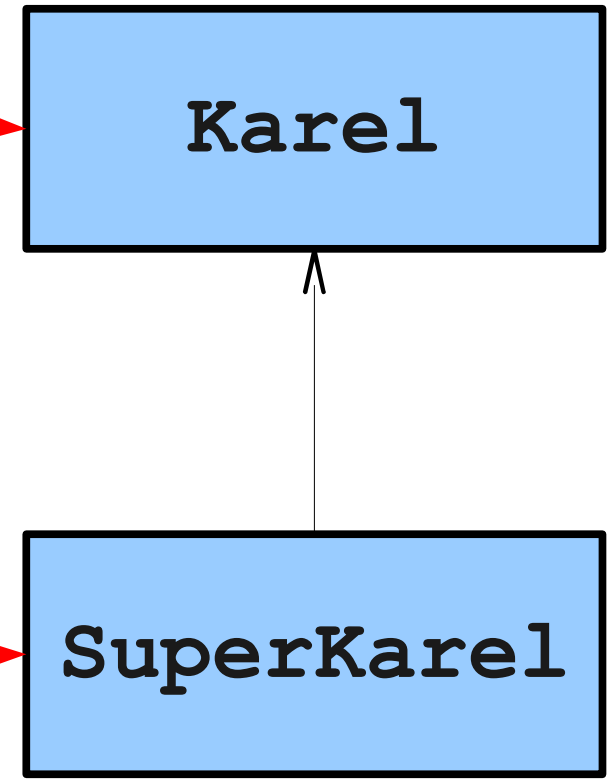
A Class Hierarchy



Classes so Far

Superclass

Subclass



