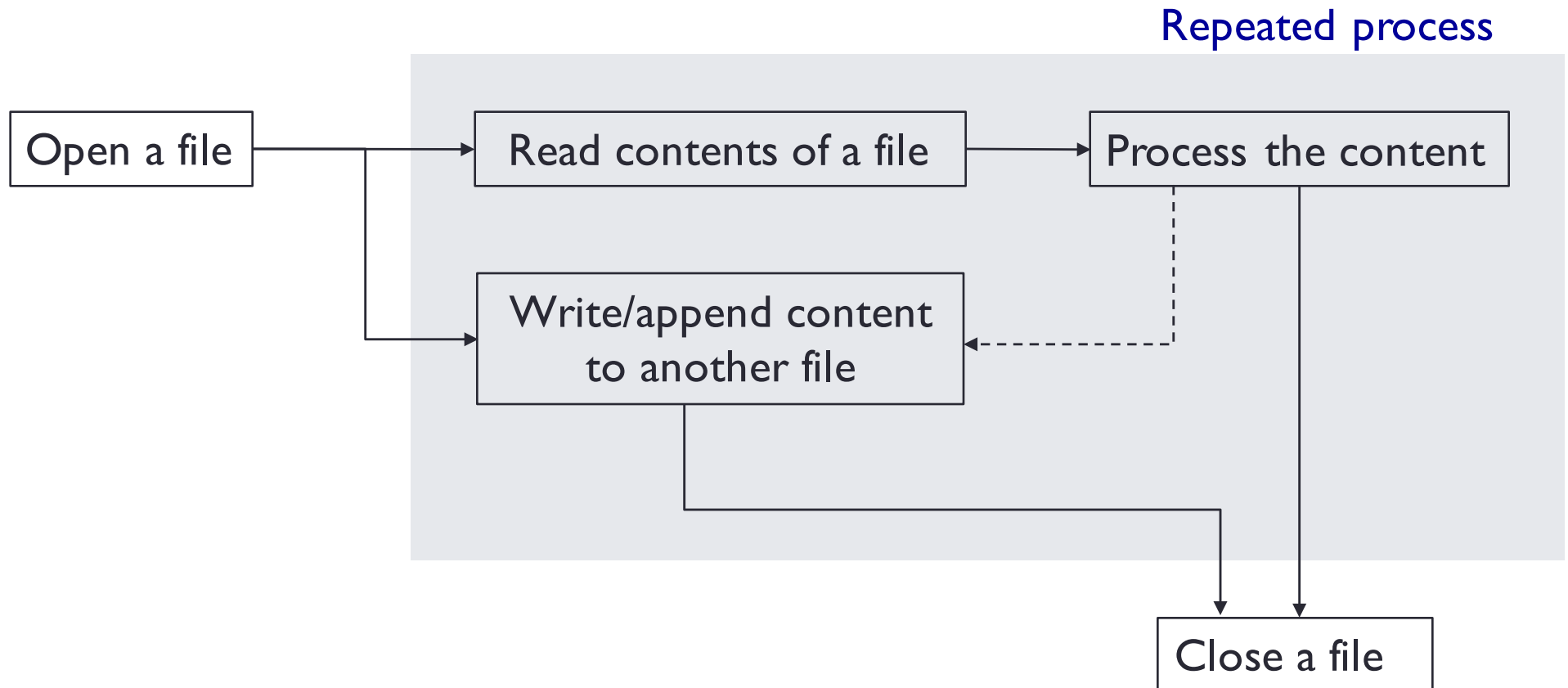


# **File Processing**

---

**CS 1111**  
**Introduction to Programming**  
**Spring 2019**

# Overview: File Processing



- Open before read or write
- Open before close
- Read or write before close
- Be **extra careful** when **writing** or **appending** the files !!

