

### Files

- Variables are temporary
  - Value is lost when program ends
  - Value is lost if computer loses power
- Files provide a less volatile form of storages
  - Values are retained after the program ends
  - Values are retained when the computer loses power

# Types of Files

- Two types of files
  - Text files
    - Encoded using ASCII or Unicode
    - Can be viewed with editors such as Emacs and Notepad
    - Examples: Python source files, web pages, ...
  - Binary files

- Contain arbitrary sequences of bits which do not conform to ASCII or Unicode characters
- Examples: Images, word processor files, ...

### **File Access**

- · Two different ways to access data
  - Sequential Access
    - Start at the beginning of the file
    - Read data from the file in the order that it occurs
  - Random Access File
    - Jump to an arbitrary location in the file
    - Read some data
    - Jump to a new location
    - Read more data
    - ...

## Opening a Text File

- Text files are opened in one of three possible modes
  - Read
  - Write
  - Append

5

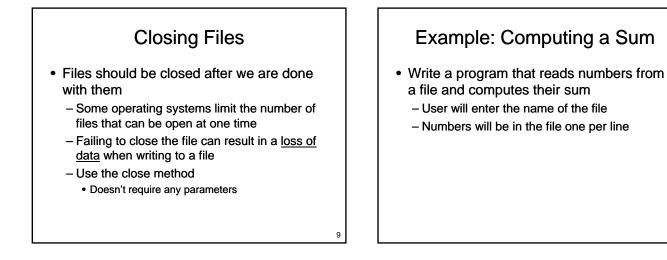
# Opening a Text File

- Files must be <u>opened</u> before they can be used
  - Use the open function
    - Parameter 1: The name of the file to open
    - Parameter 2: The one character abbreviation for
    - the mode
    - Returns a <u>file handle</u>
  - Example:

```
inf = open("input_file.txt", "r")
```

## Reading from a File

- Once a file has been opened, we can invoke methods on the file variable to read data from the file
  - read(): reads the entire file as a string
  - readline(): reads one line as a string
- Once we have read some data we often have to process it before it is ready for use by the rest of our program



Example: Computing a Sum

## **Special Characters**

- Inside of text files we have characters

   letters
  - numbers
  - punctuation marks
- Also have characters that control spacing
  - newline (sometimes a 2 char sequence!)
  - tab
  - other control characters

11

### **Escape Sequences**

- <u>Escape sequences</u> provide a mechanism for placing a character that controls spacing inside a string
  - What if we want to have a string that includes a newline character
  - What about a double quote character?
  - Use an escape sequence
    - Begins with \ (backslash)
    - Followed by one character describing the character that should be inserted

### **Escape Sequences**

- Common Escape Sequences:
  - n linefeed
  - \t tab
  - \" double quote
  - \\ backslash

## **Newline Headaches**

- Representation of newline varies by operating system
  - Unix and MacOS X newline is represented by the linefeed character, \n
  - DOS and Windows newline is represented by two characters: a carriage return followed by a linefeed, \r\n
  - On MacOS 9 newline is represented by a carriage return, \r

15

13

## **Command Line Parameters**

- Most programs require input to run
  - Can be read from the keyboard
  - Can be read from a file
  - Can come from parameters provided when the program is executed

## **Command Line Parameters**

- Command line parameters are stored in the variable sys.argv
  - A list with one element for each parameter
  - The element at index 0 is the name of the program
  - All parameters are handled as strings
  - Don't forget to import sys

17

### **Command Line Parameters**

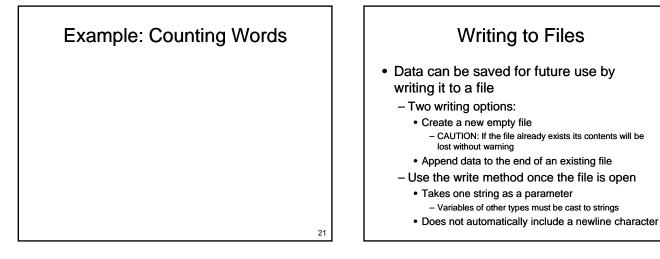
### **Command Line Parameters**

• Update our program for summing numbers so that the name of the file is passed as a command line parameter

### Example: Counting Words

- Write a program that computes some statistics about a text file
  - Number of lines
  - Number of words
  - Number of characters
- Filename will be provided as a command line parameter

19



## **Example: Counting Words**

- Extend our word counting program so that it writes the report to a file
  - Name of file will be provided as a second command line parameter

# Standard Input, Standard **Output and Standard Error**

Writing to Files

- · We have been using files since the first program that we wrote
  - Standard output is a file
    - · Values written go to screen
    - · Opened automatically when the program starts
    - Closes Automatically when the program ends
    - · File variable is sys.stdout
    - · We can write to stdout using the write method, just like any other file

23

## Standard Input, Standard Output and Standard Error

### • Standard Input

#### Also a file

- The raw\_input() function is equivalent to sys.stdin.readline().rstrip()
- The input function does additional work to determine what type of value to return

# Standard Input, Standard Output and Standard Error

- Standard Error
  - Another output file
  - Values written go to screen
  - Intended for displaying error messages instead of program output
  - Allows us to redirect program output separately from error messages
  - Useful for debugging

