BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter

(Individual Work)

Student Information:	Surname	Given Names	Student ID Number
1.	**Jusman**	**Christy Natalia**	**2301890365**

Course Code	**: COMP6502**	**Course Name**	**: Introduction to Programming**
Class	**: L1AC**	**Name of Lecturer(s)**	**: Ida Bagus Kerthyayana Manuaba**
Major	**: CS**		
Title of Assignment (if any)	: Quadratic Equations Graph		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 17-01-2020**	**Submission Date**	**: 14-01-2020**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.

2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Christy Natalia Jusman

"Quadratic Equations Graph"

Name : Christy Natalia Jusman

ID : 2301890365

A. Project Specifications

1. Project Description

As the final project, I have made a program that can make a graphic for Quadratic equations. This program aimed to help its user to make the graph from the algebra quadratic equations without calculating things in the paper. Usually, students have to make a graph by themselves. By using this program, the users only have to input the value of A,B, and C. This program also can help them to check their work if it is true or not to prevent from human error on counting.

The standard of quadratic function form can be written as: $ax^2 + bx + c$. Then the graph of the function is a curve called parabola. Parabolas may open upward or downward and vary in "width" or "steepness", but they all have the same basic "U" shape. Using Matplotlib, the program creates an output for the quadratic equations graph.

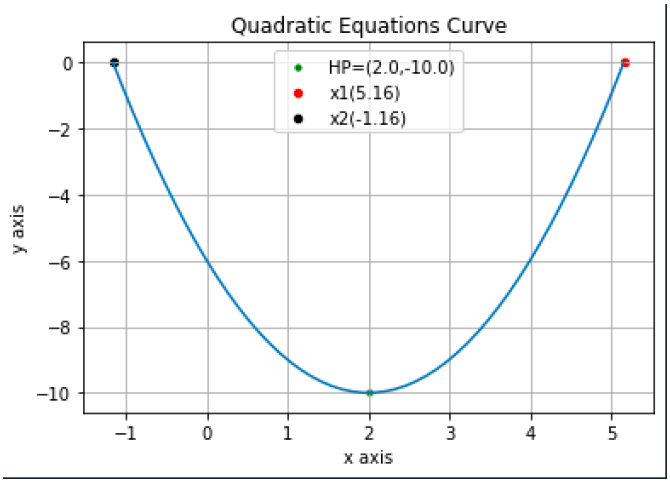
B. Solution Design

1. Design of the program

The design of this program is very simple, so many people can understand on how to use this program. Therefore, this program was built to help people especially student to make a graph. This program provides the value of discriminant, the highest

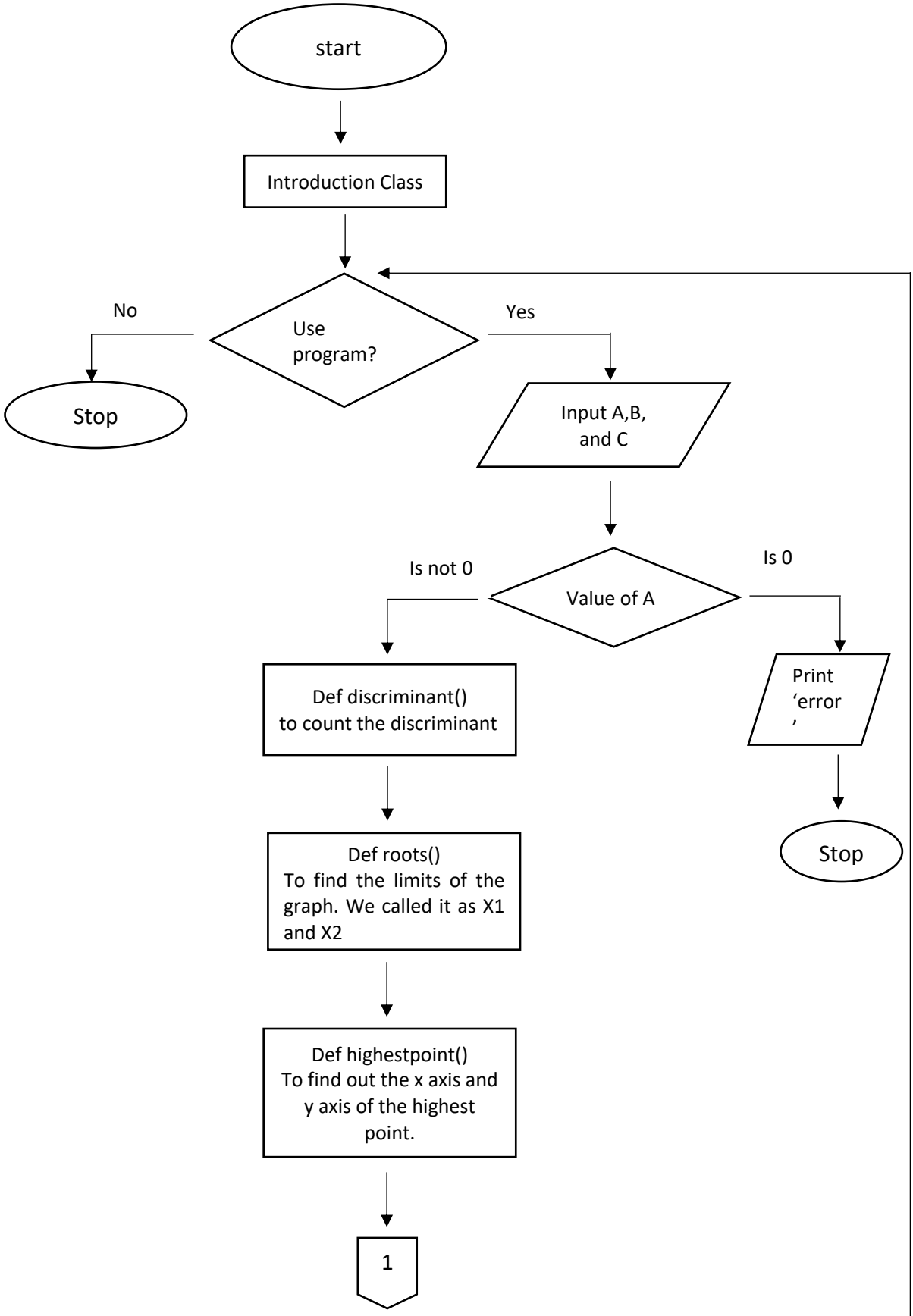
point, value of X_1 and X_2, conclusion and the graph. Not only that, there is also a loop function in this program. Every time you had done with the graph, it will ask the user to continue using this program or close this program. This is really efficient rather than the users have to re-run the program. It also has a history features. It will save the value of A, B, and C that we already used before.

