Lab 08 Specification – LISP
50 points

(Individual Lab -) Due by: 11/08/2021 2:00 PM

Lab Goals

- To explore a new programming language in LISP.

Learning Assignment

If not done previously, it is strongly recommended to read all of the relevant "GitHub Guides", available at the following website:
https://guides.github.com/
that explains how to use many of the features that GitHub provides. This reading assignment is useful to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, it is also recommended to do the reading assignment from the section of the course textbook outlined below:

- PLP chapter 08 and chapter 10.

Assignment Details

In this lab, students will get an opportunity to explore and implement in a new programming language in LISP. This is a good practice to understand functional programming languages and retain the knowledge from the class discussions so far.

Students are not expected to achieve technical mastery in any single language in this course. Instead, we are trying to learn different languages in the context of studying the principles of programming languages. In other words, our goal is to put the principles of programming languages at the forefront and not achieving language mastery.

At any duration during and/or after the lab, students are recommended to team up with the Professor and the TL(s) to clarify if there is any confusion related to the lab and/or class materials. The Professor proofread the document more than once, if there is an error in the document, it will be much appreciated if you can communicate that to the Professor. The class will be then informed as soon as possible regarding the error in the document. Additionally, it is highly recommended that students will reach out to the Professor in advance of the lab submission with any questions. Waiting till the last minute will minimize the student's chances to get proper assistance from the Professor and the Technical Leader(s).

Students are recommended to get started with this part in the laboratory session, by discussing ideas and clarifying with the Professor and the Technical Leader(s). Late submission is accepted for the part(s) in this section, based on the late policy outlined in the course syllabus.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share code files and/or other implementation details. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.

2. Submitting a copy of the other's program(s) and technical reports is strictly not allowed. Please note that all work done during lab sessions will be an opportunity for students to learn, practice, and master the materials taught in this course. By doing the work, students maximize the learning and increase the chances to do well in other assessments such as skill tests, exams, etc · · ·

Preliminary Steps

It is important that you can set up Docker and GitHub to complete the rest of the lab. Please follow the guidelines below to complete the preliminary steps.

1. **[Docker Setup.]** At this point, I expect the MAC, Linux, and Windows Pro users, to have this step completed based on our previous class discussions. For those who had not completed this step, the documentation below should provide more details regarding the download and installation setup.

   - Get Docker setup completed on your laptops:
   - Docker Mac Setup:

     `https:/docs.docker.com/docker-for-mac/install/`

   - Docker Ubuntu Setup

     `https://www.digitalocean.com/community/tutorials/`
     `how-to-install-and-use-docker-on-ubuntu-18-04`

   - Docker Windows Setup:

     `https:/docs.docker.com/docker-for-windows/install/`

   - If the setup goes correctly as desired, you should be able to get started and validate the Docker version and run the hello world docker container using the following commands:

     docker –version

     docker run hello-world

   - There are some more documentation for Docker get started to test your installation in the link provided below:

     `https:/docs.docker.com/docker-for-mac/`

     `https:/docs.docker.com/docker-for-windows/`

2. **[Loading Docker Container.]** There are two steps in loading the container, namely:

   - Build the container
   - Connect and Run the container

   Build the container: So to build the container. the following steps should be performed.

   (a) First, accept the lab URL provided in Slack. After downloading the lab folder from the GitHub classroom, navigate to the cmpsc201-fall-21-lab08 directory using terminal (Mac/Ubuntu) or Command Prompt/Docker quick start terminal (windows).

   (b) Build the docker image using the following command:

   docker build -t cs201lab08 .

   Please note, you are required to have the period in the command above.

   (c) Note: In the command above, cs201lab08 is the user-provided image name. This could be random. But it is recommended to use the same name to easily follow the rest of this document. Additionally, it is required to be inside the `cmpsc201-fall-21-lab08` directory to run the build command. If you are not inside the `cmpsc201-fall-21-lab08` directory, you may receive an error message.

   (d) Upon successful build, it is recommended to verify the correctness of image creation by using the following command:

   docker image ls

   (e) The image named "cs201lab08" should be listed as one of the outputs from the command above.

   Connect and Run the container: So to create and run the container. the following steps should be performed.