## **Example: Counting Words**

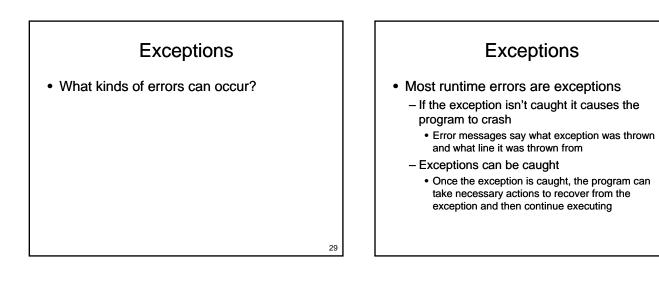
- Extend our program so that it is more flexible
  - If no parameters are provided input is read from the keyboard and output is sent to the screen
  - If one parameter is provided input is read from the file named as a parameter and output is sent to the screen
  - If two parameters are provided, input is read from a file and output is written to a file

27

25

### Files

- Text files provide an easy mechanism for loading and storing small amounts of data
  - Impractical for larger amounts of data that is frequently accessed (esp. in random order)
  - Accessing disks is slow
    - Important for programs that read and write frequently to optimize access to files
    - Common solution: Use existing database package



31

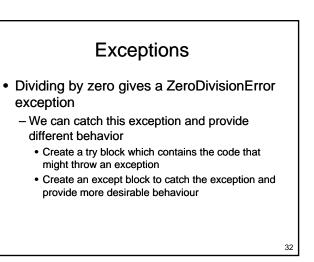
## Exceptions

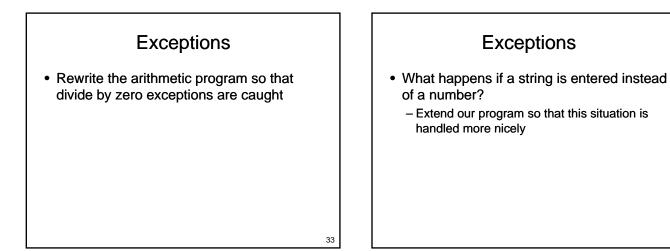
• Consider the following program:

```
a = input("Enter a number: ")
b = input("Enter another number: ")
```

```
print a,"+",b,"=",a+b
print a,"-",b,"=",a-b
print a,"*",b,"=",a*b
print a,"/",b,"=",a/float(b)
```

• What can go wrong?





## **Exceptions**

- Most file operations can throw exceptions
  - Try to open a file that doesn't exist
  - Try to read from a file that you don't have permission to read
  - Someone removes memory stick / CD while you are reading from it
  - These exceptions should be caught, even if the exception handler simply displays a meaningful message and quits the program

35

## **Multiple Exceptions**

**Exceptions** 

- · Each try block can have several except blocks
  - Each kind of exception needs to be handled differently
  - May include a default except block that doesn't specify the type of exception to catch · Catches all exceptions that aren't caught by
    - another except block
    - Typically displays an error message and exits

36

## Exceptions

• Exceptions:

- are thrown when an error occurs
- can be caught to recover from the error
- We have only scratched the surface:
   What happens if an exception is thrown inside a function?
  - How do we throw an exception ourselves?

### Databases

- A structured collection of records organized for ease of search and retrieval
  - Many commercial packages exist
    Oracle, Microsoft Access, SQL Server, DB2, ...
  - Some free options too
    MySQL, ...
  - Packages often provide a GUI so that a user can access the data

### Databases

- Most current databases use the relational model
  - Database consists of two parts
    - Schema: Describes the structure of the data
    - Data: The actual records being stored

### - Data is organized into tables

Each table consists of one or more (almost always) columns

39

37

## Example

• A table to store data about people



- How do we efficiently model the relationship that a person lives at an address?
  - Can more than one person live at an address?
  - Can a person have more than one address?

### Primary Keys and Foreign Keys

- Primary Key:
  - A unique value associated with each row in a table
  - Typically an integer
- Foreign Key:
  - A primary key value from another table residing in the current table

Primary Keys and Foreign Keys

• How do we model the possibility that many people may reside at one address?

Primary Keys and Foreign Keys

• How do we efficiently model the possibility that people may have several addresses?

43

41

## Accessing a Database

- Database runs as a separate process
  - Commands sent to database
    - Use a language known as SQL (Structured Query Language)
    - Human readable, reasonably intuitive
    - Much like working with QuickDraw
  - Results returned, often as lists
    - Program uses values retrieved from database to compute and present the desired result

45

### Databases

- Provide a ready-made solution for dealing with larger amounts of data
  - Careful database design is important
    - Avoid data duplication
    - Queries on large databases may need to be optimized
  - Tools are readily available
    - MySQL is free to download
    - Python libraries available for interacting with many different database packages

## Wrapping Up - Files

- · Files provide longer term storage of data
  - Types
    - Text files
    - Binary files
  - Can be opened for
    - Reading
    - Writing
    - Appending
  - Separate databases are commonly used to manage larger amounts of data

47

## Wrapping Up - Exceptions

- Exceptions
  - Many runtime errors are exceptions
  - Default behaviour: crash program
  - Exceptions can be caught
    - Put code that might cause an exception in a try block
    - Use except blocks to catch exceptions that can be recovered from

# Where Are We Going?

- Now you have a large set of tools:
  - Input, output, variables
  - If statements
  - For loops and while loops
  - Functions
  - Lists, dictionaries and strings
  - Files and exceptions
- These tools are sufficient to solve many interesting problems