New Orleans JAZZED about Engineering Education

Paper ID #14878

The First Course of Programming: Python, Matlab, or C?

Dr. Alireza Kavianpour, DeVry University - Pomona

Dr. Alireza Kavianpour received his PH.D. Degree from University of Southern California (USC). He is currently Senior Professor at DeVry University, Pomona, CA. Dr. Kavianpour is the author and co-author of over forty technical papers all published in IEEE Journals or referred conferences. Before joining DeVry University he was a researcher at the University of California, Irvine and consultant at Qualcom Inc. His main interests are in the areas of embedded systems and computer architecture.

Miss Sogand Kavianpour, University of California - Irvine

Miss Sogand Kavianpour received her B.S. degree in Electrical & Computer Engineering from the University of California, Irvine. She is currently working with a medical device company specializing in magnetic technology.

The first course of programming: Python or C?

Which programming language should we teach in the first course of programming to our undergraduate students of engineering? Most entering freshmen have no programming experience. We have to teach them the first steps of programming and swiftly bringing them to a level of skill, where they can use computing in their other courses: to analyze data for lab reports, to learn linear algebra, to solve problems in mechanics, problems involving differential equations, and writing a program to control a robot (embedded programming).

In this paper based on the data collected from three programming courses (CS10, CS12, and CS20), we evaluate and recommend a programming language for the first course of programming to our undergraduate engineering students. Python, C, and assembly languages are considered in this analysis. We also emphasis the importance of Malab for the senior engineering courses.

1-Inroduction

Python is a high-level object-oriented programming language. It is designed to be easy to program. Guido van Rossum[4,5,6] started the design of Python in 1980. Over the years, Python has gained popularity in a broad range of fields from web development, games, scripting language, science, and engineering. Python is an open source software, and can thus be distributed freely, even for commercial use. This openness makes that Python plays well with the other languages and is easily expandable. Python is a general purpose language, which means that many things are made easy. Examples are string processing, reading/writing files, sockets, websites, and databases. This is why it has been adopted by so many universities. The ease of use for general tasks makes Python very suitable for education. For science this is also an advantage, as scientist often need to load data, visualize it, and maybe control it via a user interface. For commercial applications this means that many things work out of the box, saving time and money. Many users recommend Python as the best beginner language because of its simplicity yet great capabilities. The code is easy to read and enforces good programming style[7,8]. In Python, you can spend most of your time focusing on problem decomposition and data type design. These are important components of programming in any language. Python's large standard library allows students to create functional programs almost immediately. Python's interactive interpreter

allows you to test features while programming. Python's popularity is also rising quickly today because of its adoption on popular websites like Instagram.

Matlab[2] was designed at the University of New Mexico by Cleve Moler in the late 1970s. It soon spread to other universities and found a strong audience within the applied mathematics community. Later, Jack Little, a Stanford University engineer, recognized its commercial potential and founded Mathworks in 1983. Matlab is a commercial numerical computing environment and programming language. It has a lot of advanced toolboxes and functions, and they are oriented to circuit analysis, signal processing, financial analysis, economics, business, statistics, and optimization .These functions are well documented and developed. However they are proprietary and you cannot see the code of most of the algorithms you are using. The proprietary nature also makes it difficult for third parties to extend the functionality of this language. Matlab is not an open source language and you have to buy a license. However it is easier for beginners, because the package includes all you need.

Python and Matlab are interpreted language, which means that rather than compiling the complete program, bits of code can be executed in a running program/session. This is what makes it so suitable for scientific computing: you can quickly run part of code in a sequence, or run the same piece of code repeatedly while improving it.

C[1,3,9] language was developed in 1969 by AT&T, it is also considered a general-purpose programming language. Unlike Python, however, C is not a dynamic language easily adapted to script programming. C remains the most widely used programming language and it has seen much standardization and improvement throughout the years. Applications for Apple are written C. The static nature of C helps prevent unintended operations. The drawback is that as a new programmer, you may find it difficult to master quickly. It is still used extensively in operating system architecture. Systems including UNIX (and Linux derivatives) rely on C for many functions.