   (a) Run the docker container based on the image created in the previous steps using the following command:

   Mac/Ubuntu:

   ```
   docker run --rm -v $(pwd)/src:/root -it cs201lab08
   ```

   Windows:

   ```
   docker run --rm -v "%cd%/src":/root -it cs201lab08
   ```

   Ignore if you receive a bash command not found error while connecting to the docker container. The command above simply takes us into the docker container so that we can start to execute the code.

   (b) If you get an error while executing the command above on your windows laptop, then type the following:

   ```
   docker run --rm -v $pwd\src:/root -it cs201lab08
   ```

   (c) To run the above command, it is required to be inside the `cmpsc201-fall-21-lab08` directory. And, please note, you will log in to the container after entering the above command.

   (d) After creating the container, the run command above creates a mount between the host machine and the container with a shared folder space. So, any files placed inside the host mount directory can be easily accessible inside the container mount directory and vice versa.

3. **[GitHub Setup.]** Take a look at the detailed documentation for getting started with GitHub, which is available at: `https://docs.github.com/en/get-started`

   You are required to know the procedure to git clone, git pull, git add, git commit, and git push to access the lab specification folder and to submit your lab for grading purposes. If there is an issue with your GitHub setup please discuss it with your Technical Leader(s) and/or the Professor.

Part 01 - Exploring LISP (20 points)
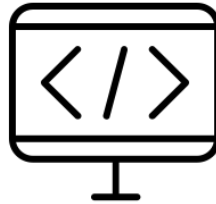
LISP is an oldest language after Fortran. One of the pioneers of Functional Programming Languages, LISP is best used to process any dataset that can be structured as a list. Our goal is not to become a master in LISP programming. But instead, our goal is to use LISP as a base, to understand some fundamental concepts behind Functional Programming. In the last few classes, we had discussed some fundamental concepts in Functional Programming and LISP such as Variable Scoping [Local Variables and Global Variables], Conditional Logic, Recursion, LIST processing techniques such as "CAR" and "CDR". By using the preliminary learning that we had done in our class so far, let us try to do more LISP coding in this lab assignment. You may either use the docker container or use the online lisp compiler using the link: `https://www.tutorialspoint.com/execute_lisp_online.php`

If you are using Docker to run your lisp code, the command to execute is:
clisp filename.lisp

Where Do I Start?

Remember, our goal is to learn new programming language on the fly and gradually remove the fear of PL's. By training our mind to tune to this mindset, we will increase our chances to be successful as a computer scientist and sustain a long career in CS. The labs in this course are aimed to achieve this goal. Now, let us learn a new functional programming language in LISP.

Go to the online tutorial "LISP tutorial" at: `https://www.tutorialspoint.com/lisp/` and just start reading. Skip the parts you think you don't need, but try the examples.

If you are one of those, who like to learn by watching a video tutorial, watch this YouTube video at: `https://www.youtube.com/watch?v=ymSq4wHrqyU&t=2461s`

The video itself can be long [1hr, 15 minutes]. Skip the parts you think you don't need, but try some of the examples mentioned in the video.

There are many other online tutorials available in the web. Feel free to explore other web resources. I recommend the above mentioned tutorials, as I find those as an interesting sources to learn LISP programming. But, feel free to look at other resources as well based on what you find.

I recommend you to also review the textbook chapter 10, lecture slides, lecture notes, code in the coding folder, and in-class activities to get more insight on LISP programming. You may also look up online resources to read about and answer these questions. Make sure to cite the references appropriately.

By using the `Technical-Report` document, answer the following questions:

1. How is functional programming different from imperative programming? You may refer to online sources to best answer this question. Additionally, I recommend presenting your solution by using a table that list all the differences that you are able to find.

2. Explain one fundamental concept in Functional Programming and provide an implementation example using LISP programming language? You may provide a simple example, just to showcase your conceptual understanding.

3. Present one example of Recursion by providing an implementation example in both Java and LISP? Compare both Java and LISP by clearly stating the difference between both approach in term of implementation details. We had already discussed Fibonacci example in class. So, present an example that is different from Fibonacci sequence.

4. Explain the difference between "CAR" and "CDR" by providing an example? Is it possible to use the combination of "CAR" and "CDR"? If your answer is yes, then state clearly if there is an actual limit (in terms of number of times) to utilize such a combination in LISP.

