OO software systems are systems of interacting objects.

Objects have

- properties:
  - these are things that objects know
  - e.g. what you had for breakfast
- behaviors:
  - these are things objects do
  - e.g. being able to reply to the question “What did you have for breakfast?”

University at Buffalo
The State University of New York

cse@buffalo

CSE115 / CSE503
Introduction to Computer Science I

Dr. Carl Alphonce

343 Davis Hall

alphonce@buffalo.edu

Office hours:

Tuesday 10:00 AM – 12:00 PM*

Wednesday 4:00 PM – 5:00 PM

Friday 11:00 AM – 12:00 PM

OR request appointment via e-mail

**Tuesday adjustments: 11:00 AM – 1:00 PM on 10/11, 11/1 and 12/6*

Sapphire project

Dr. Bina Ramamurthy

ANNOUNCEMENTS

UB ACM
Association for Computing Machinery

&

UB Robotics

BBQ

Friday, September 9th, 2016
Starts at 6:00pm
Greiner Patio

OPEN TO EVERYONE!
JOIN US FOR...

Burgers Veggie Burgers Hot Dogs
Ice Cream Giant Jenga Frisbee
Remote Control Racecars

SA STUDENT ASSOCIATION

ROADMAP

Class today

- Announcements (Robotics BBQ, Sapphire project)
- Instruction decoding
- Fetch/Decode/Execute cycle
- Low-level and high-level languages

Coming up

- Expressions and objects

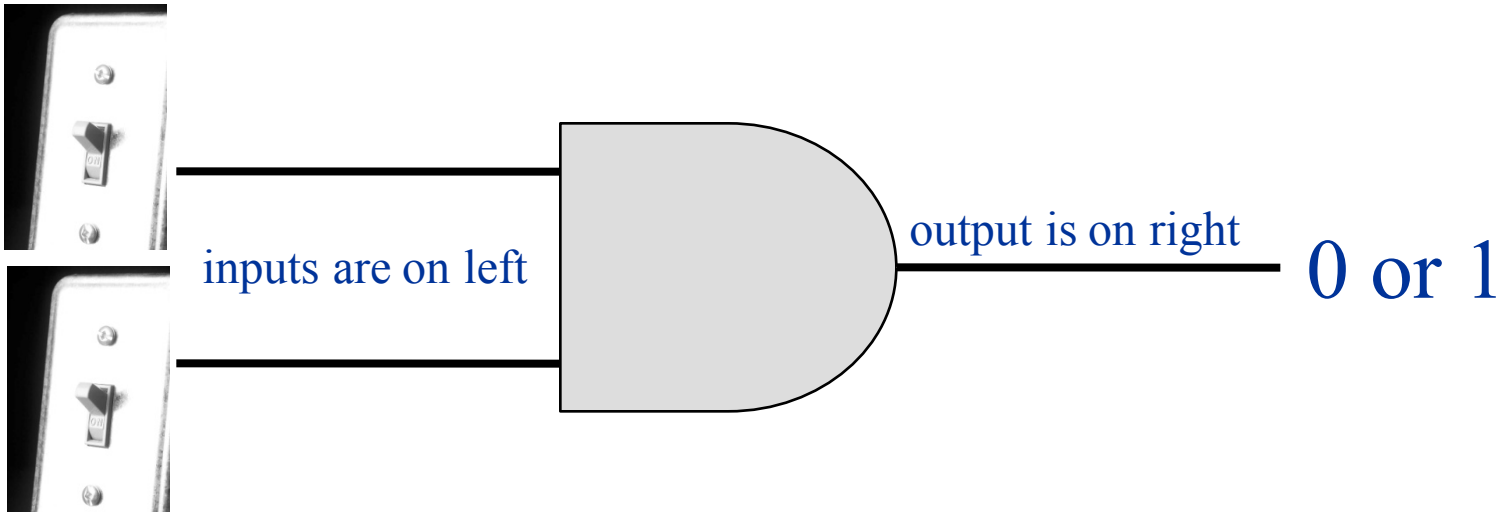
PROFESSIONALISM

Please turn off and put away electronics:

- cell phones
- pagers
- laptops
- tablets
- etc.

REVIEW

AND gate



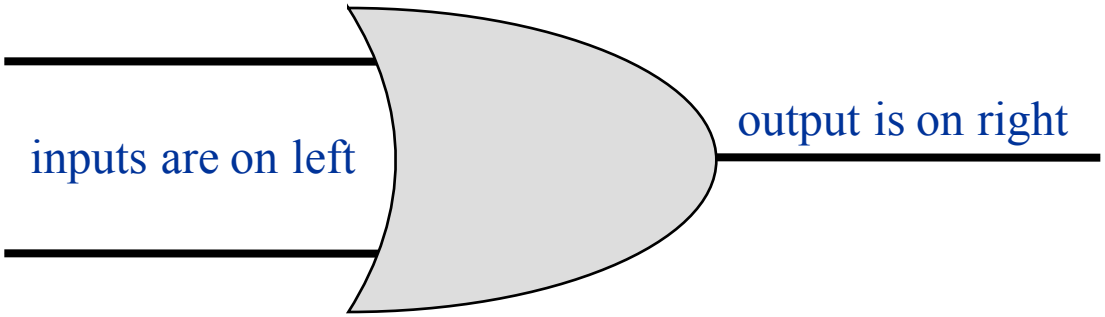
For which input values is output 1?

For which input values is output 0?

Truth table for AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR gate



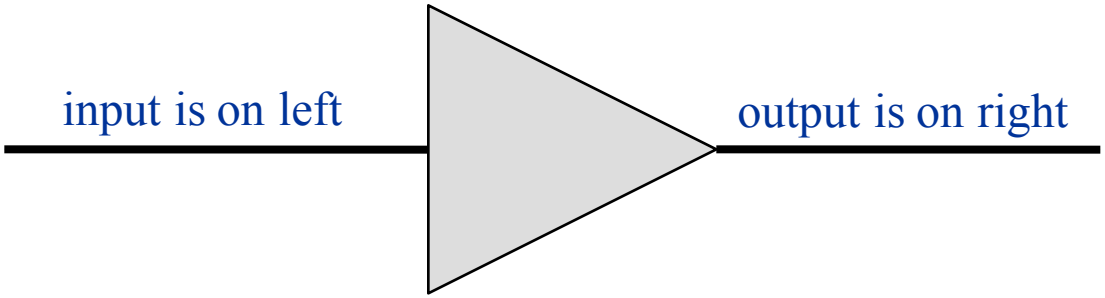
For which input values is output 1?

For which input values is output 0?

Truth table for OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate



For which input value is output 1?

