Open LANSI

Open LANSI – extending python as a language for (L)inux server automation scripting (A)nalytics (N)etworking (S)ecurity (I)ntegration of (L) (A) (N) (S).

Project -01

Starts: 24th March 2018.

Due: 19th April 2018.

What is Open LANSI?

Open LANSI is an initiative for developing a Domain Specific Language (DSL), by extending python for following domains:

1. Linux server automation scripting.
2. Analytics
3. Networking
4. Security

Open LANSI will provide an integration platform for all 4 domains.

With Open LANSI, most complicated coding issues will be simplified, and everything will be simply offered as an object to the class of the extended python language.

With the intention to simplify the work for the global community we start developing this domain specific language, which not only focuses on solving the problem, but to make sure that solution proposed are more effective and worth the efforts, Open LANSI will also be training the developers, testers and leaders in technical as well as non-technical aspects. The training provided will be intensive and will test the participants not only in the technical aspects but also in terms of concentration, determination and self-confidence.

Open LANSI – is an open source project, in which all the people who are interested in their personal development can join. Forget about the technical skills you will learn, they are endless – but at the same time, you will also be learning some important life skills like – time management, setting up goals and priorities, which is way more take away then any technical skill set you will acquire.

We believe and trust people on their moral commitments. The more you invest yourself here, the more you will be progressive in the community as well as in your personal life as well.

Training rules:

As described below, the training will be intensive and focusing only on polishing the technical knowledge of the participants, but also their soft-skills and dedication.

Let's get on the same page with some basic rules:

1.) The training is INTENSIVE. The only way you can survive this is having PhD i.e. accepting that your knowledge is poor – leave the illusion that you know everything, show the hunger to gain more knowledge and be driven towards your goal.
2.) This will be a project based training, and specific time slots will be allocated for referring the concepts as well as for completing the projects.
3.) The projects will be posted on GitHub: https://github.com/yash7118/Open-LANSI/
4.) Each project must be completed in a single (PC – Project Cycle). If the project is large, it will be divided into separate project cycles from the very start itself. It will be made sure that a realistic deadline is given.
5.) Each PC = 18 days where day 1 = the release day of the project and day 18 is the 18th day from the day of release.
6.) Training is self-based. Well, the final decision of where the hints will be given and where not to be given will be dependent on you corresponding mentor.
7.) It is an open-ended project, no questions asked.
8.) After the project is due and submitted on the deadline, you will have 9 days from the date of submission to improvise the project. This is how it will work:
    a. On a common platform, where the project is uploaded, you will have to download, run and test 3 other project-submission of the same batch.
    b. Comment the flaws or appreciate the work if you find anything better in their code. Check and analyze the code from quality point of view.
    c. If you get a comment on your code, either explain why the code was right from your point of view and settle the doubt and improvise the code.
    d. More the interaction, more will be the chances of you being promoted to the mentor and leader's quadrant. As with the community expanding, we will need more leaders and mentors to train the community.
9.) If the projects are not submitted within the given time-line, the community will not wait. There is a solution to every problem.

Project 00:

Before you start project 01 – you are expected to know the basics of following programming languages:

1.) C
2.) Cpp
3.) Java
4.) Object Oriented Concepts.
5.) How Agile Software Development Life Cycle works?
6.) Git-hub – at least graphically. (For version control)

If you are not aware of any of there, get your hands dirty with them.

Project 01:

We will start with data structures and the first DS we are focusing on is Linked-list.

Questions like:

1.) What is linked-list and how it works?
2.) What is the time-complexity of a linked-list?
3.) How do I implement linked-list?

Figure out the answers on your own. Also make sure you learn the working of the various variants of linked list.

Let's skip to the project statement.

Project name: Evaluating Polynomials.

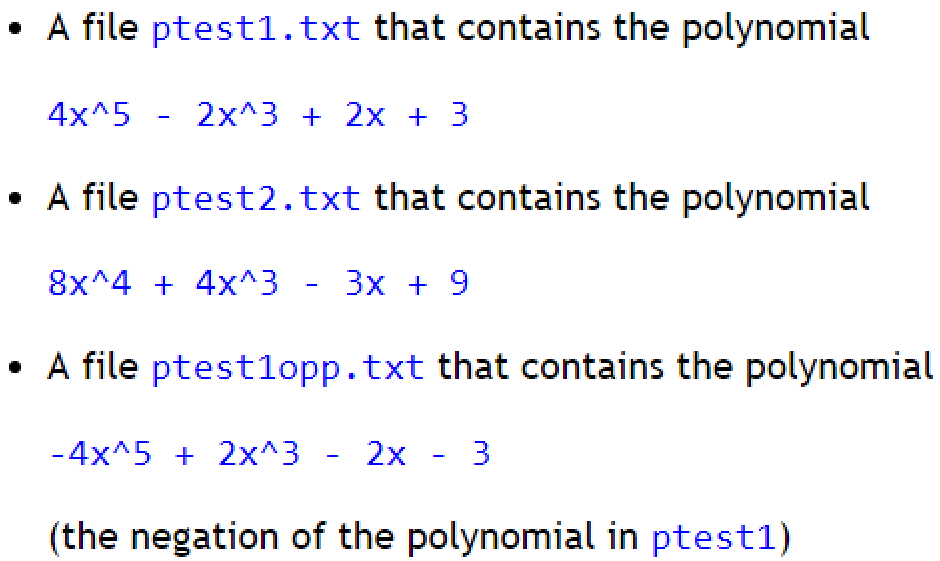
Project statement:

Write a program in the language of your choice to add, subtracts and multiplies the polynomials.