# Opening Files (Local): `open()`

```
file_variable = open(filename, mode)
```

'r' = read only (default mode)

'w' = write (If the file exists, erase its contents. If it does not exist, create a new file to be written to)

'a' = append a file (If the file exists, append data to the file. If the file does not exist, create a new file)

The location of a file – specify a path to a file

- If a data file and program are in different folders, use an *absolute* path
- If a data file and program are in the same folder, use a *relative* path
- For CS 1110/1111 homework submission/testing purpose and to avoid a “file not found” problem, **put all data files in the same folder as your python files** and specify a *relative* path.

Example

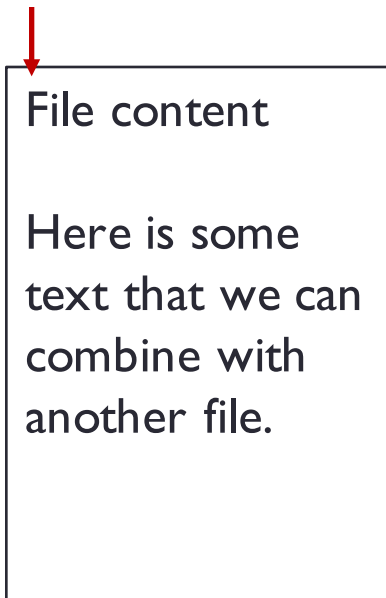
```
test_file1 = open('myfile.txt', 'r') # relative path
```

```
test_file2 = open('c:/cs1111/myfile.txt', 'r') # absolute path
```

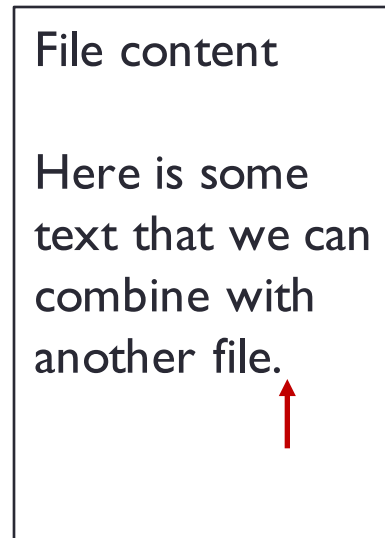
# Opening Files

---

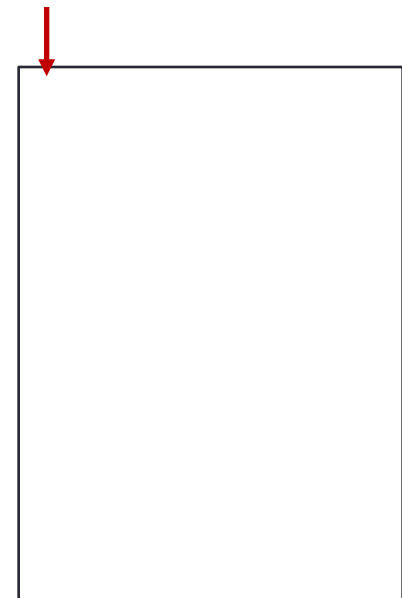
Open with  
a "read" mode



Open with  
a "append" mode



Open with  
a "write" mode



# With open

---

```
# check if a file is csv and exists  
with open (filename) as handle_name:  
    statements
```

The file will be closed automatically after the statements block

```
def read_file_with_open(filename):  
    with open(filename) as infile:  
        file_contents = infile.read()  
  
    print(file_contents)
```

# Reading from Files: `readline()`

```
file_variable = open(filename, 'r')  
file_variable.readline()
```

```
def readline_file():  
    # open a file named students.txt to be read  
    infile = open('students.txt', 'r')  
  
    # read 2 lines from the file  
    line1 = infile.readline()  
    line2 = infile.readline()  
  
    print(line1, line2)  
    print(line1, line2)  
  
    # close the file  
    infile.close()
```

Read **one line**

# Reading from Files: `readline()` with loops

```
file_variable = open(filename, 'r')
file_variable.readline()
```

```
def readline_file_with_loop():
    # open a file named students.txt to be read
    infile = open('students.txt', 'r')

    line = infile.readline()
    while line != "":
        print(line.rstrip('\n'))
        line = infile.readline() # what happen if this line is commented out

    # close the file
    infile.close()
```

