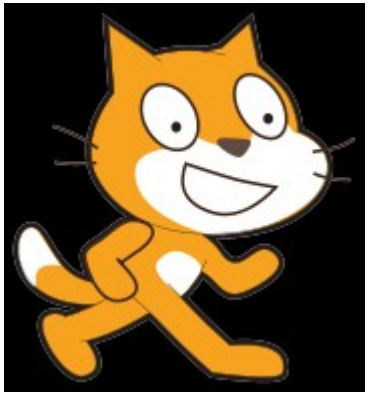
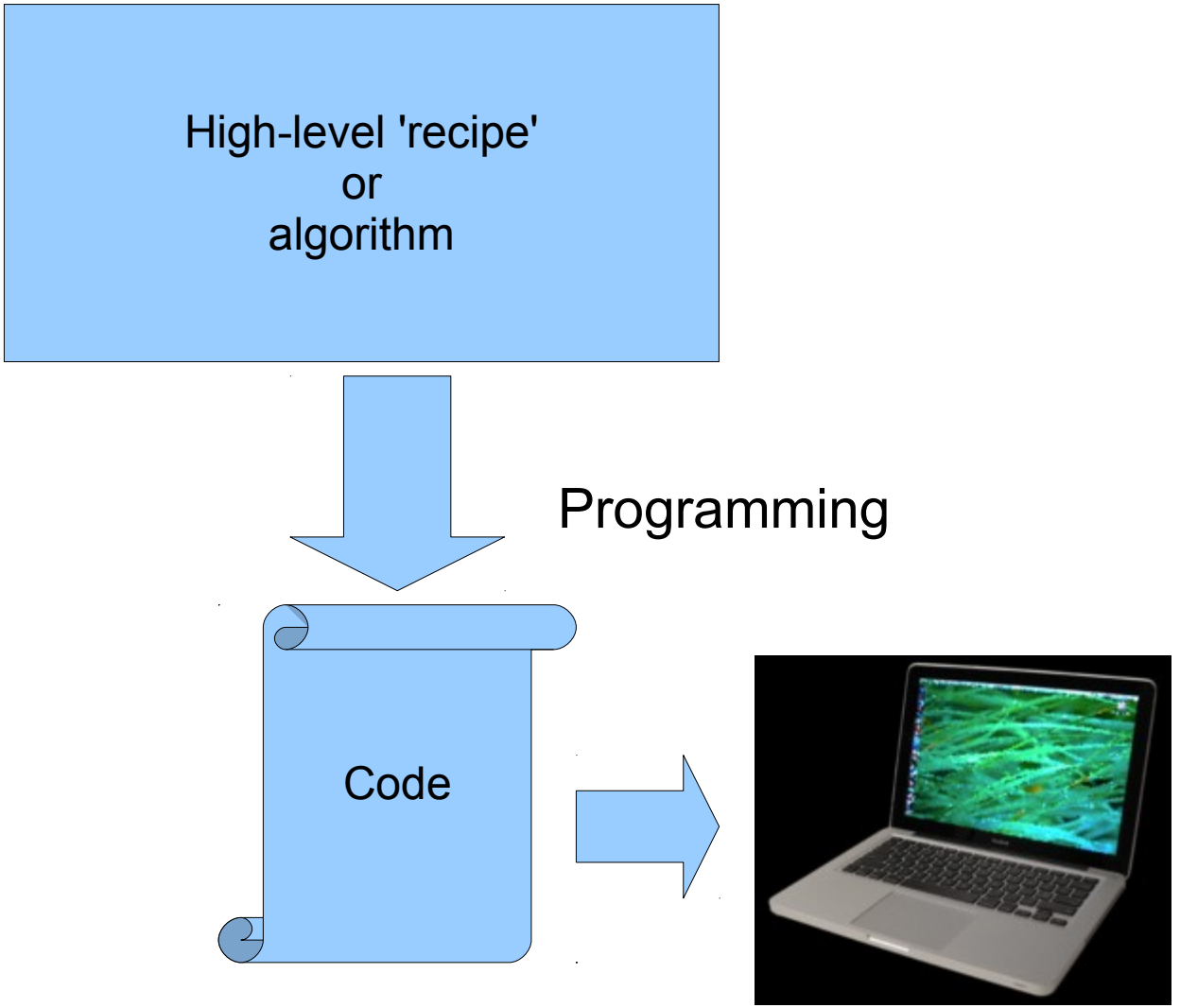
Introduction to programming

Programming



Programming in 131



Activity

- Socrative.com Room: 187417

- What is three plus four times five?

- What is the difference between eleven and 6?

Programming languages

- First generation: 1101010000000000

Programming languages

- Second generation (Assembly)

```
addcc %r3,%r5,%r3
```

- Third generation (programmer-friendly)