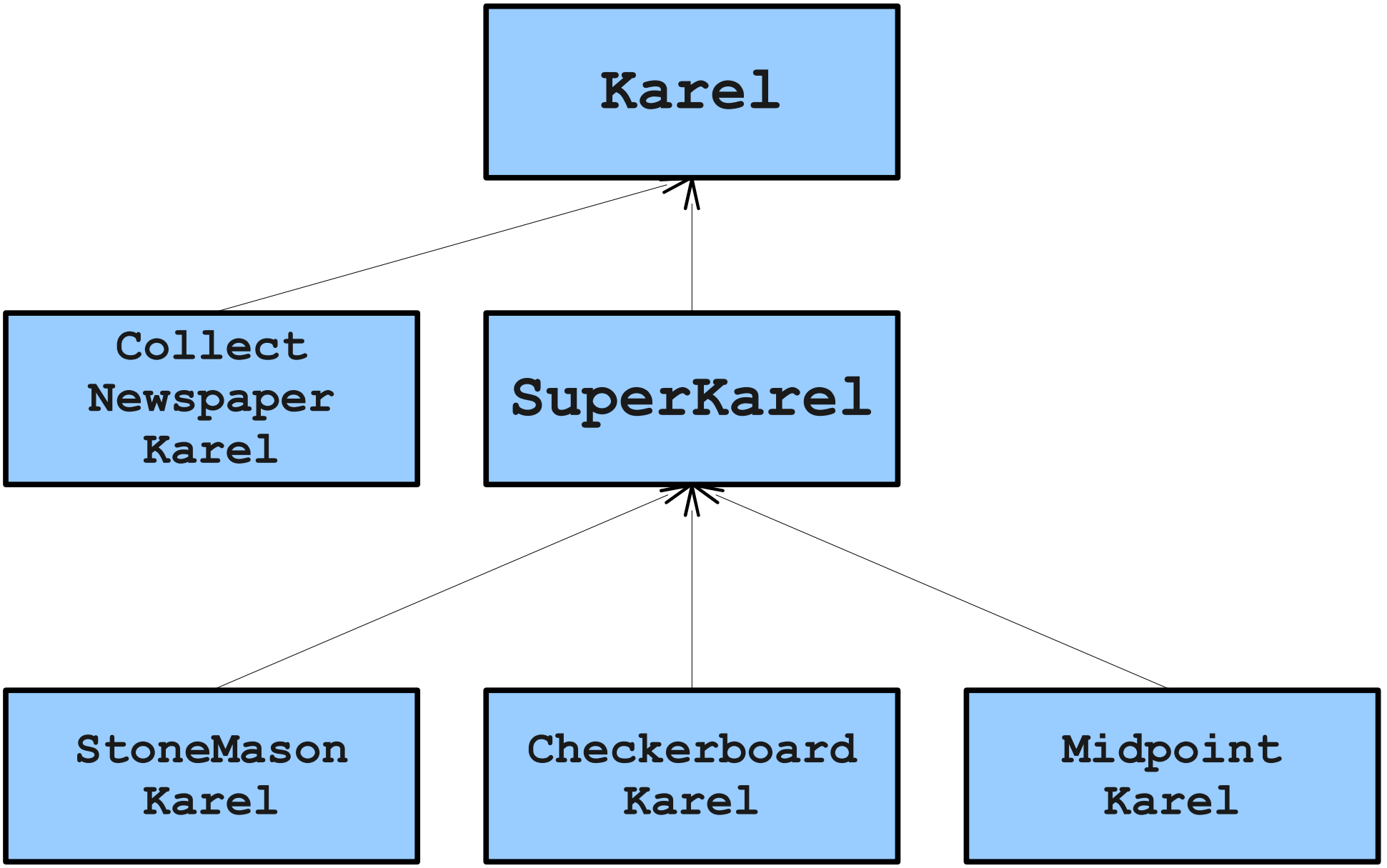

```
move
turnLeft
pickBeeper
putBeeper
```

```
turnAround
turnRight
```