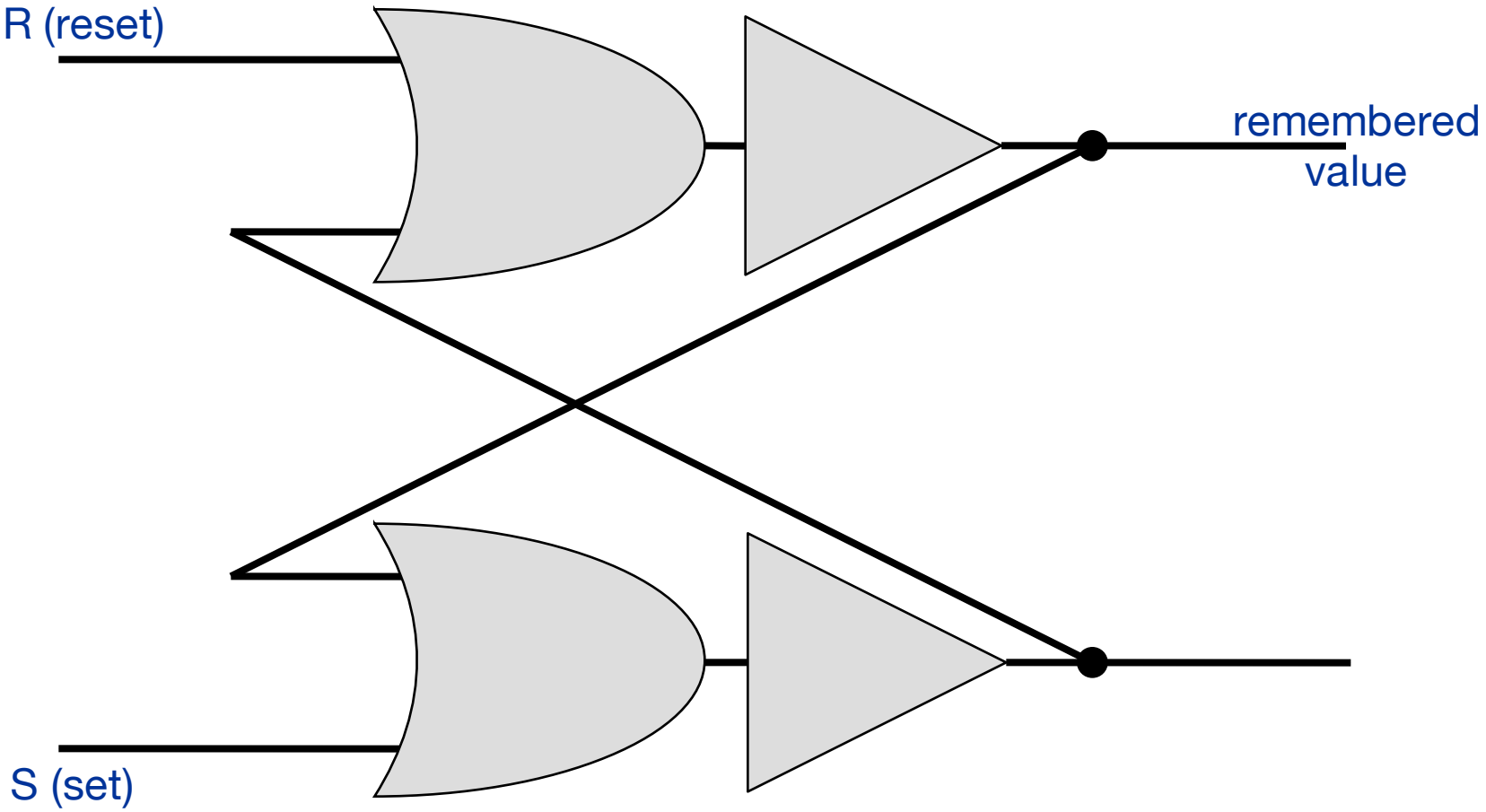
For which input value is output 0?

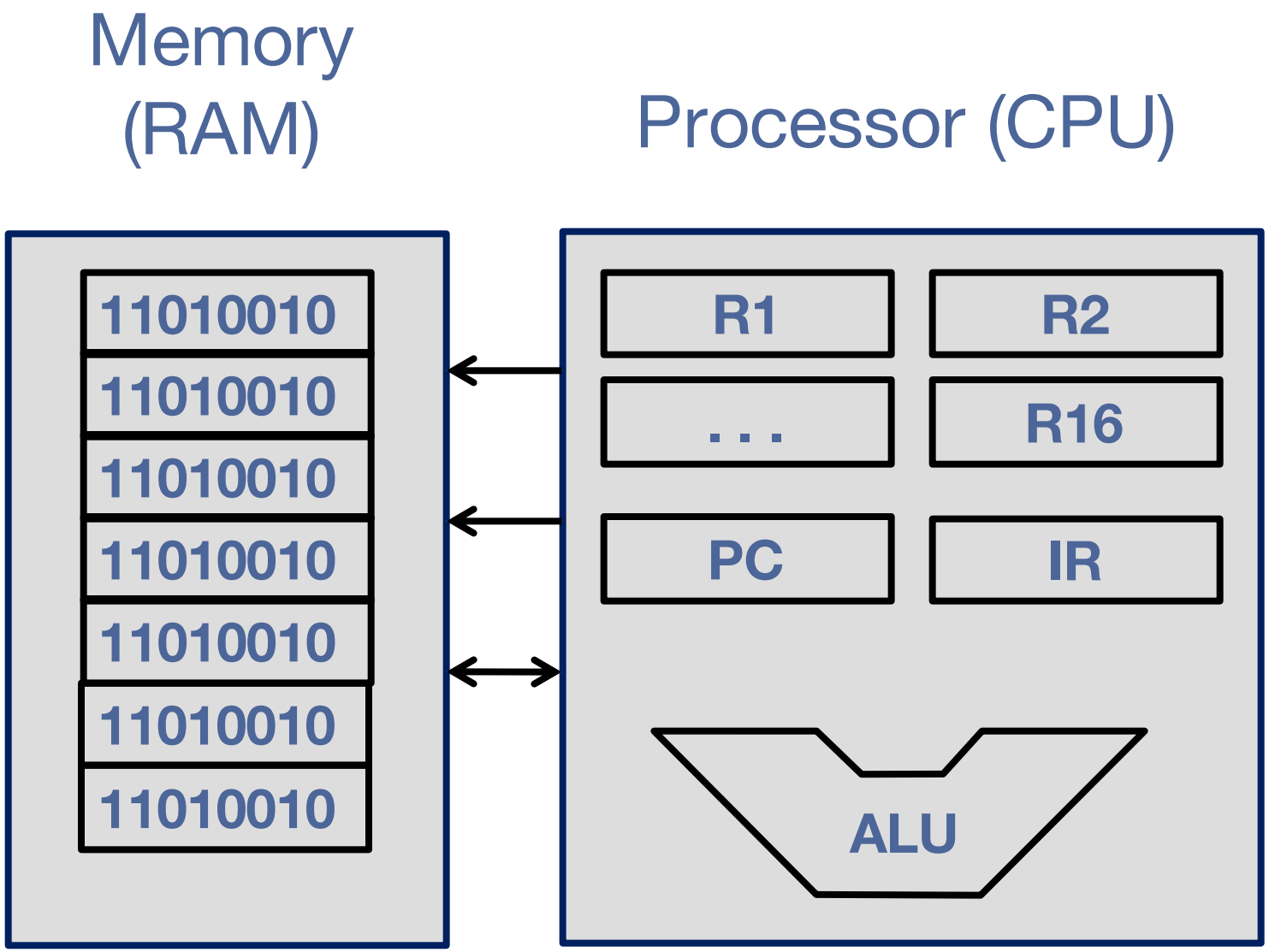
Truth table for NOT

Input	Output
0	1
1	0

MEMORY

Flip-flop (a bit of memory!)