Eg: A polynomial: 4x^5 – 2x^3 + 2x + 3 can be represented in linked list as follows:

(3,0) -> (2,1) -> (-2,3) -> (4,5)

Create 3 test-files, where each test file will contain the polynomials as follows:



- A file ptest1.txt that contains the polynomial

  ```
  4x^5 - 2x^3 + 2x + 3
  ```

- A file ptest2.txt that contains the polynomial

  ```
  8x^4 + 4x^3 - 3x + 9
  ```

- A file ptest1opp.txt that contains the polynomial

  ```
  -4x^5 + 2x^3 - 2x - 3
  ```

  (the negation of the polynomial in ptest1)

Hints:

1.) You don't need to worry about "x" in the test-files, they can be inserted on the fly during the run time.
2.) You might want to create special files, which can be used to add zero or multiply one to a given equation – to check the identity properties.
3.) Above all, observe the sample output provided below properly.

Languages allowed: Cpp or Java. – Cpp preferable by Open LANSI for this project.

Sample output:

```
My name  => Yash Shah
LANSI id => LANSI-18

Enter the name of the polynomial file => ptest1.txt

4.0x^5 + -2.0x^3 + 2.0x + 3.0


1. ADD polynomial
2. MULTIPLY polynomial
3. EVALUATE polynomial
4. QUIT
        Enter choice # => 1
Enter the file containing the polynomial to add => ptest2.txt

8.0x^4 + 4.0x^3 + -3.0x + 9.0

Sum: 4.0x^5 + 8.0x^4 + 2.0x^3 + -1.0x + 12.0


1. ADD polynomial
2. MULTIPLY polynomial
3. EVALUATE polynomial
4. QUIT
        Enter choice # => 1
Enter the file containing the polynomial to add => ptest1opp.txt

-4.0x^5 + 2.0x^3 + -2.0x + -3.0

Sum: 0


1. ADD polynomial
2. MULTIPLY polynomial
3. EVALUATE polynomial
4. QUIT
        Enter choice # => 1
Enter the file containing the polynomial to add  => ptestnull.txt

0

Sum: 4.0x^5 + -2.0x^3 + 2.0x + 3.0


1. ADD polynomial
2. MULTIPLY polynomial
3. EVALUATE polynomial
4. QUIT
        Enter choice # => 2
Enter the file containing the polynomial to multiply  => ptest2

8.0x^4 + 4.0x^3 + -3.0x + 9.0

Product: 32.0x^9 + 16.0x^8 + -16.0x^7 + -20.0x^6 + 52.0x^5 + 38.0x^4 + -
6.0x^3 + -6.0x^2 + 9.0x + 27.0


1. ADD polynomial
2. MULTIPLY polynomial
3. EVALUATE polynomial
4. QUIT
        Enter choice # => 3
Enter the evaluation point x  => 2
Value at 2.0: 119.0


1. ADD polynomial
2. MULTIPLY polynomial
3. EVALUATE polynomial
4. QUIT
        Enter choice # => 4
```

What to submit:

1.) Your “well-commented ingenious” code – either .cpp or .java file.
2.) The code must compile on any environment.
3.) Mention the compiler version and the environment you are using. Try using the latest one. Also mention the LANSI-ID you are using.
4.) The Open LANSI community appreciates if the Code is developed as per standard nomenclature.
5.) The test-case files that you have used.
6.) A READ-ME file if you feel is required.

When to submit:

1.) The project starts from 24th March 2018.
2.) Time allocated to the project is 1.5 PC = 27 days.
3.) The project must be submitted by 19th April 2018.
4.) Reason for extended PC is because some of the members will have to complete project 00 before starting project 01.
5.) After this, we will have a check up session for next 9 days.
6.) If you are in leader’s quadrant, the deadline is half the deadline for the developers which makes it 6th April 2018.

Maintaining the project integrity:

1.) You can work “with” your group but not “in” group. As in you can discuss with whoever you want, take help from any online resource you wish to use, and get it done. But make sure when you submit the code, you submit your own original “self-developed” code.
2.) The purpose of the project is to make sure you develop not only the coding skills but also time management skills as well.
3.) Make your own logic, the community will not ask any question or judge you for any flawed logic or copied material. Remember – Your work here will be a direct impression of who you are. So, face it.

Logging:

1.) Log the number of hours you work on the project.
2.) Logging hours must be classified as follows:
    a. Learning hours – where you have learnt the basic concepts.
    b. Earning hours – where you productively implemented the project.

Happy Coding!

Your code is your reflection of your personality.