CS 1111 Notebook

8/31/15

Get a student membership at http:www.acm.org/membership/student/benefits/

Join <http://acm.cs.virginia.edu/>

A computer program is a file or collection of files that instruct a computer what to do. Write an algorithm first (set of instructions written specifically for one purpose)

9/2/15

Hardware: physical devices that make up a computer

* CPU (central processing unit)- part of the computer that runs programs \*most important, without, CAN’T run software!
* Main memory: where computer stores a program while program is running, and data used by program
  + KNOWN as Random Access Memory
  + ALSO has secondary storage which holds data for long periods of time
    - Disk drive, solid states drive (faster than disk drive, stored in solid state memory), flash memory (portable, no physical disk), optical devices (data encoded optically)
* Input data the computer collects from people/other devices
  + Input decide: component that collects the data
    - Keyboard, mouse, joystick
* Output data produced b computer for other devices
  + Can be audio/text/image
* Output devices: format and presents output
  + Video display and printers are examples
  + Disk drives and Cd recorders can be considered output devices
* Software: Everythig the computer does
  + Application software: programs that make computers useful for day tasks EG word processing, email, games
  + System software anre programs that control and mage basic operations of a computer: OS = controls operations of hardware omponents
* HOW computers store data?
  + All data in computers are stored in squerences of 0s and 1s, computers use binary numbering system and determines value of binary number sum position values of the 1s (position of digit j is assigned value 2^j-1)
* STORING CHARACTERS: Data stored in computer must e stored as binary number🡪 converted to numeric code and stored in memory.
* MOST important coding scheme is ASCII: limited b/c defines codes for only 128 characters.
* NOW unicode coding scheme is being used b/c more characters can be stored.
* ADVANCED NUMBER STRAGE to store neg. numbers and real numb, computers use binary numbering and encoding schemes. Used two’s compliment.
* Other types of data: digital is anything that stores data as binary numbers, pixels can be stored as numbers for colors, music etc. bits.
* HOW PROGRAM WORKS: CPU designed to perform operations of pieces of data: eg. Reading data, adding/subtracting numbers. Understands instructions written in machine language and included in its instruction set
  + Each brand of CPU has own instruction set.
* Program must be copied from secondary memory to RAM each time CPU executes it. CPU executes program in cycle: Fetch: read next instruction from memory into CPU 🡪 decode: CPU decodes fetched instruction to determine which operation to perform 🡪 execute: performs operation.
* FROM MACHINE LANGUAGE TO ASSEMBLY LANGUAGE: Impractical for people to write in machine language. Assembly language uses short words (mnemonics) for instructions instead of binary numbers AKA easier to work with. Assemblers (machines) translate assembly language to machine language for execution by CPU.
* High level v. low level language: Low level: close in nature to machine language (e.g. assembly language). High level language allows simple creation of powerful and complex programs… no need to know how CPU works or write lots of instruction. More intuitive to understand.
* KEYWORDS: predefined words used to write program in high level language
  + Operators perform operations on data (e.g. math operators to perform arithmetic)
  + Syntax: set of rules to be followed when writing program
  + Statement: individual instruction used in high-level language
* Programs written in high level languages must be translated into machine lang. to be executed
  + COMPLILER: translated high level language program into separate machine lang. program
  + Machine language program can be executed at any time.
  + INTERPRETER: translates and executes instructions in high level language program
    - Used in Python language, one instruction at a time, no separate machine language program
  + Source code: statements written by programmer… can have syntax errors and prevents code from being translated
  + IDE’s (Integrated development environments) are places where you can test code in the same place you write it and compile it. (EG PyCharm, eclipse)

HOW TO DO IT IN THIS CLASS:

# author, program description, input, algorithm @ top of program output\*expected &test output (as comments)

USING PYTHON

* Python must be installed with interpreter. Can be used in two modes:
  + Interactive mode: enter statements on keyboard
  + Script mode save statements in Python script. Will use in this mode.

WHEN IN INTERACTIVE MODE, NOT SAVED AS PROGRAM. To have a program use script mode can run in command line or in IDE

THE IDLE (Integrated Development Program) is the single program that has tools to write and execute and cover syntax. Downloaded with python. Runs in interactive mode. But need to save work.

Go watch “How computers work” by Joshua Hawcroft.

9/7/15

Why are we using an IDE (pycharm) for this? – It will help you suggest the programming and put down thoughts in a different format.

Designing a program

* Design the program MUST do before written in pseudocode, flowcharts, etc.
  + Most important step of cycle
  + Work with customer to get sense of what program should do
  + Ask questions about program details
  + Create one or more software requirements
  + ALGORITHM WRITTEN: set of well-defined logical steps that must be taken to perform a task
* Write the code
* Correct syntax errors
* Test the program
* Correct logic errors

Function: piece of prewritten code that performs an operation

Argument: data given to a function

Statements in a program execute in the order that they appear from top to bottom

Strings sequence of characters that is used as data

String literal: string that appears in actual code of a program

9/14/15

Data types

* Int= integer whole numbers
* Float = sed for decimal numbers / flating point numbers
* Str= strings = sequence of characters
* Bool = coolens, holds true or false
* List= stores multiple values of other data types in ordered sequence
* Set = like a list, but unordered
* Dict = short for dictionary, type stores info in key/value pairs

Nesting functions are functions within function. Like print(format(Word, font/decimal))

Saying format(variable, ‘.2f’), sep=’’) will put NO space near it.

NOTE: ‘’’ is the COMMENT LONG TERM multi-line output.

List(range(6)) will make a list (0-6) out of the range to 6.

List(range(1,10,2)) will go 1,3,7,9 and skip by 2’s.

Repetition operator can be used.

Numbers = [1] \*6

Output = [1,1,1,1,1,1]

The output of output[0] would be the first list.

BUT interesting, output[-1] would be the last value, output[-2] would be second to last value, etc.

Length function🡪 len(list) = size of list, number of items.

For n in numbers:

Print (n)

Will automatically indent. Need the indentation for the loop itself. Really big deal! Will just go through the list. (need for… in..)

9/23/15

While loop is a condition controlled loop causes a statement or set of statements to continue as a condition is true.

Need a control variable.

While k\_g == ‘y’ or k\_g == ‘Y’

Name = input (“Enter name:”)

Sales = float(input(“Enter sales:”))

Comm rate = float (input(“Enter commission rate”))

Commission = sales\*comm\_rate

K\_g = input(‘Do you want to evaluate another? Press y for yes or n for no’)

This will allow for the loop to continue or you can stop the loop. Sentinel value.

For loop iterates through string, list, dictionary. Is not eternal.

For num in [1,2,3,4,5]:

Print (num)

1

2

3

4

5

~ will also do strings

For n in [“Abc”, “def”, “ghi”]

Print n

Will print Abc, Def, Ghi on separate lines

For num in range(4)

Print(num)

1

2

3

4

Starts with 1, ends in 4

For n in range(1,4) \*will go through begin at 1 but does not print out 4. Goes to 3.

For num in range (1,5,2)will do increments or step values. Will not go up to 5 bc of range again

For n in range(1,11):

Square=n\*\*2

Print (n, ‘\t’, square) 🡨 will do a tab between n and square