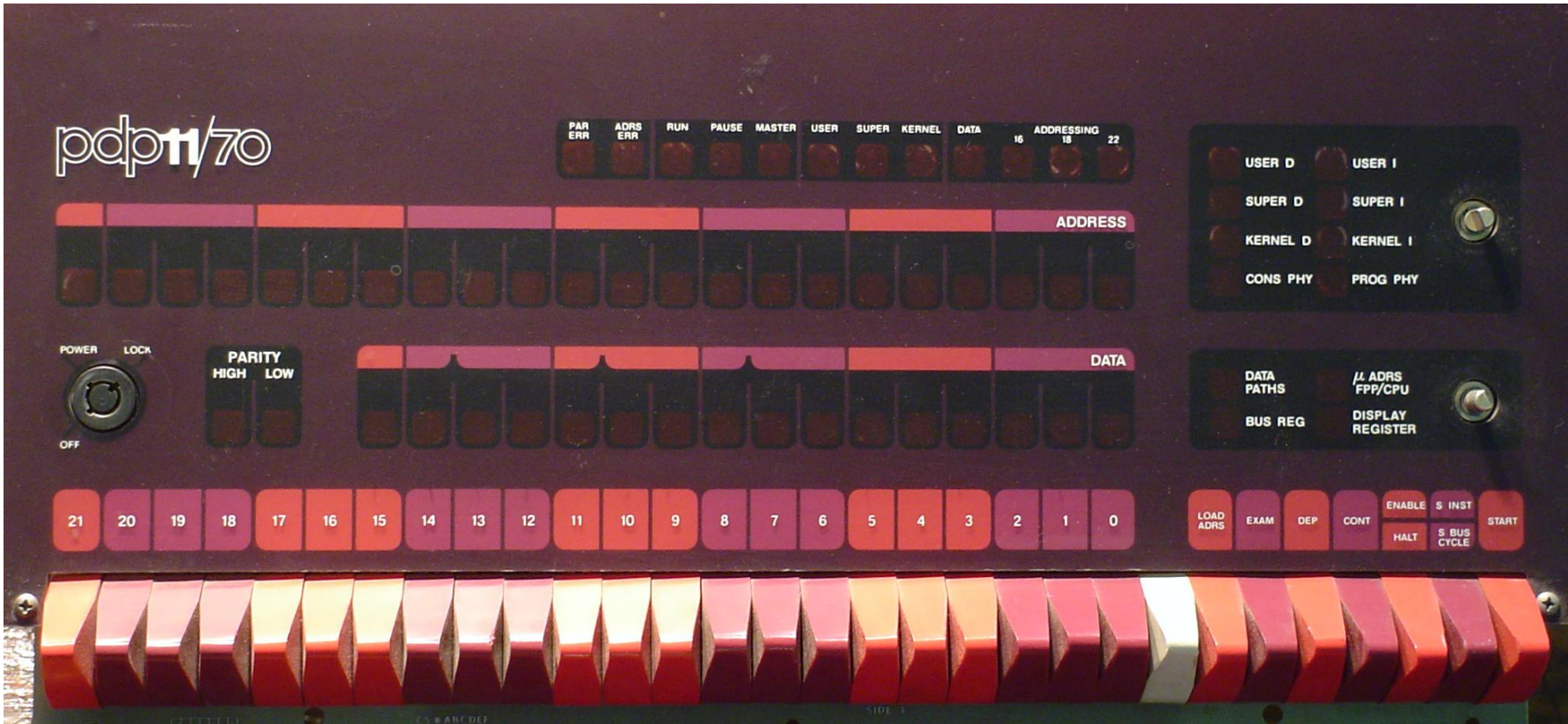

Memory
(RAM)
Processor (CPU)
11010010
11010010
11010010
11010010
11010010
11010010
11010010
R1
R2
. . .
R16
PC
IR
ALU

MOVING ON

INSTRUCTION DECODING

Example:
Digital Equipment Corp PDP 11

photo credit:
https://commons.wikimedia.org/wiki/File:Pdp-11-70-panel.jpg
by Dave Fischer



Encoding instructions (PDP 11)

0110000001000010

Encoding instructions (PDP 11)

0110000001000010	a sequence of 0s and 1s
0110 000001 000010	bit groupings

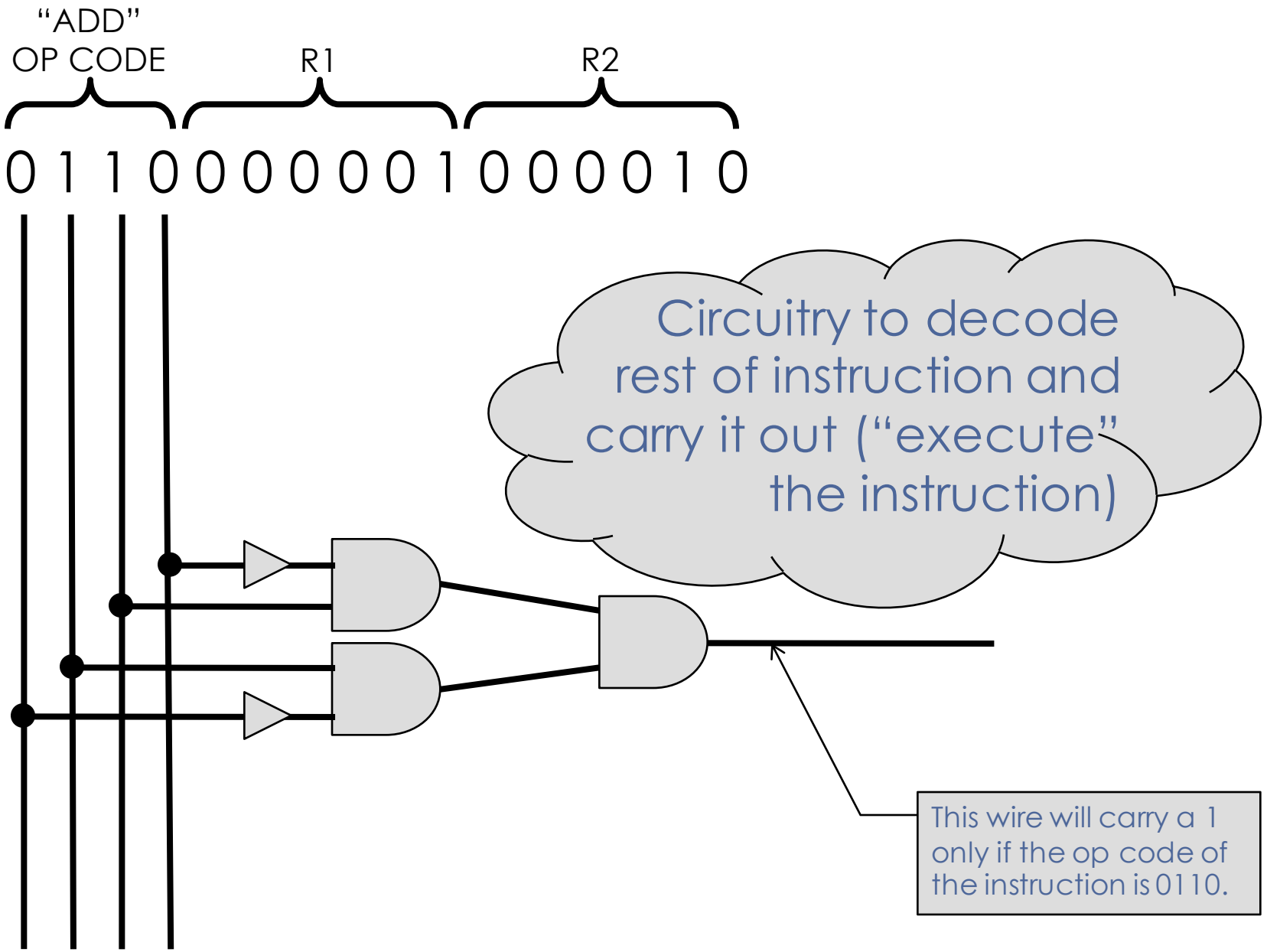
Encoding instructions (PDP 11)

