

## READING AND WRITING FILES

### 2.4 Reading and Writing

#### read

To transfer data from a file to program variable(s).

#### write

To transfer data to a file from program variable(s) or constant(s).

- The read and write operations are performed on the logical file with calls to library functions.
- For read, one or more variables must be supplied to the read function, to receive the data from the file.
- For write, one or more values (as variables or constants) must be supplied to the write function, to provide the data for the file.
- For unformatted transfers, the amount of data to be transferred must also be supplied.

#### 2.4.1 Read and Write Functions

##### Reading

- The C++ *read* function is used to read data from a file for handle level access.
- The read function must be supplied with (as an arguments):
  - The source file to read from
  - The address of the memory block into which the data will be stored
  - The number of bytes to be read(byte count)
- The value returned by the *read* function is the number of bytes read.

Read function:

Prototypes:

```
int read (int Handle, void * Buffer, unsigned Length);
```

Example:

```
read (Input, &C, 1);
```

##### Writing

- The C++ *write* function is used to write data to a file for handle level access.
- The handle write function must be supplied with (as an arguments):
  - The logical file name used for sending data
  - The address of the memory block from which the data will be written
  - The number of bytes to be write
- The value returned by the *write* function is the number of bytes written.

Write function:

Prototypes:

```
int write (int Handle, void * Buffer, unsigned Length);
```

Example:

```
write (Output, &C, 1);
```

#### 2.4.2 Files with C Streams and C++ Stream Classes

- For FILE level access, the logical file is declared as a pointer to a FILE (FILE \*)
- The FILE structure is defined in the stdio.h header file.

#### Opening

The C++ *fopen* function is used to open a file for FILE level access.

- The FILE fopen function must be supplied with (as arguments):
  - The name of the physical file
  - The access mode
- The value returned by the *fopen* is a pointer to an open FILE, and is assigned to the file variable.

fopen function:

```
file = fopen (filename, type);
```

Prototypes:

```
FILE * fopen (const char* Filename, char * Access);
```

Example:

```
FILE * Input;
```

```
Input = fopen ("Daily.txt", "r");
```

The access mode should be one of the following strings:

**r**

Open for reading (existing file only) in text mode

**r+**

Open for update (existing file only)

**w**

Open (or create) for writing (and delete any previous data)

**w+**

Open (or create) for update (and delete any previous data)

**a**

Open (or create) for append with file pointer at current EOF (and keep any previous data) in text mode

**a+**

Open (or create) for append update (and keep any previous data)