# Reading from Files: `read()`

```
file_variable = open(filename, 'r')  
file_variable.read()
```

```
def read_file():  
    # open a file named students.txt to be read  
    infile = open('students.txt', 'r')  
  
    # read the file's contents  
    file_contents = infile.read()  
  
    print(file_contents)  
  
    # close the file  
    infile.close()
```

Read the **entire** file

- Read from the beginning of the file until the end of file, EOF (i.e., empty string)



# Reading from Files: `readlines()` and `read()`

Read **all the lines** in the file and return a **list of strings**

```
file_variable = open(filename, 'r')
buffer = file_variable.readlines()
buffer = file_variable.read()
```

```
def read_file():
    # open students.txt to read
    infile = open('students.txt', 'r')

    # read the file's contents
    file_contents = infile.readlines()

    for line in file_contents:
        print(line)

    # close the file
    infile.close()
```

```
def read_file():
    # open students.txt to read
    infile = open('students.txt', 'r')

    # read the file's contents
    file_contents = infile.read()

    for line in file_contents:
        print(line)

    # close the file
    infile.close()
```

# Writing to Files: `write()`

```
file_variable = open(filename, 'w')  
file_variable = write(string_to_be_written)
```

```
def main():  
    # open a file named students.txt  
    outfile = open('students.txt', 'w')  
  
    # write the names of three students to the file  
    outfile.write('John\n')  
    outfile.write('Jack\n')  
    outfile.write('Jane\n')  
  
    # close the file  
    outfile.close()
```

Where is this file?  
What does it look like?

```
John\nJack\nJane\n
```

Beginning of the file

End of the file

# Appending to Files: `write()`

```
file_variable = open(filename, 'a')  
file_variable = write(string_to_be_written)
```

```
def main():  
    # open a file named students.txt  
    outfile = open('students.txt', 'a')  
  
    # write the names of three students to the file  
    outfile.write('Mary\n')  
  
    # close the file  
    outfile.close()
```

Where is this file?  
What does it look like?

John\nJack\nJane\n

Beginning of the file      End of the file

John\nJack\nJane\nMary\n

Beginning of the file      End of the file

# Closing Files: `close()`

---

```
file_variable.close()
```

```
def read_file():  
    # open a file named students.txt to be read  
    infile = open('students.txt', 'r')  
  
    # read the file's contents  
    file_contents = infile.read()  
  
    print(file_contents)  
  
    # close the file  
    infile.close()
```

# Wrap Up File Operations

Function header (or signature)

argument

```
def read_list_of_names(filename):
```

```
    names = []
```

```
    datafile = open(filename, "r")
```

```
    outfile = open(filename, "a")
```

```
    for line in datafile: ← For each line in datafile
```

```
        line = line.strip() ← Strip leading and tailing spaces
```

```
        names.append(line) ← Append string to a list
```

```
        outfile.write(line) ← Write string to outfile (must be string)
```

```
    datafile.close() ← Close files
```

```
    outfile.close()
```

```
    return names ← Value-return function
```

Function call

```
print(read_list_of_names("names.txt")) names
```

Why isn't a file opened with a "write" mode ?

If we want to open a file with a "write" mode, how should we modify the code ?

← Open a file with a "read" mode

← Open a file with an "append" mode

← Write string to outfile (must be string)

← Close files

← Value-return function

parameter

# Validating a File

---

```
from os.path import *
```

```
# get file name
```

```
file_name = input("Enter csv file name: ")
```

```
# check if a file is csv and exists
```

```
while (not file_name.endswith("csv") or not isfile(file_name) ):
```

```
    file_name = input(file_name + " is not csv or does not exist. Enter csv file name: ")
```

# Summary

---

- Must know (based on exam2 topic list, as of 03/25/2019)
  - Read from files
    - `open(filename)`
    - `connection.read()`
    - `connection.readline()`
    - ways of iterating lines
      - `connection.readlines()`
      - `connection.read().split('\n')`
      - `for line in connection`