0110000001000010			a sequence of 0s and 1s
0110	000001	000010	bit groupings
OpCode	Arg1	Arg2	general pattern

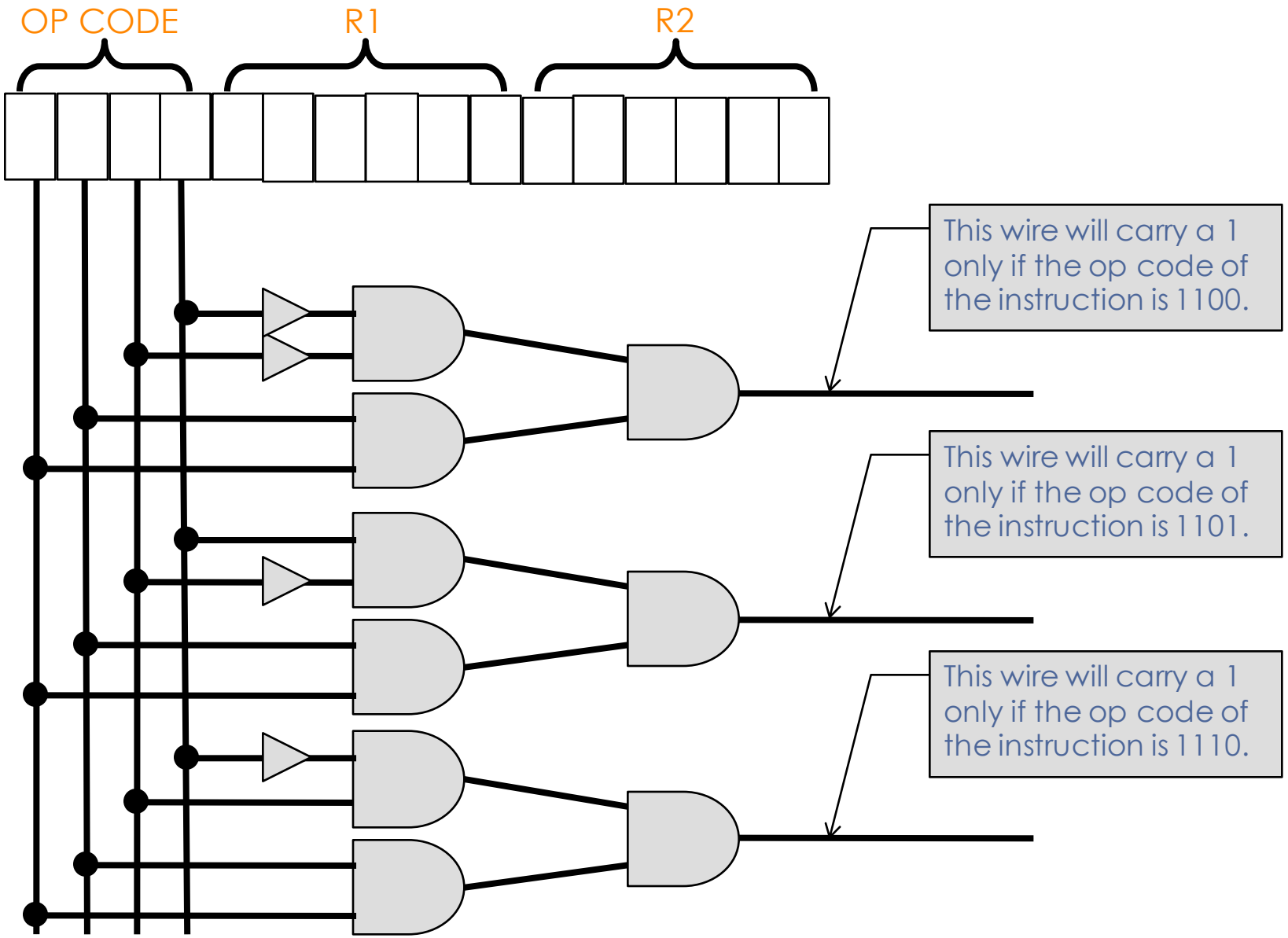
Encoding instructions (PDP 11)

0110000001000010			a sequence of 0s and 1s
0110	000001	000010	bit groupings
OpCode	Arg1	Arg2	general pattern
ADD	Reg1	Reg2	specific instruction

