An Abstraction of the Machine

Aside:
Casting for primitives: converts a value of one type to a value of another

double d = 3.7;
int j = d; \times \text{compiler error} \quad \text{info. loss}
double x = j; \text{//OK}
int j = (\text{int}) d;

Casting for classes: the value is unchanged—just treating an object as a certain type.
CPU (Central Processing Unit)

- ALU
- Control Unit
- Registers

- Fetch → Decode → Execute Cycle

- Memory (RAM)

- Addresses

- STORE
- LOAD

- Bus

- Program counter (memory address of next instruction)
```
int i = 3;
int j = 5;
i = i + j;
```

```
STORE 3, 100
STORE 5, 104
LOAD 100, R1
LOAD 104, R2
ADD R1, R2, R3
STORE R3, 100
```
Circle

x, y, radius

program memory

the Heap

the Stack

where methods "run"

Memory

instructions