

Programs Demonstrated in Lectures

Lecture 1:

- 1-1 – Introductory program demonstrating comments and multiple commands.
- 1-2 – Preview program for next lecture. Demonstrates input and variables.
- 1-VideoExercise – Video exercise solution

Lecture 2:

- 2-1 – Shows assignment to variables
- 2-2 – Illustrates variable types
- 2-3 – Illustrates computations with variables
- 2-4 – Basic interest calculation
- 2-5 – String concatenation
- 2-6 – Calculation with integer variables – days in weeks
- 2-7 – Calculation with string variables – concatenating strings to form an address
- 2-8 – Calculations for incrementing, decrementing, multiply-by, and divide-by
- 2-9 – Solution – decreasing an account balance
- 2-10 – Concatenations and the print statement
- 2-11 – The input command
- 2-12 – Illustrates that input is taken as strings
- 2-13 – Illustrates type conversion between different types, and problem with converting a float to an int
- 2-14 – Converting input to ints
- 2-15 – Solution – asking for a user's name
- 2-VideoExercise – Video exercise solution

Lecture 3:

- 3-1 – Nested conditional
- 3-2 – Boolean variable
- 3-3 – Assigning an expression to a Boolean variable
- 3-4 – Using an expression in a conditional
- 3-5 – Solution – defining conditions for nested conditionals
- 3-6 – Nested conditions – car maintenance
- 3-7 – Range checking with nested conditions
- 3-8 – Range checking with single Boolean expression
- 3-9 – Checking many options using nested if statements
- 3-10 – Checking many options with elif
- 3-11 – Dice value comparison for game of craps, using elif statements
- 3-12 – Dice value comparison for game of craps, using Boolean expression
- 3-VideoExercise – Video exercise solution

Lecture 4:

- 4 – Overall program developed

Lecture 5:

- 5-1 – Basic while loop
- 5-2 – Basic loop for counting to a limit
- 5-3 – Basic loop for counting down to some limit
- 5-4 – Loop for getting ages of those in a group
- 5-5 – Range examples
- 5-6 – While-else example
- 5-7 – Solution – Times tables from 1 to 10
- 5-8 – Solution – Printing pairs of people
- 5-9 – Example of continue statement
- 5-10 – Example of break statement
- 5-VideoExercise – Video exercise solution

Lecture 6:

- 6-1 – Opening and closing files
- 6-2 – Writing a string to a file
- 6-3 – Writing numerical data to a file
- 6-4 – Reading from a file
- 6-5 – Solution: Reading numerical data from a file
- 6-6 – Reading all lines from a file
- 6-7 – Examples of escape characters
- 6-8 – Examples of multi-line comments

Lecture 7:

- 7-1 – Accessing elements of a list
- 7-2 – Solutions: days in each month of the year, and handling leap year
- 7-3 – Solution: Looping through all elements of a list
- 7-4 – Solution: Days in year from list of days in each month
- 7-5 – Combining and appending lists
- 7-6 – Building a list by appending
- 7-7 – Negative index and Slicing examples
- 7-8 – Solution: slicing to get parts of a year
- 7-9 – Slicing on strings
- 7-10 – String manipulation functions and List examples
- 7-11 – Solution: working with lists of lists
- 7-12 – Tuple example
- 7-VideoExercise – Video exercise solution

Lecture 8:

- 8-1 – Examples of split command, including solution
- 8 – Final program developed
- Datafile1 – Data file used in program

Lecture 9:

- 9-1 – Simple function definition and call
- 9-2 – Repeated function calls for one definition
- 9-3 – Functions to represent discrete ideas
- 9-4 – Returning values from functions
- 9-5 – Returning multiple values from functions
- 9-6 – More examples of returning multiple values from functions
- 9-7 – Passing a parameter to a function
- 9-8 – Defining a function to hide inner workings from main code
- 9-9 – Taking multiple parameters and building more complex functions by calling simpler ones
- 9-10 – Side effect – function changing list contents
- 9-11 – Docstring example
- 9-VideoExercise – Video exercise solution

Lecture 10:

- 10-1 – Parameter passing and return
- 10-2 – Scope of local parameters
- 10-3 – Accessing data in main program from a function
- 10-4 – Global variable declarations in functions
- 10-5 – Mutable and immutable data types passed as parameters
- 10-6 – Mutable data being modified or assigned to
- 10-7 – Default parameters
- 10-8 – Madlib showing default parameters and

Lecture 11:

- 11-buggy – Buggy version of program that is debugged in the lesson
- 11-1 – Exception handling example
- 11-2 – Multiple exception blocks
- 11-3 – Raising exceptions
- Datafile1 – Data file used in program

Lecture 12:

