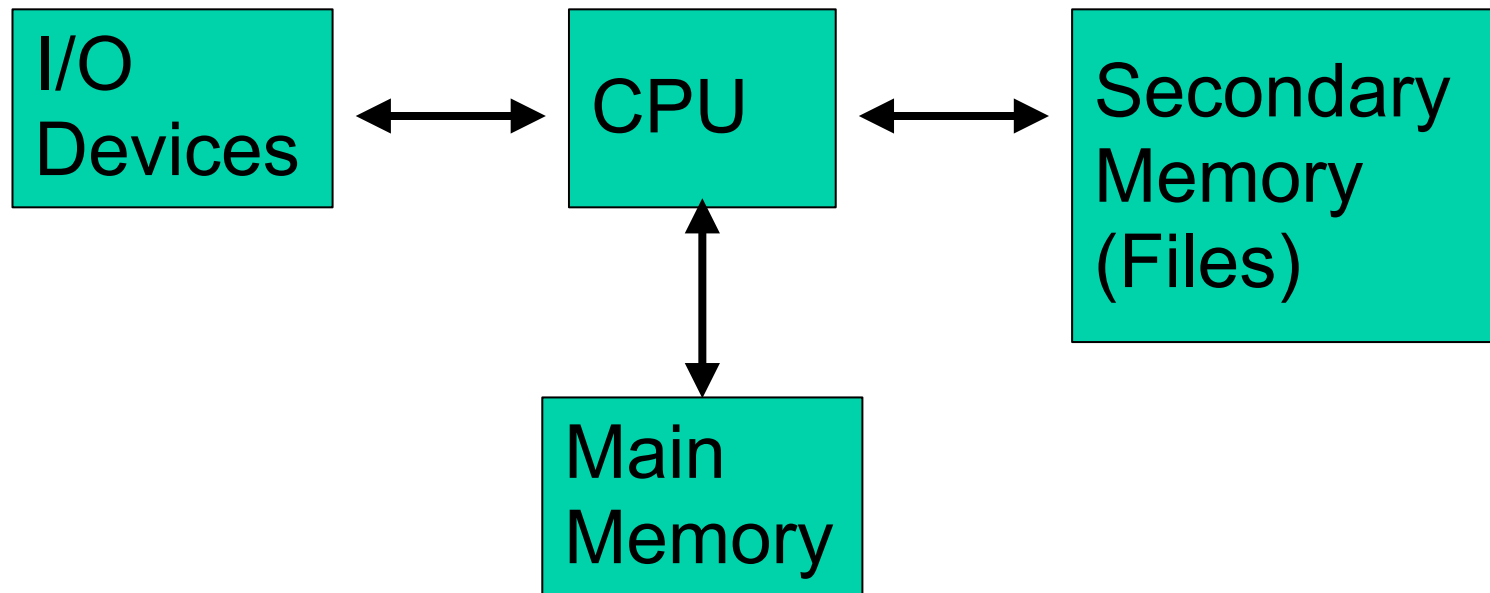# Basics of Programming

L07: I/O
Mar 2020

Dr. Ram P Rustagi
Professor, CSE Dept
KRP, KSGI
rprustagi@ksit.edu.in

Resources and Acknowledgements

- https://www.python-course.eu/ passing_arguments.php
- A first course in programming
  - https://introcs.cs.princeton.edu/python/20functions/
- Python for everybody
  - https://www.py4e.com
-

Computer Architecture

```
┌──────────┐        ┌──────────┐        ┌──────────┐
│ I/O      │ <────> │ CPU      │ <────> │ Secondary│
│ Devices  │        │          │        │ Memory   │
└──────────┘        └────┬─────┘        │ (Files)  │
                         │              └──────────┘
                         ↕
                    ┌──────────┐
                    │ Main     │
                    │ Memory   │
                    └──────────┘
```

Files

- Two kind of files
  - Text files : contains only ASCII charactrs
    - e.g. computer programs (.py) , text contents
    - consists of number of text lines
      » separated by line separator (\n, \r\n, \r)
      » can be created by notepad (or equivalent)
  - Binary files
    - (e.g. images, .exe files)
    - Word documents: (not a text file)

