

# CPSC 231 - Lab

---

SYSTEM ARGUMENTS, FILE READ/WRITE, EXCEPTIONS

# System Arguments

---

You can pass some values when you run a python program

Python <program name> <Arg1> <Arg2> ...

## System Arguments (Cont.)

---

```
import sys
for i in range(1, len(sys.argv)):
    print("Hello", sys.argv[i].strip(), "!")
```

Note! The first argument is the program file name

# File Input/Output

---

How can you store data in a file:

1. Text file: Store Strings, ASCII or Unicode
2. Binary Files: Store bytes e.g. multimedia files or Objects

# File Input/Output

---

How can you store data in a file:

1. Text file: Store Strings, ASCII or Unicode
2. Binary Files: Store bytes e.g. multimedia files or Objects

# Open File

---

```
f = open("test.txt", "r") #Read file
```

```
f = open("test.txt", "w") #Write to file
```

```
f = open("test.txt", "a") #Append to file
```

If the file does not exist you cannot use "read" mode!

# Read File

---

```
f = open("test.txt", "r")
```

```
f.read() #read the whole file
```

```
f.readline() #read one line from the file
```

```
f.readlines() #read all the lines from the file and put in a list
```

```
f.seek(<integer number>) #move pointer to given position
```

# Write/Append to File

---

```
f = open("test.txt", "w")
```

```
f.write("test") #clear all the file and write "test" to file
```

```
f = open("test.txt", "a")
```

```
f.write("test") #keep the file data and write "test" to end of the file
```



# Exception

---

An exception is an event, which occurs during the execution of a program that disrupts the normal flow

try :

    <code segment that may cause error>

except <type> as <variable name>:

    <action to take when an exception occurs>

# Exception

---

try:

```
    a = int(input("Input a number"))
```

```
    print(10/a)
```

except ZeroDivisionError as e:

```
    print("The number is zero")
```

except ValueError as t:

```
    print("The input value is not an integer")
```