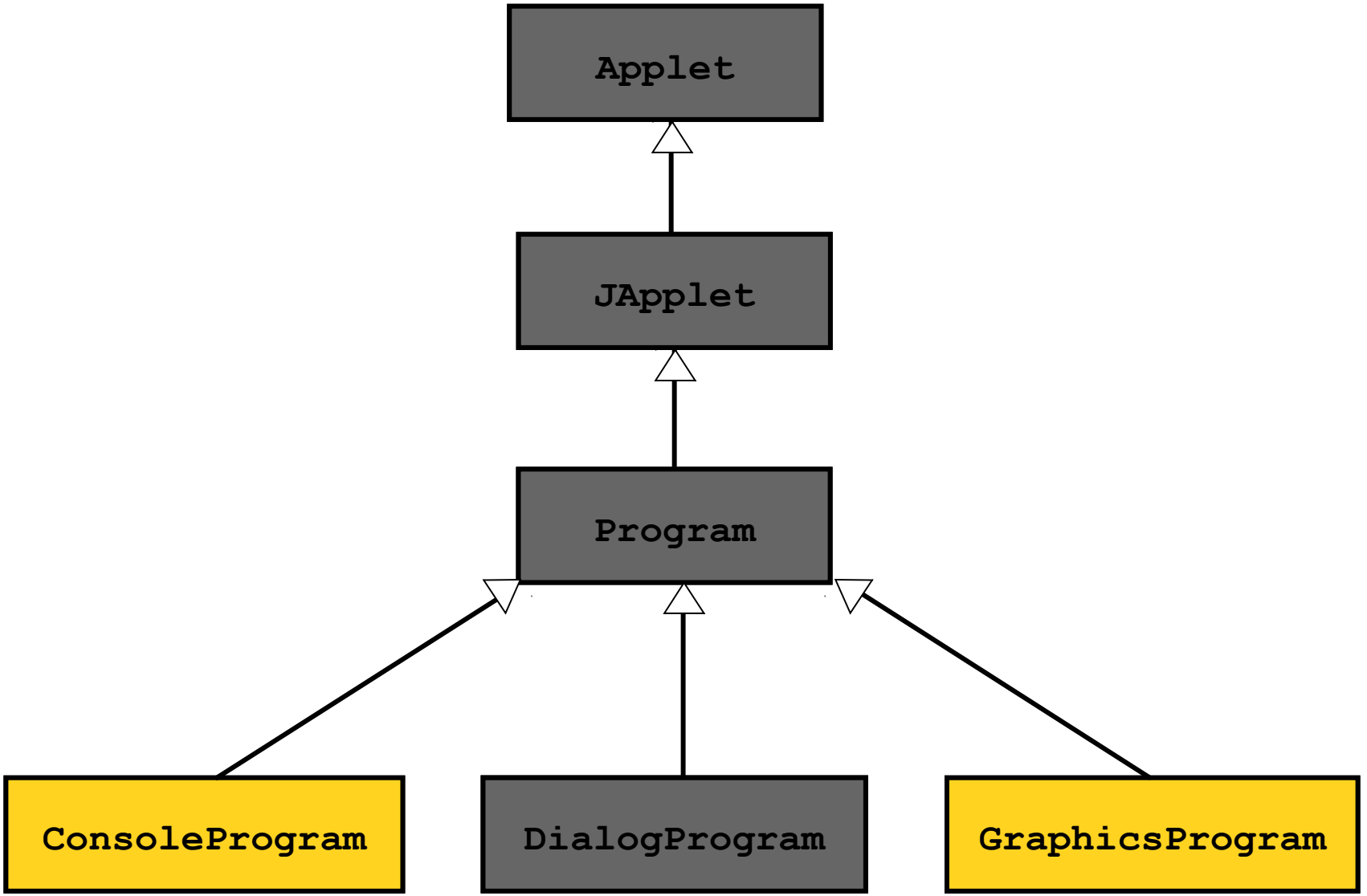
```
/* File: RoombaKarel.java
 *
 * A Karel program in which Karel picks up all the beepers in a
 * square world.
 */

import stanford.karel.*;

public class RoombaKarel extends SuperKarel {
    public void run() {
        while (leftIsClear()) {
            cleanOneRow();
            moveToNextRow();
        }
        cleanOneRow();
    }

    /* Precondition:  Karel is facing East at the start of a row.
     * Postcondition: Karel is facing East at the start of a row,
     *                but the row has all beepers cleared from it
     */
    private void cleanOneRow() {
        sweepRow();
        moveBackToStart();
    }

    /* ... */
}
```

```
/* File: RoombaKarel.java
 *
 * A Karel program in which Karel picks up all the beepers in a
 * square world.
 */

import stanford.karel.*;

public class RoombaKarel extends SuperKarel {
    public void run() {
        while (leftIsClear()) {
            cleanOneRow();
            moveToNextRow();
        }
        cleanOneRow();
    }

    /* Precondition:  Karel is facing East at the start of a row.
     * Postcondition: Karel is facing East at the start of a row,
     *                but the row has all beepers cleared from it
     */
    private void cleanOneRow() {
        sweepRow();
        moveBackToStart();
    }

    /* ... */
}
```