2-Teaching objectives and outcomes in computing

The main objective in teaching a computer language is to enable the students to convert engineering problems into an algorithm. This is a difficult task that requires analytical thinking. The conversion of this algorithm into a program written in one programming language is of secondary importance because it is, in principle, an algorithmic procedure and requires less effort. Consequently, the choice of the teaching language should be governed by which language provides the best support to the student in performing the implementation part of the problem.

3-Teaching Experience

3.1-Freshman programming courses

In this section data collected from several programming courses will be analyzed. One of these courses is the first course of computing for engineers. This course offered with Python in the Fall term (CS12) and with C language in the Spring term (CS10). Course Objectives and Outcomes for this course is as follows:

- **Course Objectives:**
  - To be able to design, develop, compile, and debug programs in a high-level programming language.
  - To be able to develop programs for solving numerical problems applicable to engineering.
- **Course Outcomes:**
  - Express data of multiple types (numbers, strings, lists) in a high level language.
  - Write expressions and functions in a high level language statements.
  - Use control flow constructs to combine multiple statements into a larger subprogram.
  - Test and debug programs with I/O constructs.

We will call these two courses as: CS12 and CS10. Data from the homework and final exams will be analyzed. The following homework assigned to both courses.

Homework 1: Write a program in Python/C to assign seats on each flight of the airline's only plane (capacity: 50 seats). Your program should display a graphic window as in the figure.

For occupation, type 1 for "student" type 2 for "staff", and type 3 for "faculty". If the user types 1, the program should assign a seat in the first class section (seats 1–20). If the user types 2, the program should assign a seat in the business section (seats 21–30). If the user types 3, the program should assign a seat in the economy section (seats 31–50). This program should display the seat number and then save all the information in a file called **passenger.txt** and then prints a boarding pass indicating the person's seat number and whether it is in the first class, business, or economy section of the plane. Use a List to represent the seating chart of the plane. Initialize all the elements of the List to 0 to indicate that all seats are empty. As each seat is assigned, set the corresponding element of the List to 1 to indicate that the seat is no longer available. Your program should, of course, never assign a seat that has already been assigned. When the first class section is full, the program should ask the user if it is acceptable to be placed in the business or economy sections (and vice versa). If the answer is yes, then make the appropriate seat assignment. If answer is no, then print the message "We are booked". The boarding pass format is as follows:

John Smith 19 First class.

Homework 2: (**Guess the Number**) Write a Python/C program that plays the game of "guess the number". Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 1000. The program then types:

I have a number between 1 and 1000. Can you guess my number?

Please type your first guess.

The player then types a first guess. The program responds with one of the following:

1. Excellent! You guessed the number! Would you like to play again (y or n)?
2. Too low. Try again.
3. Too high. Try again.

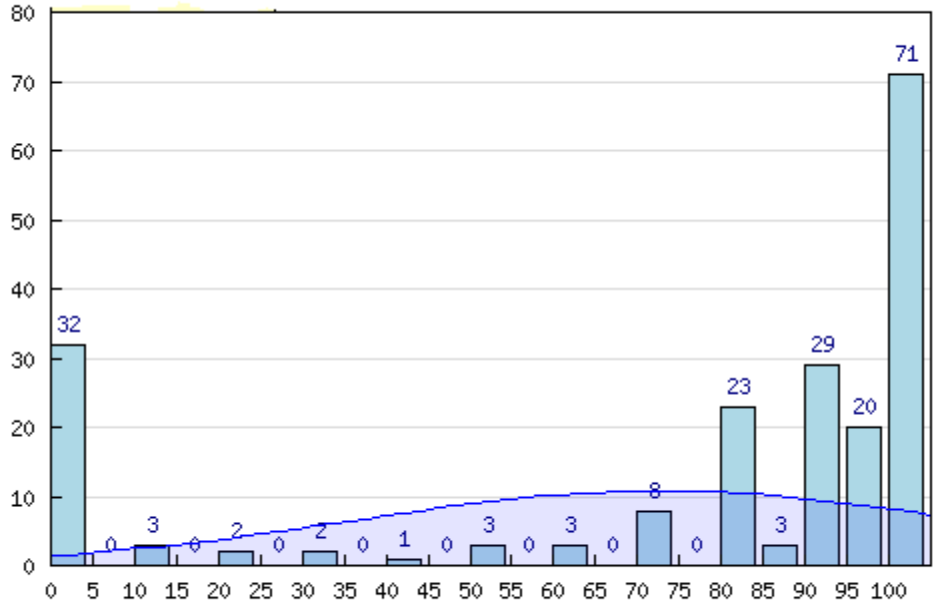
If the player's guess is incorrect, your program should loop until the player finally gets the number right. Your program should keep telling the player too high or too low to help the player "zero in" on the correct answer.