File Operations

- Create
- Delete
- Read
- Write
- Append
- Modify/update
- Seek

File Operations

- Opening a file
  - to read, write (create new or overwrite)
  - `fh=open(`*filename*`, [`*mode*`])`
    - e.g. returns a file handle
    - *filename* is a file on the system (text string)
    - *mode*: indicates: `r, w, a r+, w+, a+`
      - `rb, wb, ab, rb+, wb+, ab+`
      - when not specified, then default is "`r`" (read)
    - essentially, a variable to work on the file
    - example: `fh=open("students.txt", "a")`
      - `print(fh)` #prints value of fh and not file content
    - The handle `fh` acts more like a list in iteration

Text File

- Consists of a number of text lines
- Each line is terminated by a newline character($\backslash$n)
- A line can have different size from other
  - variables length size
- Example:

```
x="Python\nProgramming"
print(x)
```
Python
Programming

- Can iterate over file handle like a list

```
fh = open("students.txt")
for line in fh:
    print(line)    #prints each line
```

Processing Text file

- Counting number of lines in a file

```
filename = "students.txt"
fh = open(filename)
cnt = 0
for line in fh:
  cnt += 1
print("number of lines = ", cnt)
```

- Reading whole file in a string

```
fh = open(filename)
content = fh.read()
print(content[:5])
```

Processing Text file

- Checking for some text in a file i.e. searching

```
filename = "students.txt"
fh = open(filename)
patten = "I-"
for line in fh:
    if line.startswith("A"):
        print(line)
```

- Line from file contains '\n' and print() also adds it
- Display line by line without empty line

```
fh = open(filename)
for line in fh:
    print(line.rstrip())
```

Processing Text file

- Print lines that doesn't starts with "A"

```
filename = "students.txt"
for line in fh:
  line = line.strip()
  if line.startswith("A"):
    pass
  else:
    print(line)
```

Processing Text file

- Searching for a text e.g. "Abhi"

```
filename = "students.txt"
pattern = "Abhi"
for line in fh:
  line = line.strip()
  if pattern in line:
    print(line)
```

Reading from Terminal

```
inp = input()
name = input("Enter your name: ")
print("Hello " + name)
```

- **print with formatting**
```
print("%3d/%3d=%6.3f" % (1, 3, 1/3))
```
- **print with format**
```
print("{0}/{1}={2}".format(1,3,1/3))
print("{2}={0}/{1}".format(1,3,1/3))
print("{2:6.3f}={0}/{1}".format(1,3,1/3))
```

Using Command Line Arguments

- import the `sys` module.
  - `sys.argv` is the list of input parameters
  - len(`sys.argv`) is count of parameters
  - Each argument is taken as string
    - Needs to be converted appropriately
- Example of adding two numbers

```python
import sys
if (len(sys.argv) !=3):
  print("specify two arguments")
  exit()
first = int(sys.argv[1])
second = int(sys.argv[2])
sum = first + second
print(sum)
```

Copying File

- Reading from one file and write to another

```
import sys
sfile=sys.argv[1]
dfile=sys.argv[2]
fhs = open(sfile)
fhd = open(dfile, "w")
cnt=0
for line in fhs:
  fhd.write(line)
  cnt += 1
close(fhs); close(fhd)
print("Copied", cnt, "lines")
```

Exercises

- Ex 01:
  - Print 1 to 9 tables using formatted output, e.g.

```
1 x 1 = 1
1 x 2 = 2
.
.
1 x 10 = 10
2 x 1 = 2
.
.
2 x 10 = 20
.
.
.
.
9 x 1 = 9
.
.
9 x 10 = 90
```

Exercises

- Ex 02:
    - For a given text file,
    - for each line print number of characters in the line
    - For exmple, output should look like

```
Line 1 has 25 characters
Line 2 has 29 characters
 .
 .
 .
 .
Line 11 has 19 characters
```

Exercises

- Do the past exercises.

Questions