How Does Karel Fit In?



`acm.program` Hierarchy



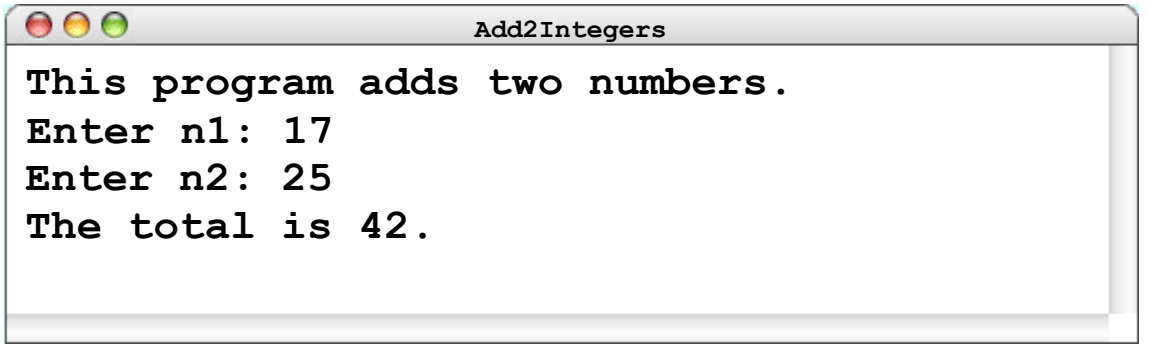
Graphic courtesy of Eric Roberts

Let's See Some Java!

The **Add2Integers** Program

```
class Add2Integers extends ConsoleProgram {
   public void run() {
      println("This program adds two numbers.");
      int n1 = readInt("Enter n1: ");
      int n2 = readInt("Enter n2: ");
      int total = n1 + n2;
      println("The total is " + total + ".");
   }
}
```

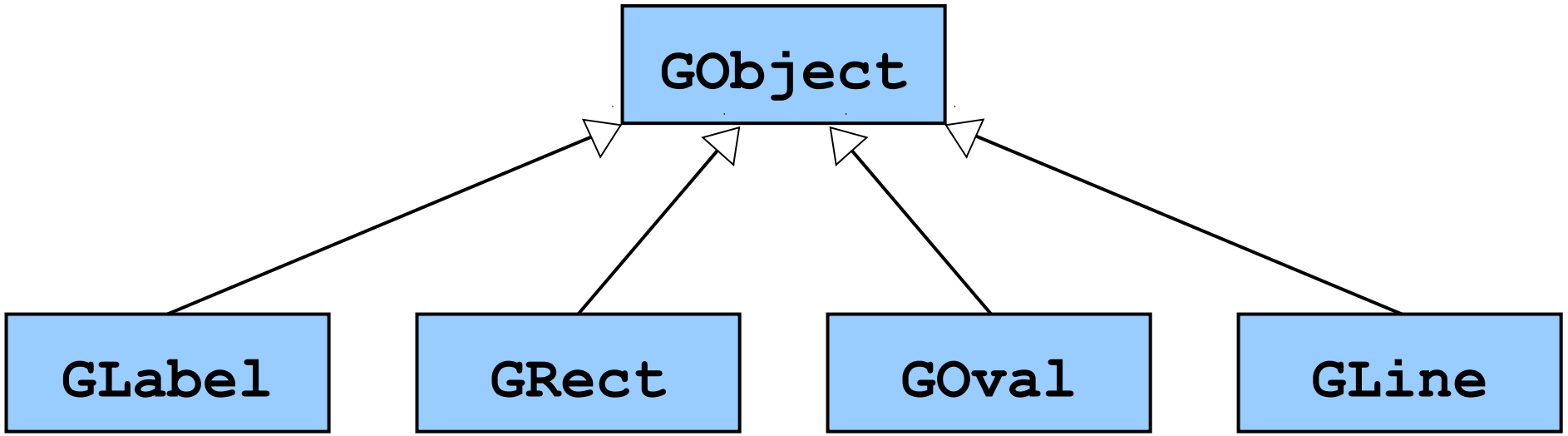
n1	n2	total
17	25	42



Graphic courtesy of Eric Roberts

The `GObject` Hierarchy

The classes that represent graphical objects form a hierarchy, part of which looks like this:



Graphic courtesy of Eric Roberts

Sending Messages to a `GLabel`

```
public class HelloProgram extends GraphicsProgram {
   public void run() {
      GLabel label = new GLabel("hello, world", 100, 75);
      label.setFont("SansSerif-36");
      label.setColor(Color.RED);
      add(label);
   }
}
```

label

hello, world

Graphic courtesy of Eric Roberts