

# Abstraction In Programming Languages

# Abstraction - Definition

- In computer science, “abstraction” reduces details so that one can focus on concepts .
- Abstraction can apply to control or to data:
  - Control Abstraction involves, for example, the use of subprograms to simplify control flow.
  - Data Abstraction involves assigning meaning to data bits or bytes within and across programs.

Bytes  Integers, Rationals, Imaginaries

Bytes  Characters  Strings,

The requirement that a programming language provide suitable abstractions is called the

**abstraction principle**

# Cobol (1959)

```
* ** SET DAYS IN FEBRUARY **  
*  
MOVE 28 TO DW-DAYS-IN-MONTH(2).  
DIVIDE 400 INTO DW-WORK-YYYY GIVING DW-WORK1  
REMAINDER DW-WORK2.  
IF (DW-WORK2 = 0)  
MOVE 29 TO DW-DAYS-IN-MONTH(2)  
ELSE  
DIVIDE 100 INTO DW-WORK-YYYY GIVING DW-WORK1  
REMAINDER DW-WORK2  
IF (DW-WORK2 NOT = 0)  
DIVIDE 4 INTO DW-WORK-YYYY GIVING DW-WORK1  
REMAINDER DW-WORK2  
IF (DW-WORK2 = 0)  
MOVE 29 TO DW-DAYS-IN-MONTH(2).  
*  
IF (DW-WORK-DD < 01)  
OR  
(DW-WORK-DD > DW-DAYS-IN-MONTH(DW-WORK-MM))  
GO TO 001000-EXIT.
```

# APL – A Programming Language (~1965)

The Sieve of Eratosthenes -The Program:

```
(~R€R◦.xR)/R←1↓120  
2 3 5 7 11 13 17 19
```

The Keyboard:



$R \leftarrow 1 \downarrow 113$

R

2 3 4 5 6 7 8 9 10 11 12 13

$S \diamond S \leftarrow R \circ . \times R$

4	6	8	10	12	14	16	18	20	22	24	26
6	9	12	15	18	21	24	27	30	33	36	39
8	12	16	20	24	28	32	36	40	44	48	52
10	15	20	25	30	35	40	45	50	55	60	65
12	18	24	30	36	42	48	54	60	66	72	78
14	21	28	35	42	49	56	63	70	77	84	91
16	24	32	40	48	56	64	72	80	88	96	104
18	27	36	45	54	63	72	81	90	99	108	117
20	30	40	50	60	70	80	90	100	110	120	130
22	33	44	55	66	77	88	99	110	121	132	143
24	36	48	60	72	84	96	108	120	132	144	156
26	39	52	65	78	91	104	117	130	143	156	169

$\sim R \in S$

1 1 0 1 0 1 0 0 0 1 0 1

$(\sim R \in S) / R$

2 3 5 7 11 13