3.1.1 Collected data

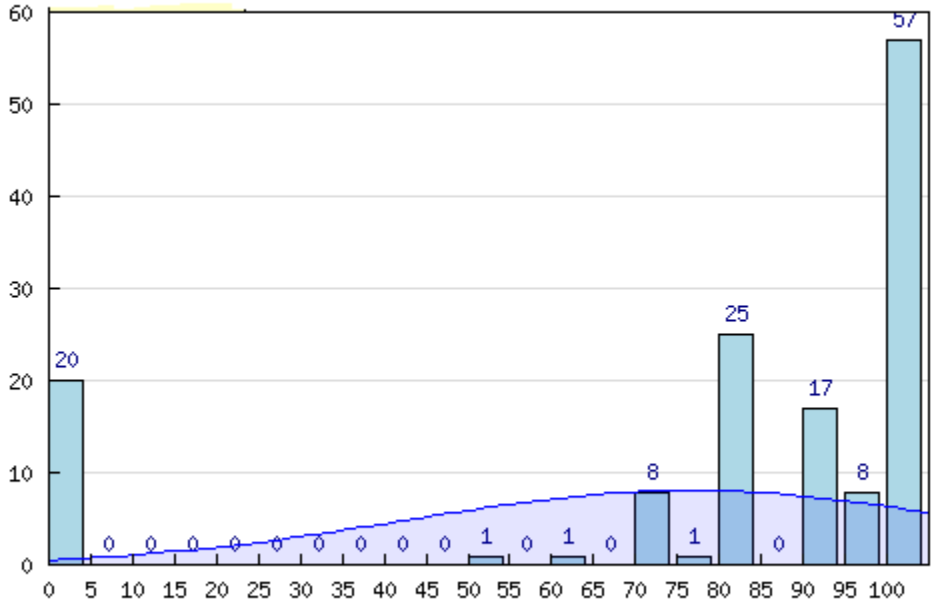
The following data collected from homework in these two courses:

Course Name	#student	Language	Mean	Median	Std
CS10	**199**	**C**	**73.86**	**90**	**36.6**
CS12	**126**	**Python**	**77.79**	**92**	**33.72**