Instruction decoding

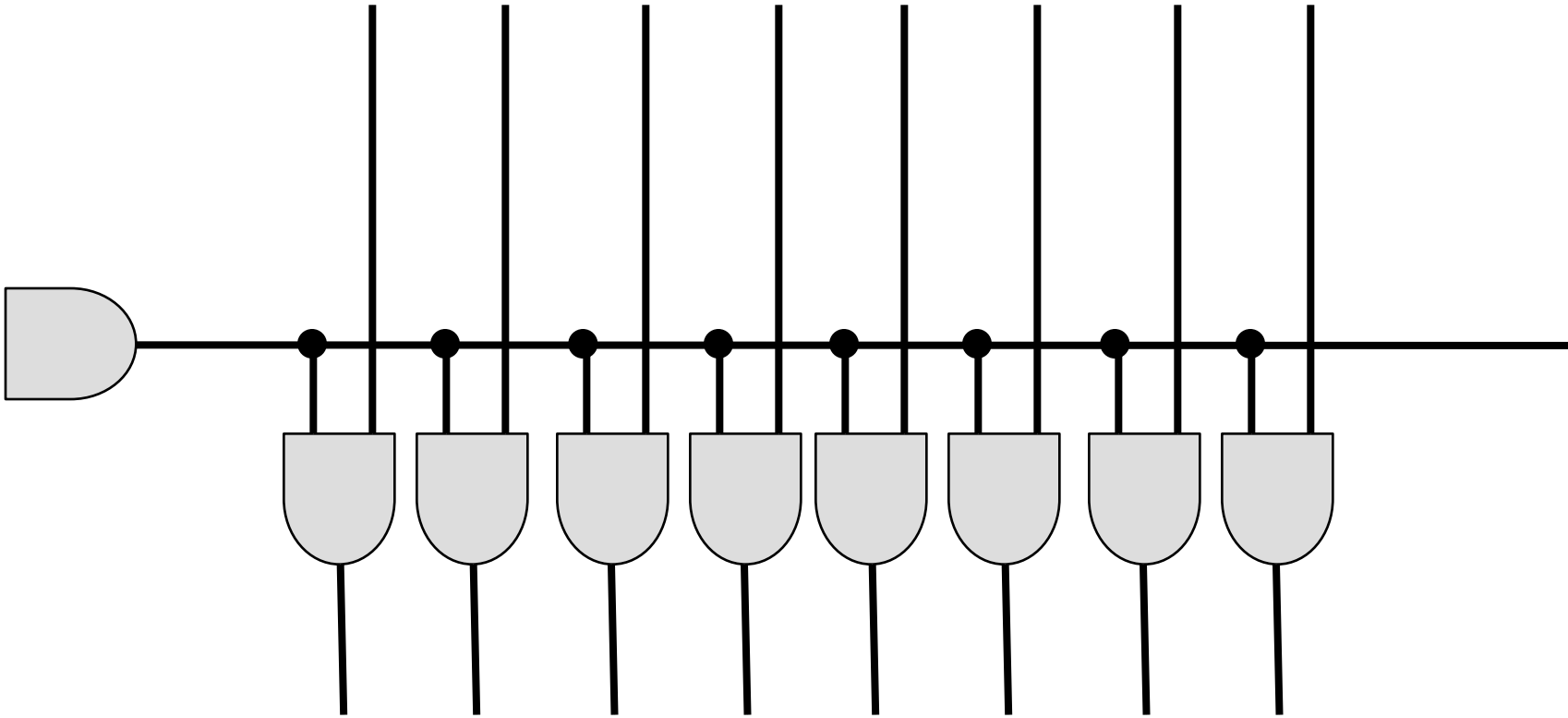


Instruction decoding



Controlling information flow

Data from a register



Data to ALU

FETCH
DECODE
EXECUTE

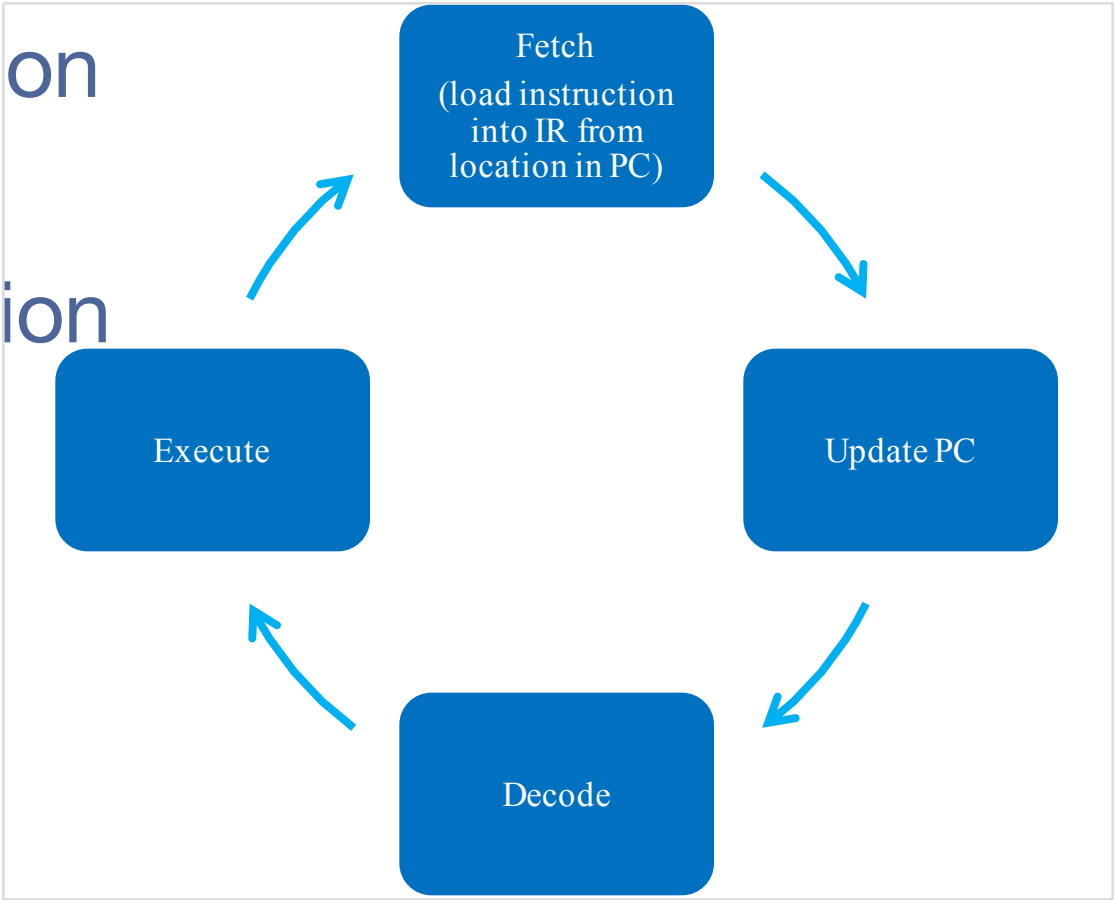
cycle

Fetch-Decode-Execute cycle

Fetch an instruction (& update PC)

Decode instruction

Execute instruction



1940s

Language levels

HARDWARE

1940s

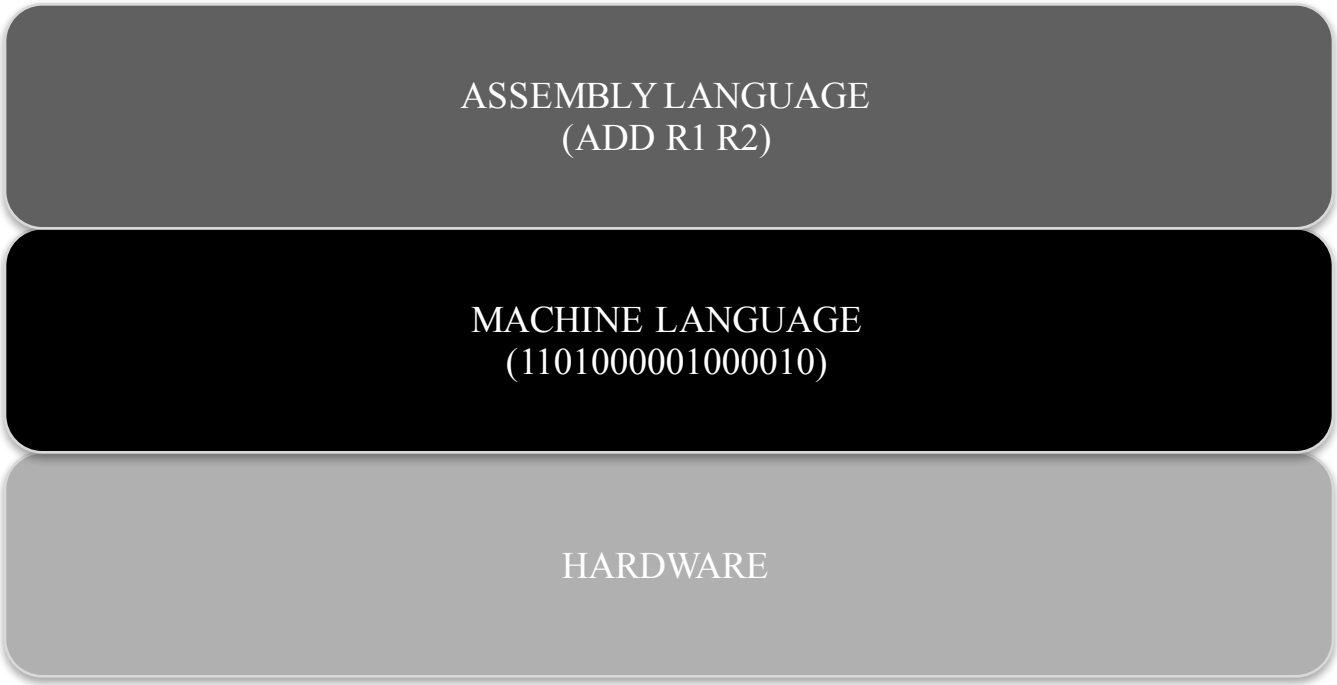
Language levels

MACHINE LANGUAGE
(1101000001000010)

