·**Author/Authors:** Tirtha Subedi

·**Motivation**: Our motivation for choosing this program is that we wanted to expand the version of project we did in C++ using gui. We also wanted to create an input boxes and implement images. To get more hands on using GUI, we wanted to challenge this project on Turtle. And we were curious about writing the program that can find age of the user and the astrological sign of the user using one time input.

·**Purpose**: Our software is an enhance of A19 which was originally in C++, but we convert to python and added more features. Our program first ask user for their date of birth, then it will check whether the date of birth user enter is correct or not. There is a method to store the date of birth of the user. It will take the user input then calculate user age by their given date of birth, then it calculate the zodiac or astrological, so after calculating it will display the correct zodiac or astrological sign on the window of GUI.

·**Audience**: Our application is for anyone’s use. It will help users calculate their age, their zodiac sign, and show an image of what their astrological or zodiac sign looks like.

·**Instructions:** It will ask for 3 separate inputs. Birth day, month, and year, after clicking submit, it will show the outputs. It is very simple to use.

·**Design**: Using CRC cards, pseudocode, and/or a mixture, create a design plan which meets the computational requirements.

-->Check user input, if valid, compare their input to find the correct zodiac sign. Then subtract their birthday from current date to find their age. Print out the results on turtle window.

·**Enhancement**: Briefly describe exactly what enhancements you made in creating this extra credit assignment.

→ We used GUI. Our original program was on textual and on our console. Using Turtle window to expand this program new and implementing images and input boxes as well.

·**Functionality**: A list of the extra credit assignment's primary functionality and characteristics.

→ It’s primary functionality is to calculate age, zodiac sign and display the correct image.

· **Files**: A list in bulleted form of the names of all files submitted (source code and input, etc.)

* finding\_age&astrogical\_sign.py
* [3d\_capricorn.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/3d_capricorn.gif)
* [Aquarius.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/Aquarius.gif)
* [Aries.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/aries.gif)
* [Cancer.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/cancer.gif)
* [Capricorn.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/capricorn.gif)
* [Gemini.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/gemini.gif)
* [Leo.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/leo.gif)
* [Libra.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/libra.gif)
* [Pisces.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/pisces.gif)
* [Sagittarus.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/sagittarus.gif)
* [Scorpio.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/scorpio.gif)
* [Taurus.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/taurus.gif)
* [Virgo.gif](https://github.com/CSC236-S18/aec-extracredit-animalguessgame/blob/master/virgo.gif)

*We have one main file and rest are images that are associated with the zodiac sign.*

·**Utilized Data Structures:** As mentioned above, this extra credit assignment must effectively use classes as well as one or more data structures studied in this course. In this section state which data structure(s) you have used, discuss why you believe these were appropriate choices, what alternate structures you might have used in its place, and why you made the choices you did.

→ We had high interest in implementing GUI for our final project. So we decided to write everything inside a Class. And we wanted to store all the input data from users. Therefore, we used a list to store user’s input. List was used because if later on, we want to compare our users on what age group used our app the most, it will be easier to access. We would check 3 items at a time that is inside the list (day, month, year) per user.

·**Big O Analysis:** Choose a section of code which makes important use of a data structure explored in this course. Clearly mark this section of code so that it can be referenced here. Compute and explain Big O analysis on that section of code.

→ We have all the user’s input being stored on a list. To append the inputs in a list it has to do one item at a time. Which takes in O(n) for n items to append.

·**Resources**: Using a formal standard for documentation, list all software resources utilized in the making of your extra credit assignment (platform, language, libraries, tools, etc.,) and describe and how you integrated the ideas or code into your program.

→ We used Turtle library in Python, and everything is inside of class.

·**Challenges**: The challenges that you encountered during this extra credit assignment’s development.

→ One of the biggest challenge we face was to implement images. Most of the image formats didn’t work and we weren’t sure why, but later we use images with format of .gif and it worked.

· **Testing**: A list in bulleted form of all input values or unit tests used for testing. Here you should be careful to select representative input cases, including both representative typical cases as well as extreme cases.

* Birth day
* Birth month
* Birth year
* We also created a function that checks whether all the inputs are valid or not.

·**Errors**: A list in bulleted form of all known errors and deficiencies.

→ All errors have been fixed. If user input is not an integer, it will keep on asking until an input is inserted.

·**Measures and Assessment**: An outline of extra credit assignment measures of success, how well you believe this extra credit assignments meets these measures.

→ We used a lot of data structures in the course. So we decided to use a list. Which saves users input in a list. It was amazing to implement GUI and data structures at the same time. We feel ready to connect data structures to different areas of computing.

· **Summary**: A brief summary description of the design and implementation, including how much your initial design plan evolved, the final result you achieved and the amount of time you spent as a programmer in accomplishing these results. This should be no more than two paragraphs.

→ At first, we had to decide which modules to use to enhance the program that we want to enhance. Then We start by drawing the program on a paper. It took us while to find out the steps that we want to follow on our program then we write down each of the steps (as a planner) then when implementing the program we followed one step at a time. We followed 100% of out initial design plan and it was straightforward. But we have faced lots of error on the process of implementing the program, debugging took more time then actual writing the code.

·**Comments**: A paragraph or so of your own comments on and reactions to this extra credit assignment

→ This extra credit help us connect GUI and data structures in a way we have never done before. It was amazing to see how information from GUI was pulled in a list and store them. It makes me think of real life situations on how computing works. For example, when we play games, it is all visual. However, all the data has to be stored somewhere for it to display and calculate outcome. So this assignment open out mind to implement data structures for real life problems.