CS10 Homework



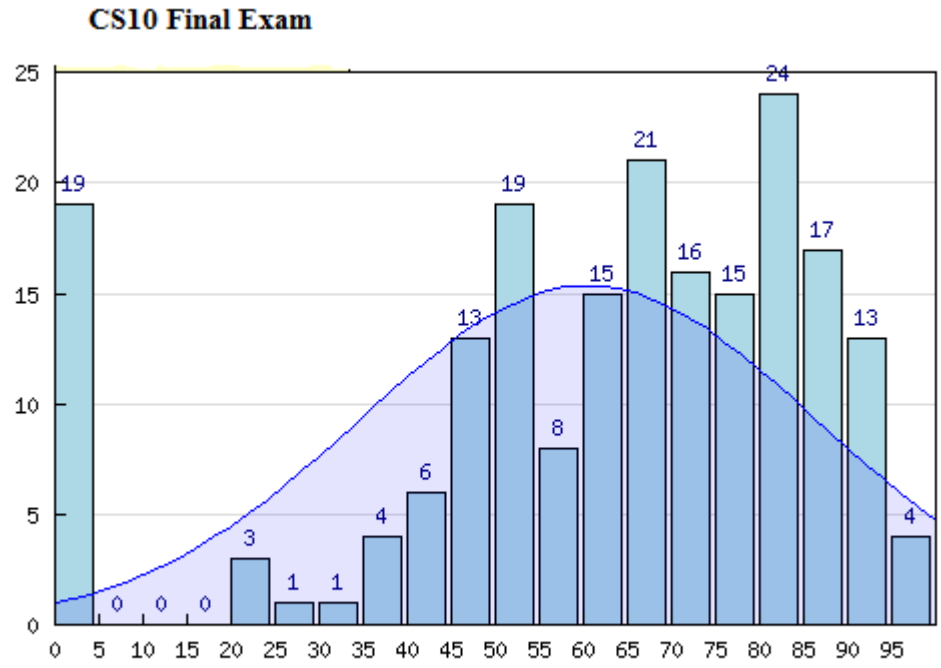
CS12 Homework

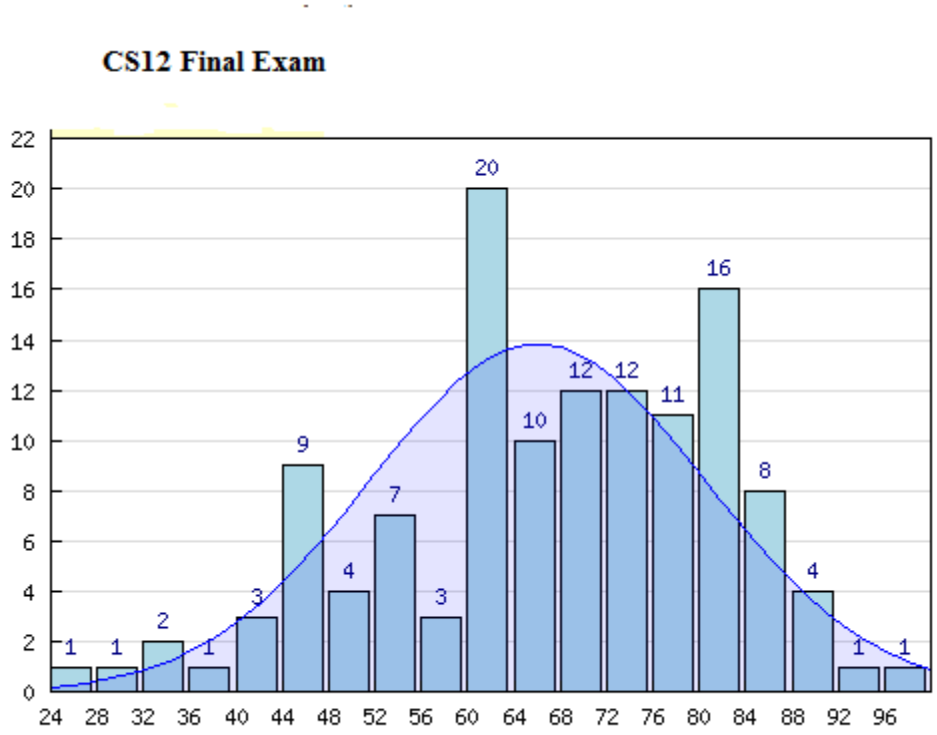


The same final exams were used in both courses; one with C language and the other with Python language.

The following data was collected from these two final exams:

Course Name	#student	Language	Mean	Median	Std
CS10	199	C	61.49	67	25.84
CS12	126	Python	66.81	68.25	14.53





Overall engineering students have better performance in the course with Python (CS12).