HARDWARE

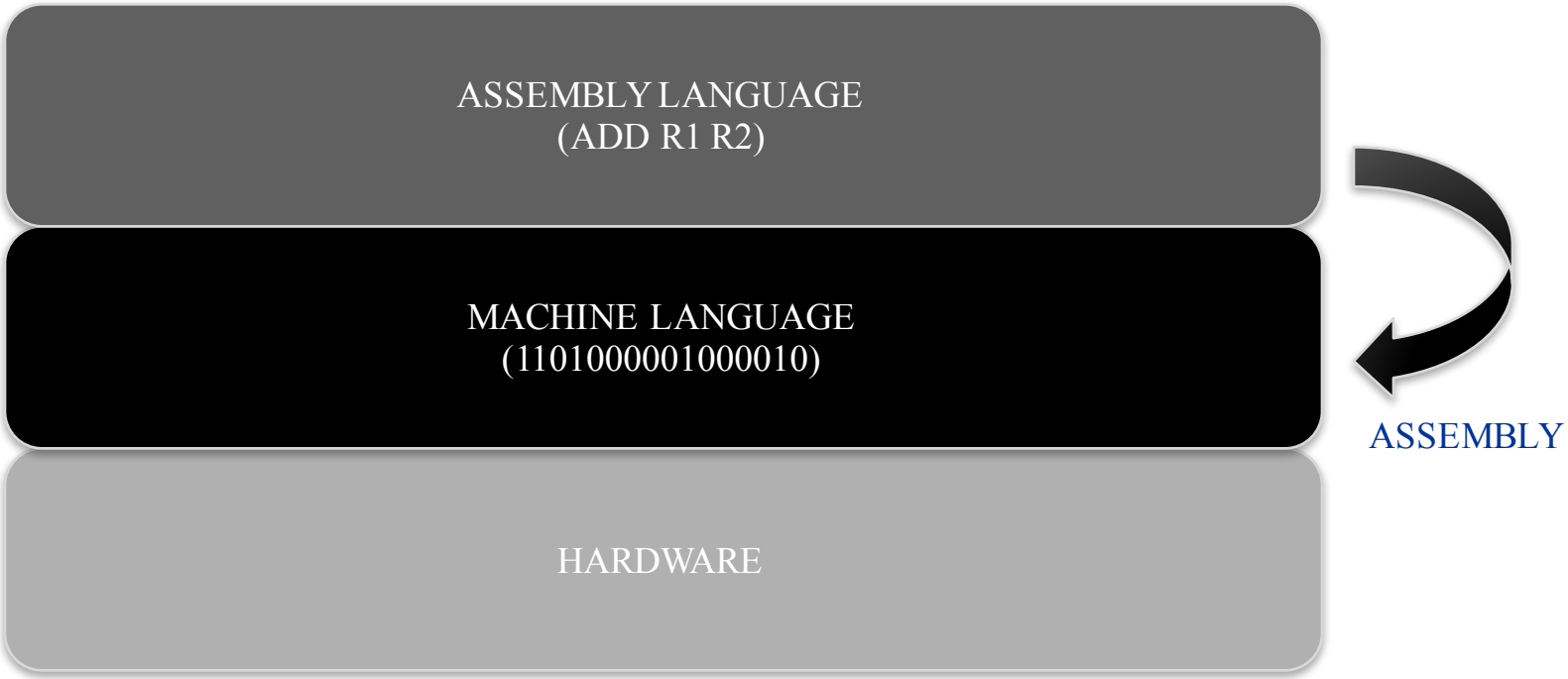
1940s

Language levels



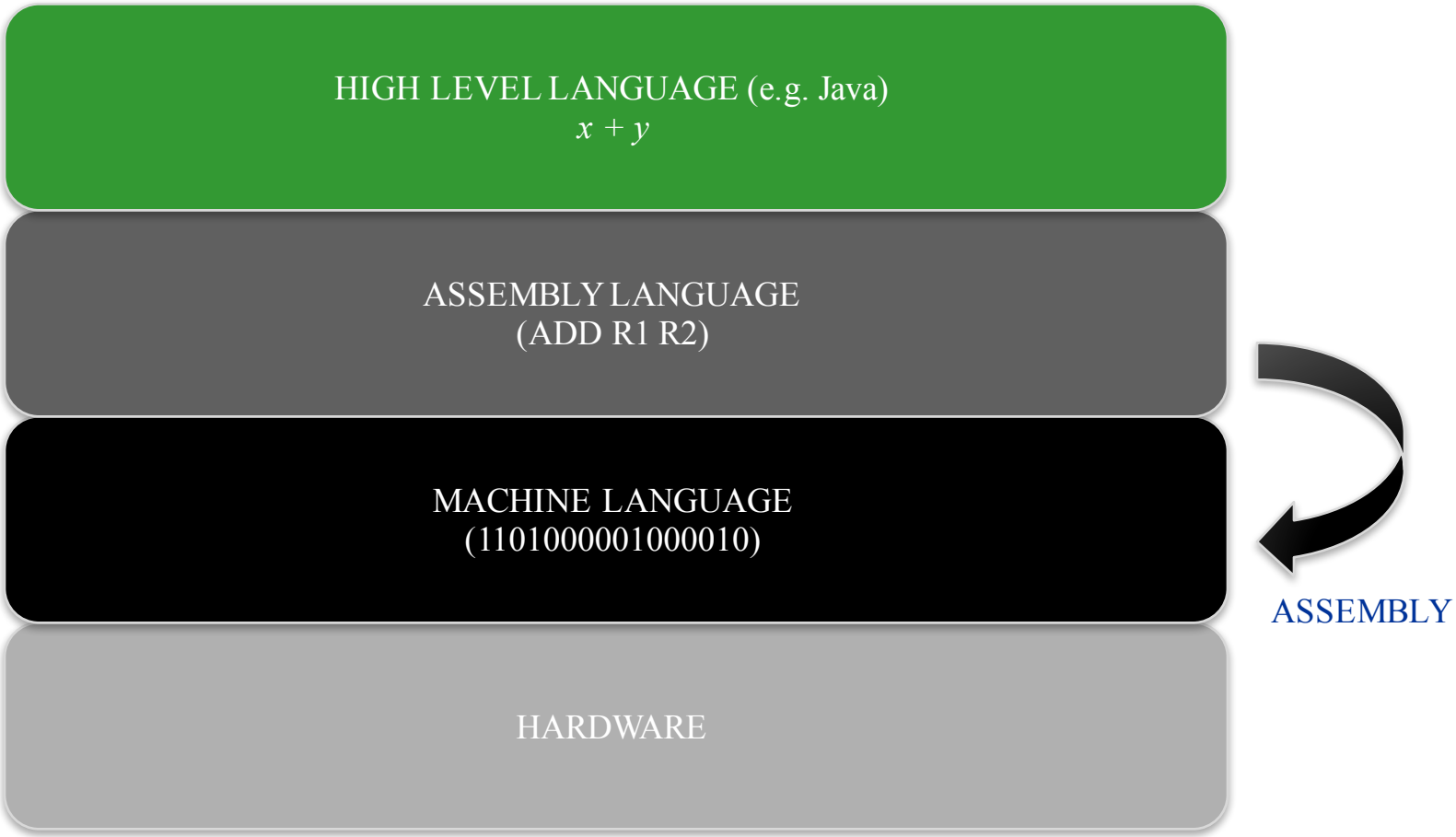
1940s

Language levels



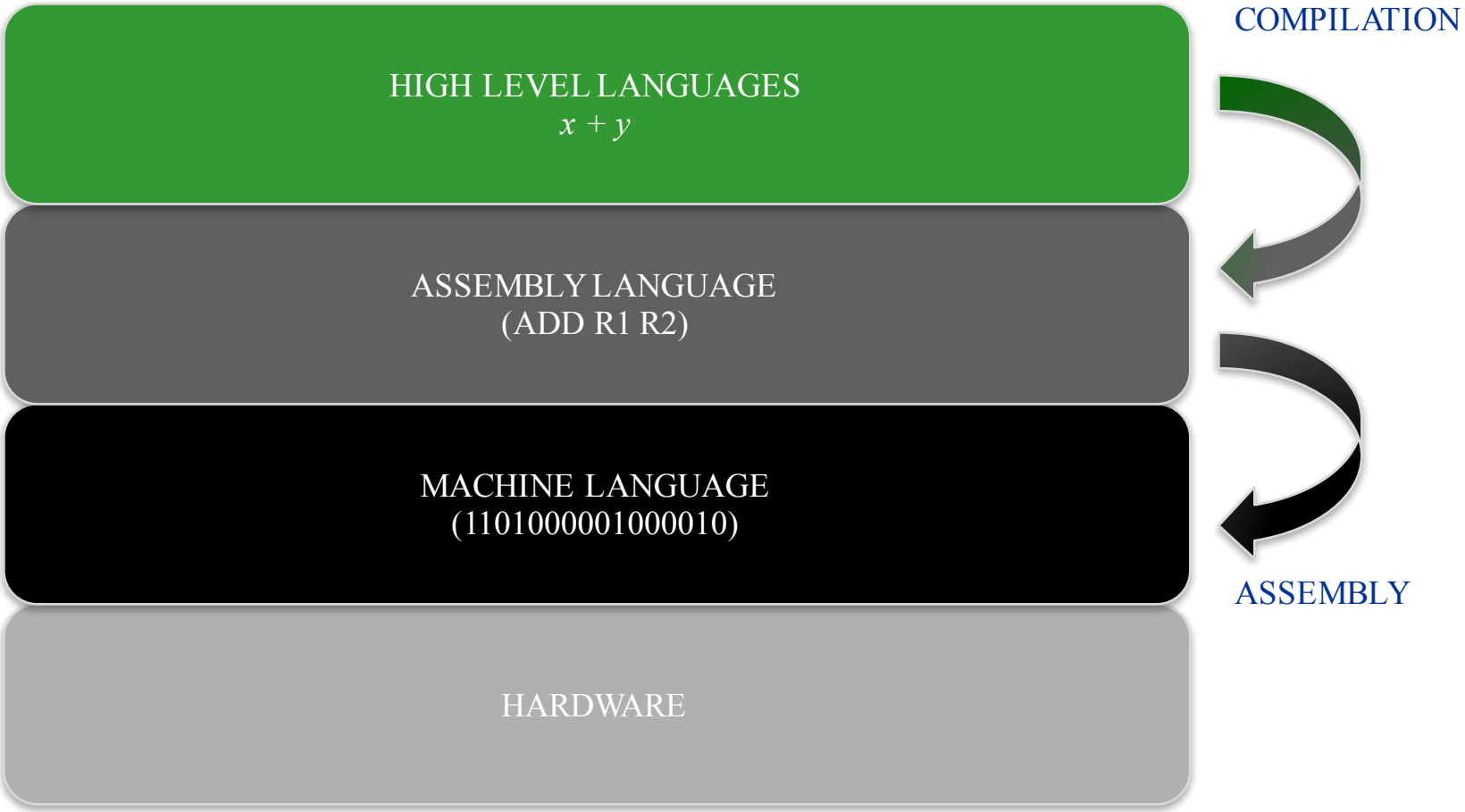
1950s

Language levels

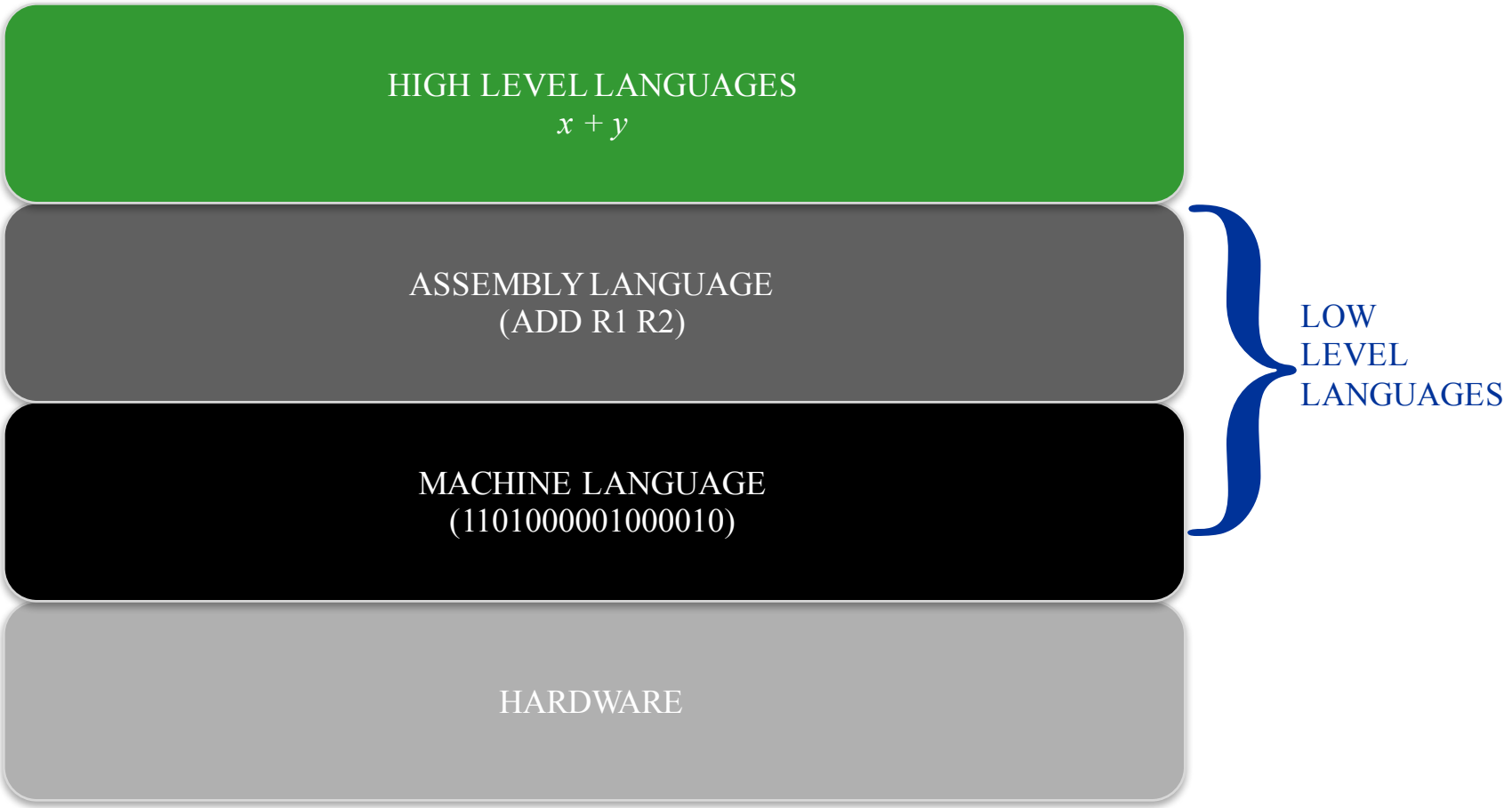


1950s

Language levels



Language levels



Java

We can write,

z = x + y

instead of something like this,

MOV (R3) R1

MOV (R4) R2

ADD R1 R2

MOV R2 (R5)

Others: C#, Erlang, Python, ML, Prolog, Lisp, etc.

NOTE

Differences between book's

&

lecture's presentation

of code

I pointed out that the text's early code examples are quite involved.

We will discuss the details in lecture over the next several days. The following slides give a sense of the order in which we will have our discussion.

DO NOT PANIC if you do not yet understand the book's code examples at this point – we don't expect you to.