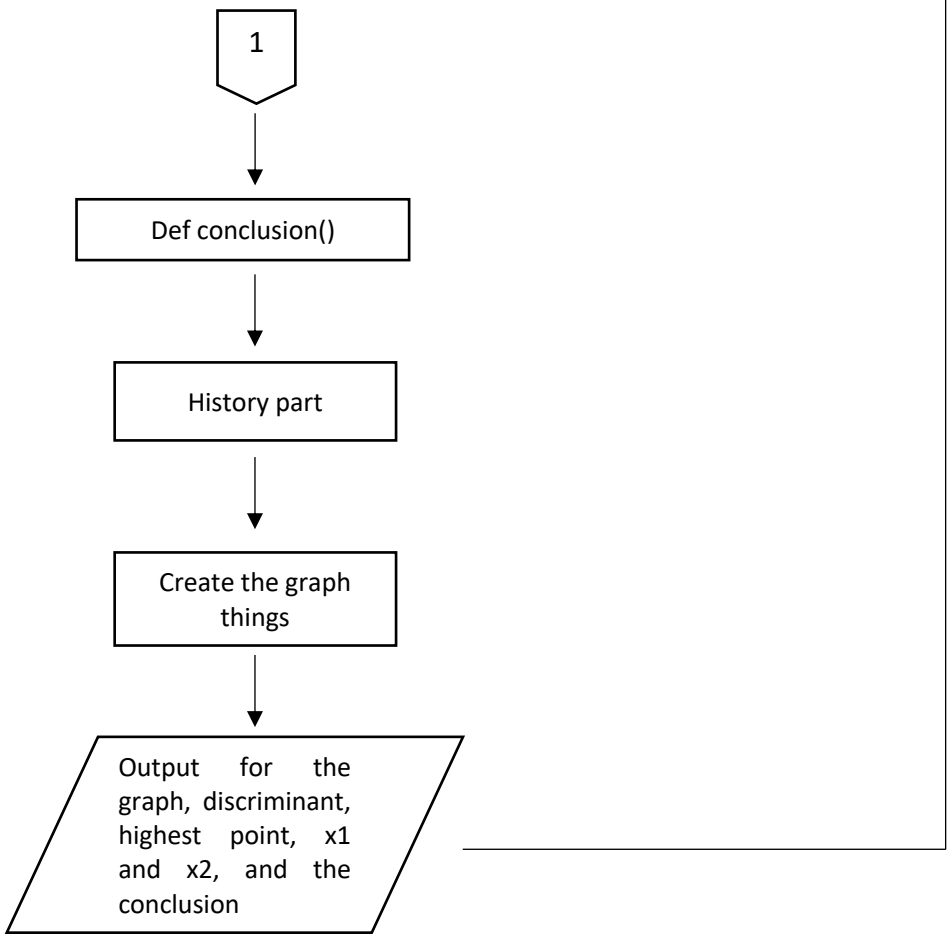
In the beginning, there is an information about things that this program can do. The final looks of the graph has 3 important dots. The green dot shows the highest/lowest point of the graph. The black dot shows the lowest range for the x-axis while the red dot shows the highest range for the x-axis. Each dots has a label to show up the coordinate point.