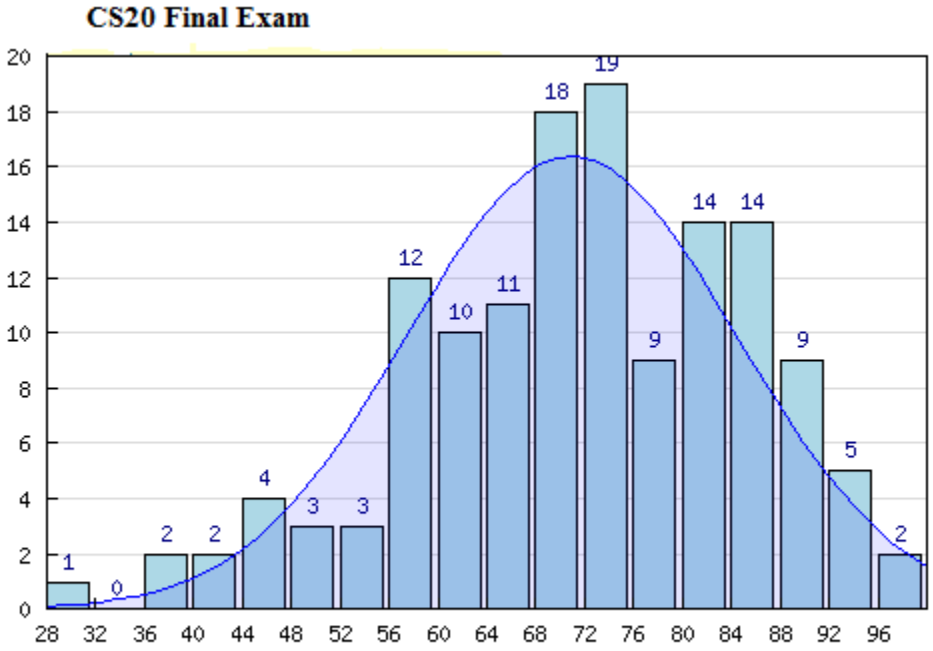
3.2 Sophomore programming course

CS20 is a programming course which is offered to the engineering students. This course is a combination of assembly language and C programming. The course objectives and outcomes for CS20 are as follows:

Course Objectives: Course objectives represent what the course strives to accomplish.
O To understand basic computer organization.
O To understand digital logic and computer arithmetic.
O To understand how advanced programming constructs and function calls are built at a very low level.
O To be able to develop programs using assembly language and C.
O To be able to develop programs that use arrays, structures, and linked lists.
Course Outcomes: Course outcomes represent what will be measured to determine if the course met its objectives.
O Describe how a computer represents and manipulates data.
O Describe and analyze the basic organization of a processor.
O Read and write assembly language and C programs.
O Perform stack operations and manipulate C pointers and arrays.
O Build programs with simple structures and linked lists.

We also compared performance of the students in CS12 and CS20 courses. The following data was collected from final exams:

Course Name	#student	Language	Mean	Median	Std
CS10	199	C	61.49	67	25.84
CS20	138	C + assembly	71.34	73	13.42



Based on the collected data, students have a better understanding of C programming if assembly language added to the course.

3.3 Senior programming course

In this section we review programming tools for senior students. Most of the senior engineering students used programming language to solve engineering problems. Matlab has a lot of advanced toolboxes and functions, and they are oriented to circuit analysis, signal processing, financial analysis, economics, business, statistics, and physics .These functions are well documented and developed. Following are some examples:

Solving non-homogeneous systems of Ordinary Differential Equation.