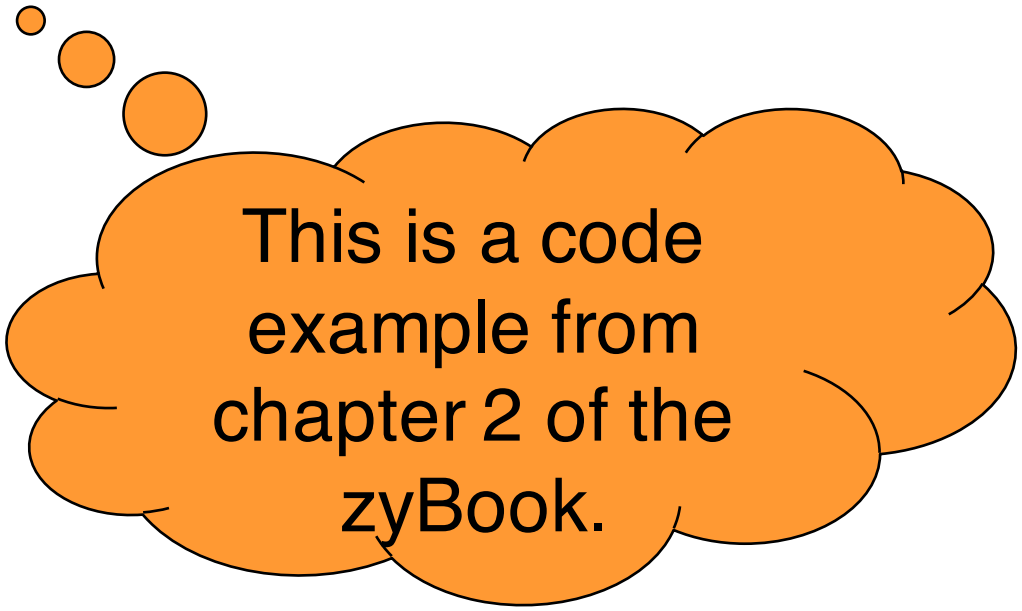
[zyBook] Figure 2.1.1: Using objects

```
public class AttendanceExample {
   public static void main(String[] args) {
      PeopleCounter attendeeCounter = new PeopleCounter();

      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();

      System.out.print("Attendee count: ");
      attendeeCounter.printCount();

      return;
   }
}
```



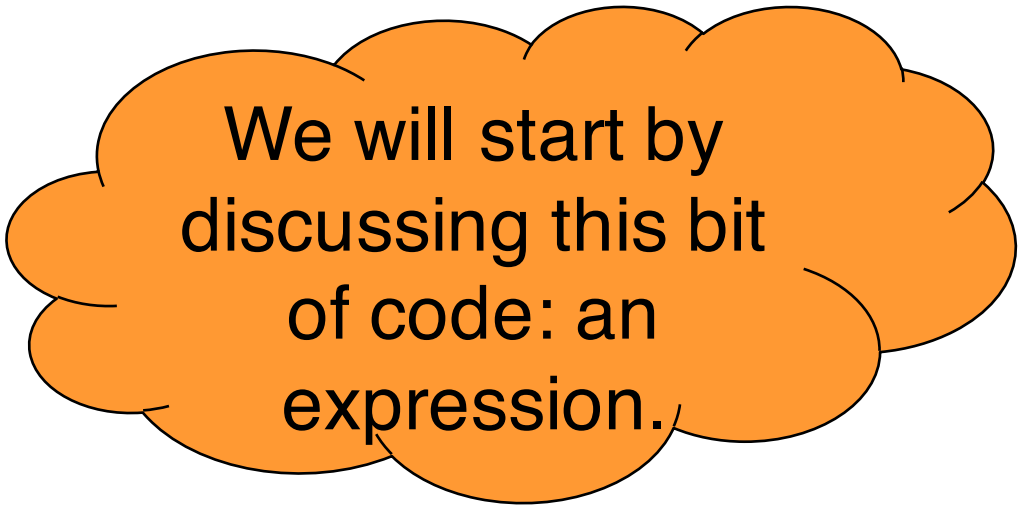
Expression

```
public class AttendanceExample {
   public static void main(String[] args) {
      PeopleCounter attendeeCounter = new PeopleCounter();

      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();

      System.out.print("Attendee count: ");
      attendeeCounter.printCount();

      return;
   }
}
```



Variable declaration

```
public class AttendanceExample {
   public static void main(String[] args) {
      PeopleCounter attendeeCounter = new PeopleCounter();

      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();

      System.out.print("Attendee count: ");
      attendeeCounter.printCount();

      return;
   }
}
```

Next we will discuss variable declarations.

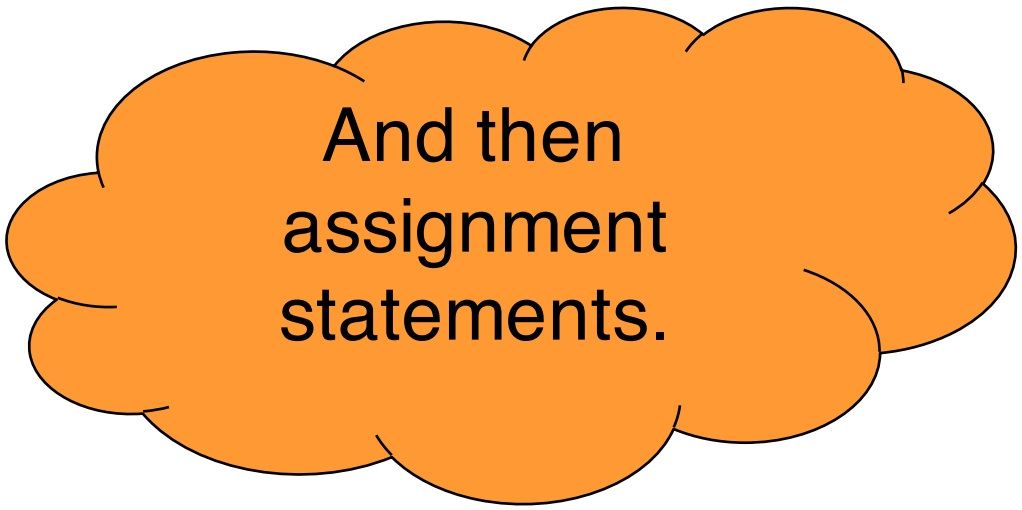
Assignment statement

```
public class AttendanceExample {
   public static void main(String[] args) {
      PeopleCounter attendeeCounter = new PeopleCounter();

      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();

      System.out.print("Attendee count: ");
      attendeeCounter.printCount();

      return;
   }
}
```



Method call

```
public class AttendanceExample {
   public static void main(String[] args) {
      PeopleCounter attendeeCounter = new PeopleCounter();

      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();

      System.out.print("Attendee count: ");
      attendeeCounter.printCount();

      return;
   }
}
```

Method calls will also be discussed, in their various forms.

Method definition

```
public class AttendanceExample {
    public static void main(String[] args) {
        PeopleCounter attendeeCounter = new PeopleCounter();

        attendeeCounter.incrementCount();
        attendeeCounter.incrementCount();
        attendeeCounter.incrementCount();

        System.out.print("Attendee count: ");
        attendeeCounter.printCount();

        return;
    }
}
```

As well as method definitions.

Class definition

```
public class AttendanceExample {
   public static void main(String[] args) {
      PeopleCounter attendeeCounter = new PeopleCounter();

      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();
      attendeeCounter.incrementCount();

      System.out.print("Attendee count: ");
      attendeeCounter.printCount();

      return;
   }
}
```