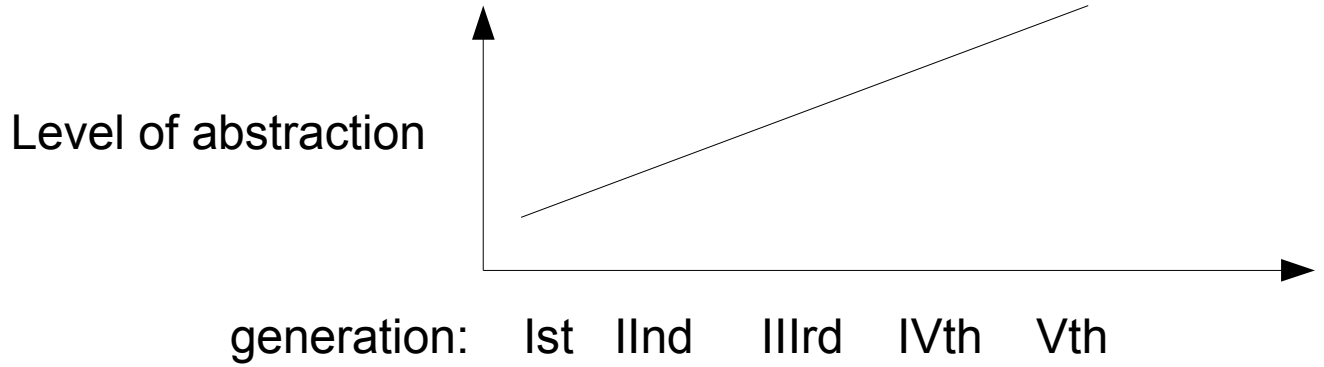
```
x = Math.pow(2, 10);
```

- Fourth generation (application-specific: SQL, R, Matlab, ...)

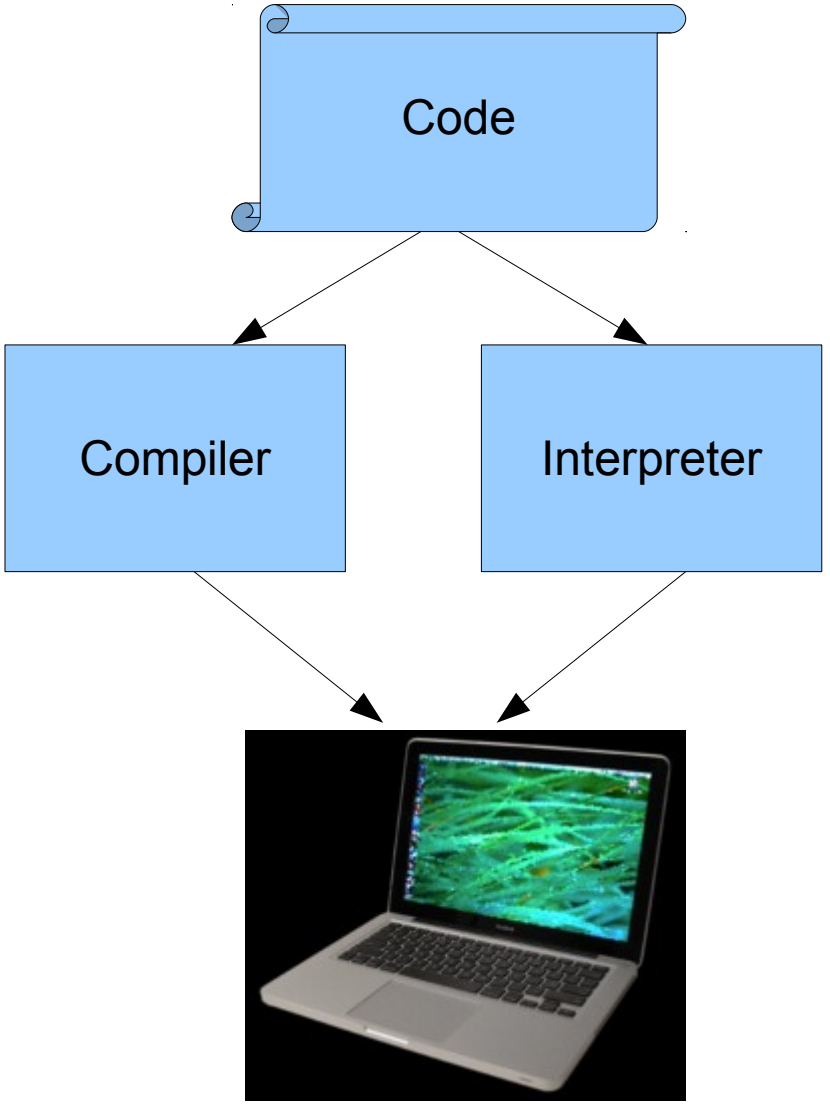
```
SELECT * FROM Book
WHERE title LIKE "%Java%"
    AND price < 20;
```

- Fifth generation (constraint/logic programming)

```
solve(A) :- clause(A,B), solve(B)
```



Programming tools



eclipse

Errors....

```
package blah;

public class BlahBlah {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        if (args <> 10 then ){
            printString(args)
        }
    }
```

Syntactic

Compile-time

```
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
The operator > is undefined for the argument type(s) String[], int
Syntax error on token(s), misplaced construct(s)
Syntax error on token "then", delete this token
The method printString(String[]) is undefined for the type BlahBlah
Syntax error, insert ";" to complete BlockStatements

at blah.BlahBlah.main(BlahBlah.java:7)
```

Semantic

System.out.println("The square root of 5 is:" + 5 * 5);

(most) programming languages need:

- Variables – names for information in memory
  - int x, y;
- Assignment operator
  - x = x + 5 (obviously false?)
  - assign to memory location x, its prior value incremented by 5
- Mathematical/logical operators
  - x * 5; y-3; ...
  - x < 20; y >= x, ...
- Conditional statements
  - if something happens do something
  - repeat operation while some condition holds
- Iterative statements
  - repeat operation....

(most) programming languages need

- Ways to group code that may be reused
  - functions/subroutines
- Ways to structure information
  - struct person {
    int SSN;
    String name;
    String address;
    }

- In object oriented programming – objects encode both data and methods/operations:
  - class Person {
    int SSN;
    Money spend()
    }