- 12-1 – using some basic modules from Python Standard Library
- 12-ModuleA – Code to be saved separately as ModuleA.py for use in other modules
- 12-2 – importing and using a user-defined module
- 12-3 – many examples of modules from Python Standard Libraryy
- 12-4 – Picture renaming
- 12-VideoExercise – Video exercise solution

Lecture 13:

13 – Game program developed in lecture

Lecture 14:

14-1 – Turtle graphics: square spiral

14-2 – Encapsulating turtle commands into a function

14-3 – Creating and combining shape functions to draw a house shape

14-4 – Generating tally marks

14-5 – Robot path for room coverage

Lecture 15:

15-1 – Early Pyglet program – printing text labels in a window

15-2 – Pyglet program to demonstrate key press capture

15-3 – Pyglet program to demonstrate mouse capture

15-4 – Pyglet program to load and display images

15-5 – Graphical interface to grid-based matching game

15-6 – Tkinter program showing button creation and operation

BlueTri.jpg – Image used for pieces

GreenSquare.jpg – Image used for pieces

OrangeDiamond.jpg – Image used for pieces

PurpleStar.jpg – Image used for pieces

RedHex.jpg – Image used for pieces

YellowCircle.jpg – Image used for pieces

Lecture 16:

16-1 – Basic data plot

16-2 – Plotting data formed automatically

16-3 – Plotting with color and symbols

16-4 – Plotting two sets of data on same axes

16-5 – Plotting 3 sets of data with 3 axes

16-6 – Graphing a mortgage over time

16-7 – Growth of a fixed investment over time

16-8 – Monte Carlo simulation of growth in an investment with fluctuating interest

16-9 – Simulation comparing many different investment options

16-10 – Simulation looking at retirement funds

Lecture 17:

- 17-1 – Basic class with an attribute
- 17-2 – Class with a mutable attribute
- 17-3 – The init constructor and instance attributes
- 17-4 – Passing parameters to the init constructor
- 17-5 – Methods in classes
- 17-6 – Accessor functions and private attributes
- 17-7 – Using a class as an attribute of a class
- 17-8 – Objects as mutable data types
- 17-VideoExercise – Video exercise solution

Lecture 18:

- 18-1 – Basic class example for review
- 18-2 – Alternative implementation (could delete this example)
- 18-3 – Football player example (complete with inheritance and polymorphism)
- 18-4 – Football player example (failure to define function)
- 18-5 – Creating, raising, and catching our own exceptions
- 18-6 – Writing to a file with JSON
- 18-7 – Reading from a file with JSON
- 18-8 – Writing objects with JSON
- 18-9 – Writing data with pickle
- 18-10 – Reading data with pickle
- 18-11 – Writing objects with pickle
- 18-12 – Reading objects with pickle

Lecture 19:

- 19-1 – Implementing a stack with a list
- 19-2 – Implementing a stack class as part of a solitaire game (note: code is not complete – will not run without writing more)
- 19-3 – Implementing a queue with a list
- 19-4 – Implementing a queue class and using it for order fulfillment
- 19-5 – Basic dictionary example
- 19-6 – Deleting from a dictionary
- 19-7 – Creating a dictionary incrementally
- 19-8 – Dictionary example for storing passwords
- 19-9 – Set operations
- 19-10 – Shopping list generation using sets

Lecture 20:

- 20-1 – Searching an unordered list
- 20-2 – Searching an ordered list
- 20-3 – Searching and returning an index
- 20-4 – Selection sort
- 20-5 – Insertion sort
- 20-6 – Using Python's built-in sort command (in-place)
- 20-7 – Using Python's built-in sort command in reverse (out-of-place)

Lecture 21:

- 21-1 – Basic recursive countdown program
- 21-2 – Merge sort
- 21-3 – Quicksort
- 21-4 – Recursive and iterative Fibonacci computations

Lecture 22:

- 22-1 – Node and edge classes for cities: finding population along a road
- 22-2 – Alternative edge formulation using indices rather than names (notice population of one city changed)
- 22-3 – Finding neighbors of cities using adjacency lists
- 22-4 – Neighbor count using an adjacency matrix
- 22-5 – Currency Arbitrage using a directed graph
- 22-6 – Node class for a tree
- 22-7 – Node class for a binary tree
- 22-8 – Binary search tree – insertion and sorting

Lecture 23:

- 23-1 – Breadth First Search to find a chain of friends
- 23-2 – Word game code
- dictionary.txt – file of dictionary words
- dictionary4letter.txt – file of 4-letter dictionary words

Lecture 24:

- 24-1 – Basic process spawning
- 24-2 – Spawning multiple processes with arguments
- 24-3 – Parallel version of word game
- 24-4 – Subprocess example
- dictionary.txt – file of dictionary words
- dictionary4letter.txt – file of 4-letter dictionary words