5. What is one thing you like about LISP programming?

6. What is one thing you dislike about LISP programming?

Part 02 - Implementing LISP (30 points)
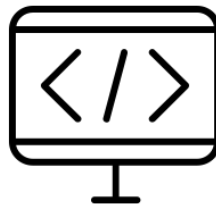
To learn new programming languages, we need to learn how to read online documentation and utilize online resources like Stackoverflow to help us implement the programs effectively. A required skill for learning new programming language and to become a better computer programmer is to learn how to use online resources effectively and ethically (that is to correctly cite references). In this lab, we will practice these skills as well.

We completed the chapter on subroutines and started discussing functional programming, so technically we could use any language to examine the concepts on data types. But I think one of the things people look forward to in this course is learning new languages, and based on a few favorable reactions from previous labs, I thought it might be a good idea to continue on in the "learn new languages" mode for a while and explore **LISP**. Note: This part of the lab is **very** open ended. You just need to explore a new language by going through a set of tutorials. This part is purposefully made open ended so that the learners understand how to learn and use new programming languages through trial and error approach. First to complete this part, we need to come out of a narrow mindset of learning and embrace the challenges to learn in open ended setting by training our mind to tune to a broader mindset of learning.

You may look up online for some help to help you answer these questions correctly. By using lisp code, answer the following questions:

1. Write a function in LISP called "EvenorOdd" inside a lisp program called EvenorOdd.lisp, that would take an input list and find if the total number of items in the list, is an even number or odd number. You are not supposed to use built-in length function, instead you may use an iterator to count the number of items in list. You may use similar approach, as we had discussed in class examples. After, finding the length of the list, you may need to use conditional operators and mathematical operators to print in console, if the length is even or odd. Inside your LISP code, you are required to call "EvenorOdd" function by passing a list with multiple elements. Test your code, to make sure the code correctly prints the even or odd statement as indicated in the requirement above.

2. Write a function in LISP called "palcheck" inside a lisp program called PalCheck.lisp, that would take an input list of characters and find if the elements in the list is a palindrome or not. An example palindrome are 'M' 'O' 'M' and 'D' 'A' 'D'. You may use built-in functions to handle this requirement. I am not providing more details on the implementation steps for this part, as I want you to think and figure out the implementation steps by yourself. Your code execution should state clearly if the given list is palindrome or not. You may hardcode the list in your LISP code.

Part 03 - Honor Code

Make sure to **Sign** the following statement in the `honor-code.txt` file in your repository. To sign your name, simply replace Student Name with your name. The lab work will not be graded unless the honor code file is signed by you.

This work is mine unless otherwise cited - Student Name

Part 04 - Reflection

Add a Reflection to the repository by modifying the `reflection` file in the lab repository. List out the biggest learning points and any challenges that you have encountered during this lab.

Submission Details

For this assignment, please submit the following to your GitHub repository by using the link shared to you by the Professor:

1. Modified source code in `EvenorOdd.lisp` file.

2. Modified source code in `PalCheck.lisp` file.

3. A document containing the technical answers to the questions on LISP in the file named `Technical-Report`.

4. A document containing the reflection of the lab in the file named `reflection`.

5. A signed honor code file, named `Honorcode`.

6. To reiterate, it is highly important, for you to meet the honor code standards provided by the college. The honor code policy can be accessed through the course syllabus.

Grading Rubric

1. There will be full points awarded for the lab if all the requirements in the lab specification are correctly implemented. Partial credits may be awarded if deemed appropriate.

2. Failure to upload the lab assignment code to your GitHub repository will lead to receiving no points given for the lab submission. In this case, there is no solid base to grade the work.

3. There will be no partial credit awarded if your code doesn't compile correctly. It is highly recommended to validate if the correct version of the code is being submitted before the due date and make sure to follow the honor code policy described in the syllabus. If it is a late submission, then it is the student's responsibility to let the professor know about it after the final submission in GitHub. In this way, an updated version of the student's submission will be used for grading. If the student did not communicate about the late submission, then automatically, the most updated version before the submission deadline will be used for grading purposes. If the student had not submitted any code, then, in this case, there are no points awarded to the student.

4. If a student needs any clarification on their lab grade, it is strongly recommended to talk to the Professor. The lab grade may be changed if deemed appropriate.