

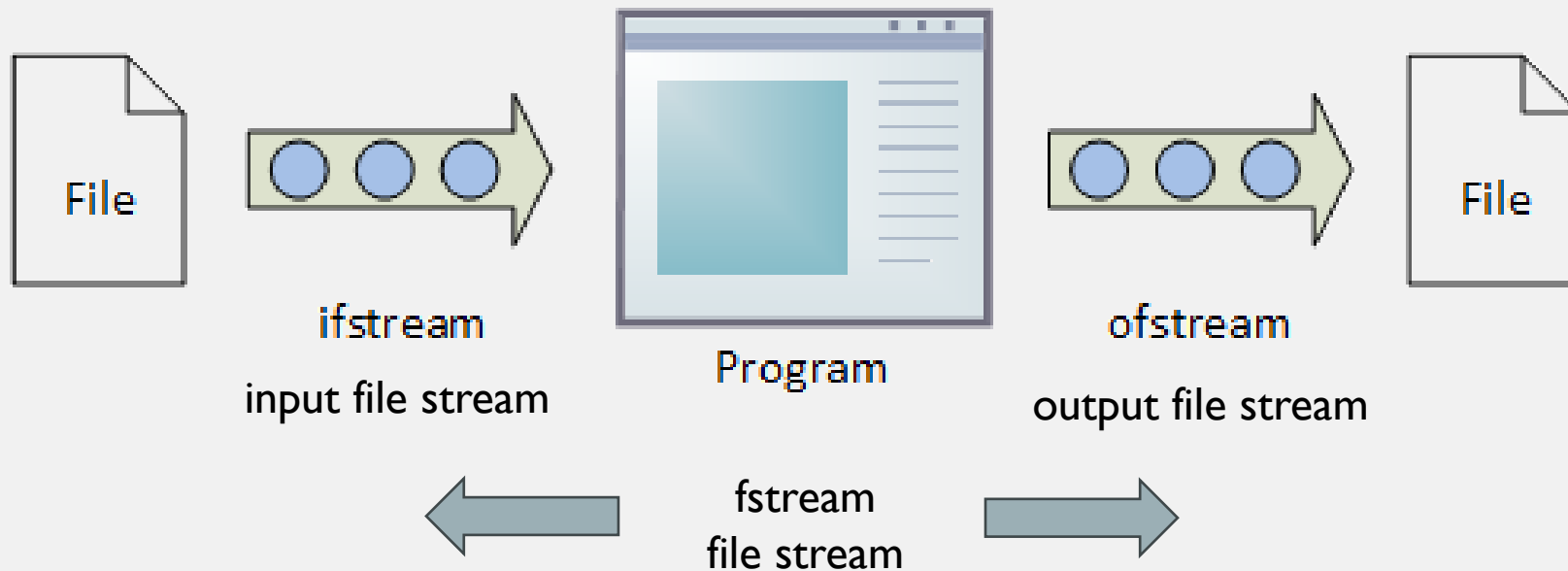


OPENING AND CLOSING FILES

C++ programs open files to use them

Open files consume system resources and must be closed

C++ PROGRAMS ACCESS FILES WITH STREAM OBJECTS





CONFIGURING STREAMS

- Global processing
 - Mapping between `\r\n` and `\n`
- How do programmers configure streams?
- How do streams “remember” their configuration?
- Flags are variables used to record a program's state or condition
- A bit vector is a set of bits, with individual bits treated as switches or flags
- A bitmap is a constant pattern of bits denoting possible stream settings

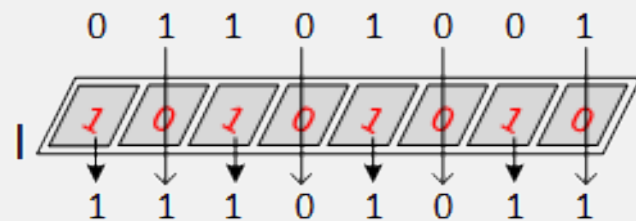


Flag	Meaning	Binary
ios::in	Open for input (reading)	0000000000000000000000000000000001
ios::out	Open for output (writing)	0000000000000000000000000000000010
ios::ate	Open at end	00000000000000000000000000000000100
ios::app	Append data to the existing	000000000000000000000000000000001000
ios::trunc	Truncate (delete) existing data	0000000000000000000000000000000010000
ios::binary	Open in binary mode	00000000000000000000000000000000100000

BITWISE-AND & BITWISE-OR: MANAGING BIT VECTORS

a	b	a b
0	0	0
0	1	1
1	0	1
1	1	1

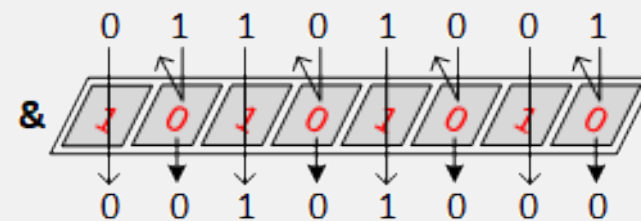
a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1



data = 0x69

mask = 0xaa

result = 0xeb



data = 0x69

mask = 0xaa

result = 0x28



STREAM STATE FLAGS

Flag	Function	Description
goodbit	good()	True (1): the file is healthy and ready for I/O
eofbit	eof()	True (1): the stream has read the end of the file
failbit	fail()	True (1): stream is at the end of file or encountered a “soft,” recoverable error
badbit	bad()	True (1): stream is corrupt and unable to perform I/O
	clear()	Sets goodbit to 1, zeros (resets) the other flags



OPENING FILES

- `openmode mode = ios::in | ios::out | ios::app | ios::binary;`
- `= 00000000000000000000000000000000101011`

- `ifstream(file_name, openmode mode = ios::in);`
- `ifstream();`
- `open(file_name, openmode mode = ios::in);`

- `ofstream(file_name, openmode mode = ios::out);`
- `ofstream();`
- `open(file_name, openmode mode = ios::out);`



VALIDATING AN OPEN FILE

```
ifstream file(file_name);
```

```
if (! file.good())  
{  
    // open failed  
}
```

```
// process the file
```

```
ifstream file(file_name);
```

```
if (file.good())  
{  
    // process the file  
}  
else  
{  
    // open failed  
}
```




CLOSING FILES

- ```
ifstream input(file_name);
// use the file
// "input" closed by destructor
```
- ```
while (....)  
{  
    ifstream input(file_name);  
    // use the file  
    // "input" closed by destructor  
}
```
- ```
ifstream input;
while (....)
{
 input.open(file_name);
 // use the file
 input.close();
}
```