CHARACTER I/O

Reading and writing files one character at a time

Delroy A. Brinkerhoff

STREAMS AND FUNCTIONS

- `ifstream in("input.txt");`
- `ofstream out("output.txt);`

- `in.open("input.txt");`
- `out.open("output.txt");`

- Text and binary

- `ostream& put(char c);`

- `int get();`
- `istream& get(char& c);`
  - `operator bool()`

ONE-READ PATTERNS

```
int c;
while ((c = in.get()) != EOF)
{
    // process c
    out.put(c);
}
```

```
char c;
while (input.get(c))
{
    // process c
    out.put(c);
}
```

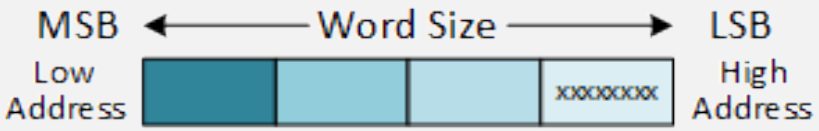
- The char type is an integer, and programs can represent characters with integers of various lengths
- Streams maintain their status with state flags
- Functions set the state flags on failed I/O operations

TWO-READ PATTERNS

```
int c = in.get();
while (! in.eof())
{
    // process the data
    out.put(c);
    c = in.get();
}
```

```
int c = in.get();
while (in)
{
    // process the data
    out.put(c);
    c = in.get();
}
```

COMPUTER DATA STORAGE



$$d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0$$

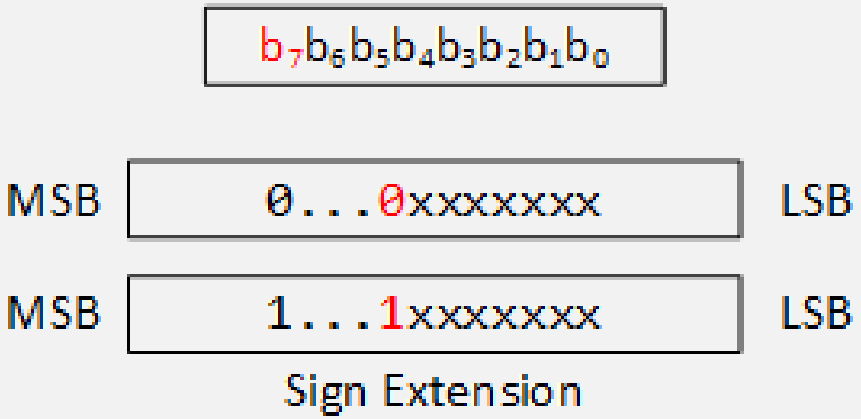
$$345 = (3)10^2 + (4)10^1 + (5)10^0$$

MSB $b_7 2^7 + b_6 2^6 + \ldots + b_1 2^1 + b_0 2^0$ LSB

MSB = Most Significant Byte / Bit
LSB = Least Significant Byte / Bit

- Integers vary in length
  - char is 8 bits or one byte
  - short is typically 16 bits
  - int and long typically 32 or 64 bits
- `put(char)`
  - Writes the least significant byte (LSB)
  - Discards higher-order bits

CHARACTER DATA INPUT



M/LSB = Most / Least Significant Bit

- `put(char)`
- `get(char)`: retains sign
  - `(int) char`
  - Sign extension copies the MSB to fill the higher-order bits
- `int get()`
  - Unsigned
  - The MSB is part of the magnitude

THE get FUNCTIONS

Decimal	Binary	get(char)	int get()
-128	1000 0000	-128	128
-127	1000 0001	-127	129
-1	1111 1111	-1	255
0	0000 0000	0	0
1	0000 0001	1	1
127	0111 1111	127	127

```
while((c = in.get()) != EOF)
   (c > 127) ? (c – 256) : c
```