For solving the following ODE

y1' = y1 + y2 + exp(2x)
y2' = y1 + y2 + 10exp(x)
The following MATLAB codes are used:
>>[y1,y2]=dsolve('Dy1 = y1 + y2 + exp(2*x)','Dy2 = y1 + y2 + 10*exp(x)')

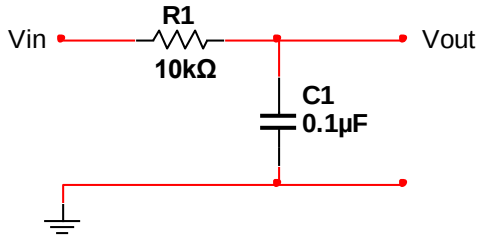
y1 =t*(exp(2*x)/2 - 5*exp(x)) - C9 + exp(2*t)*(C8 - exp(x - 2*t)*(exp(x)/4 + 5/2))

y2 =C9 - t*(exp(2*x)/2 - 5*exp(x)) + exp(2*t)*(C8 - exp(x - 2*t)*(exp(x)/4 + 5/2))

Second and Higher Order Differential Equation

For solving the second order Differential Equation
y″(x) + 8y′(x) + 2y(x) = cos(x); y(0) = 0, y′(0) = 1.
The following MATLAB codes are used:
>>eqn2 = 'D2y + 8*Dy + 2*y = cos(x)';
>>inits2 = 'y(0)=0, Dy(0)=1';
>>y=dsolve(eqn2,inits2,'x')
y =1/65*cos(x)+8/65*sin(x)+(-1/130+53/1820*14^(1/2))*exp((-4+14^(1/2))*x)
-1/1820*(53+14^(1/2))*14^(1/2)*exp(-(4+14^(1/2))*x)

Example of Bode Diagram for Low pass filter



$$\frac{V_{out}}{V_{in}} = \frac{Z_C}{Z_R + Z_C} = \frac{\frac{1}{Cs}}{R + \frac{1}{Cs}}$$

Simplify

$$\frac{\frac{1}{Cs}}{R + \frac{1}{Cs}} * \frac{Cs}{Cs} = \frac{1}{RCs + 1}$$

$$\frac{1}{RCs + 1} * \frac{\frac{1}{RC}}{\frac{1}{RC}} = \frac{\frac{1}{RC}}{s + \frac{1}{RC}} = \frac{\frac{1}{10k\Omega * 0.1\mu F}}{s + \frac{1}{10k\Omega * 0.1\mu F}} = \boxed{\frac{1000}{s + 1000}}$$

The following four lines of codes can be used to find the Bode plot:

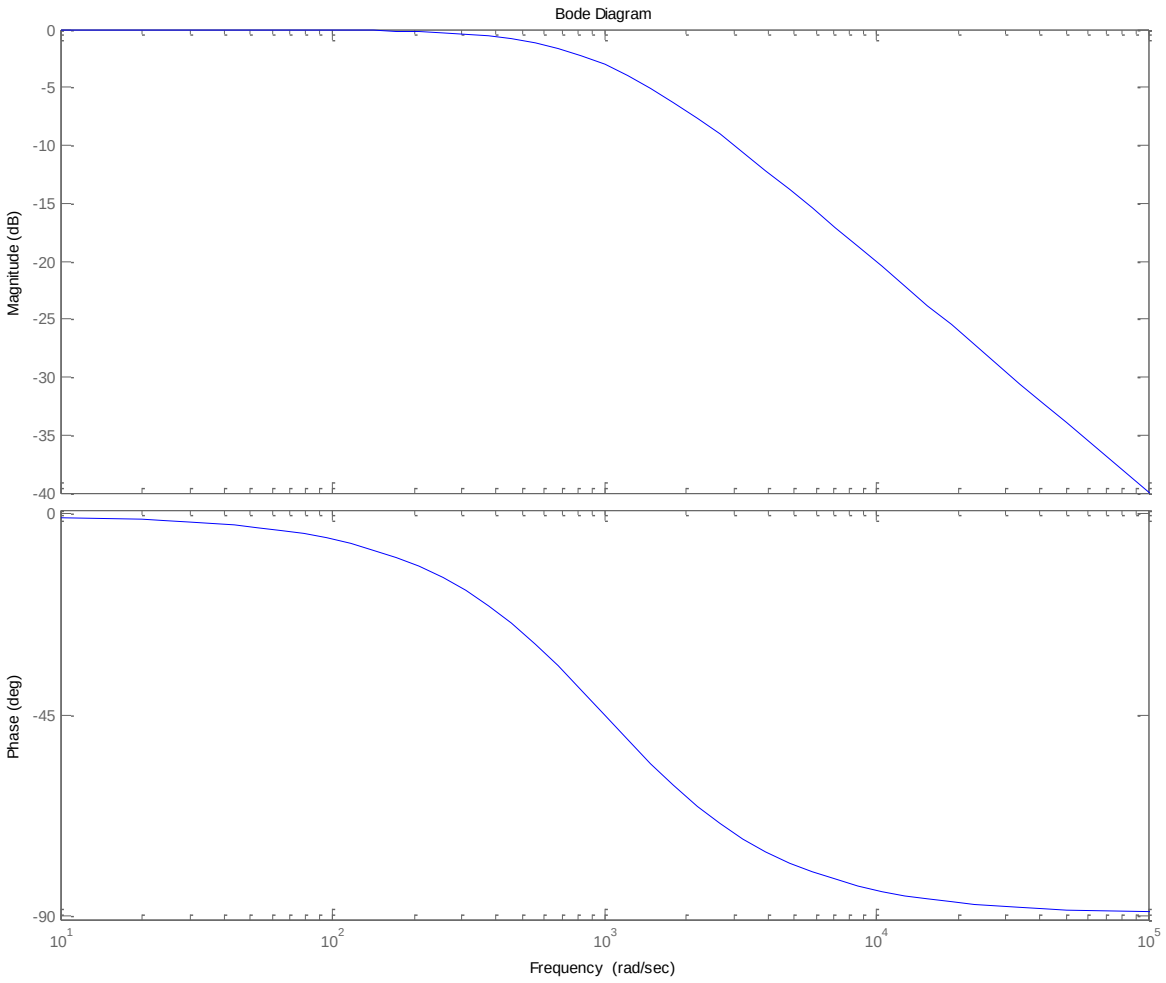
num = [1000];

den = [1, 1000];

TF = tf(num,den)

bode(TF)

1000/(s+1000)



Matlab is a useful programming tool for senior students. They can use this tool for problem solving in different areas of engineering.

4-Conclusion

In this paper, we analyzed the data collected from two freshman programming courses (CS10 and CS12) and one from sophomore programming course (CS20). Based on the outcomes of the collected data, we recommend Python for the first course of computing.
From derived data, students have a better understanding of the C language if assembly language is added to the course (CS20). Matlab is a great software tool for senior students who need to solve problems in the different areas of engineering. Further research will need to be conducted to test the efficacy of C# for higher levels courses.

5-References

1. Kernighan, B.W., Ritchie, D.M.: The C Programming Language. Prentice Hall Software Series (1988)

2. The Mathworks: Matlab (2003) www.mathworks.com.

3. Prechelt, L.: An empirical comparison of seven programming languages. IEEE Computer **33** (2000) 23–29

4. van Rossum, G.: Python tutorial. Centrum voor Wiskunde en Informatica (CWI), Amsterdam. (1995) www.python.org.

5. NumPy Team: Numerical Python (2003) www.pfdubois.com/numpy.

6. Donaldson, T.: Python as a first programming language for everyone. In: Western Canadian Conference on Computing Education. (2003)

7.Harvey M. Deitel, H. M. (Editor), Deitel P. J., Liperi J. P., Wiedermann B. A., P. Liperi, J. P., "Python, How to Program", Prentice Hall, Feb. 2002

8. Martelli, A., Ascher D. (eds): "The Python Cookbook", O'Reilly, 2003

9. Python Programming In A Day & C++ Programming Professional Made Easy (Volume 43) by Sam Key Create Space Independent Publishing, 2015, ISBN-13:9781511776752