C. How it Works

1. PROJECT CHART





2. CLASS DIAGRAM

Introduction	Data
-InputA -InputB -InputC	+intro
-newInputA() -newInputB() -newInputC()	+programIntro()

3. **Explanation of Each Function**

introduction.py

- class Introduction():
    - __init__(self):
        - Print 'This program can make a quadratic equations graph' text.
        - Ask the users to input the answer - do they want to use this program or not and then assigned the answer as intro variable.

    - programintro(self):
        - Assigns the intro variable to this function and then return it.

inputnumbers.py

- class Data():
    - __init__(self):
        - Ask the user to input the value of A and assigns it as inputA
        - Ask the user to input the value of B and assigns it as inputB
        - Ask the user to input the value of C and assigns it as inputC
    - newInputA(self):
        - Put the value of inputA on newInputA function and then return it.
    - newInputB(self):
        - Put the value of inputB on newInputB function and then return it.
    - newInputC(self):
        - Put the value of inputC on newInputC function and then return it.

main.py

- It will loops everything in here after we are done on making the graph.
- Creates a new variable called intro and assigns Introduction class from *introduction.py* as intro. Also creates a new variable called intromsg for the programintro function.
- If intromsg is yes, it will ask the users to input the value of A,B, and C. But, while the intro variable value is 'no' it will exit this program. Importing sys library then using it to exit the program.
- Create variables called valueA, valueB, and valueC and assign each values from newInputA, newInputB, newInputC functions on it.
- If the valueA is not 0, then it will be proceed to discriminant function. But, if the valueA is 0, it will turn into the error.
    - def discriminant()
        - Creates a new variable called discriminantValue and make it as a global variable. Count the discriminant. Then, assign the result as discriminantValue.

        - The value of discriminant will be printed.

    - def roots()
        - If the discriminantValue is smaller than 0, we can't make the graph because they don't have any roots for it.
        - If discriminantValue is greater than 0, creates a new variable called roots to find out the roots of discriminantValue. Importing sqrt from math library to get the roots of it.
        - Creates new variables called X_1 and X_2 to get the range for x-axis.
        - The result will be printed
    - def highestpoint()
        - Creates 2 new variables called highx and highy and make both of them as a global variable.
        - Using the math formulas to get the highest point x-axis's coordinate and then assign it as highx.
        - Using the math formulas to get the highest point y-axis's coordinate and then assign it as highy.
        - The result will be printed
    - def conclusion()
        - It will print the conclusion based on the data and regulations that apply on quadratic equations graphic.
        - The result will be printed
    - def printHistory()
        - Append each of the values to the historyList.
        - Do the looping to count the length of the list we already used this program .
        - Using the format function to replace the first {} with the i, and the second {} for the value at the list number on i.
        - The result will be printed

OUTSIDE THE FUNCTIONS

- Using x1 and x2 variables from roots functions. X2 will be the starting point, while X1 is the stop point. At this time, 100 means there are 100 items that

are on experiment based on the range from x2 until x1. As much as the items, so the curve will be look like as a parabola.

```
x=numpy.linspace(float(x2),float(x1),100);
```

- For the Y-axis, the start and stop point will be based on the x-axis. Just use the mathematics formula to find the y.
- Importing pyplot from matplotlib library and then assign it as plt. The plt function is to make the dots for the important points. We have to make the coordinate of x-axis into float because if we didn't change it, the coordinate will be rounded and it's not as same as the result on the highestpoint functions. We have to do that on both x-axis and y-axis. Using the label to show up the dots coordinate.

```
plt.scatter(float(highx),float(highy),s=20, c= "green",label
= 'HP=' + '('+ str(highx) + ',' + str(highy)+')')
```

- Using the pyplot.show() and plt.show() to show the graph.

D. Evidence of Working Program

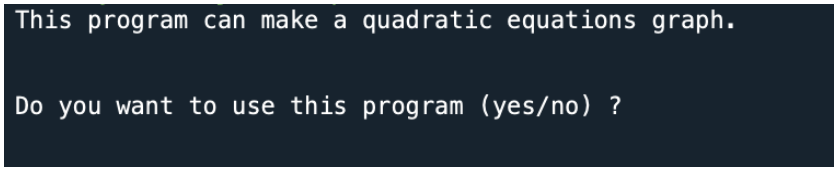


Figure 1.1

is the User Interface that will be shown once *main.py* is run.

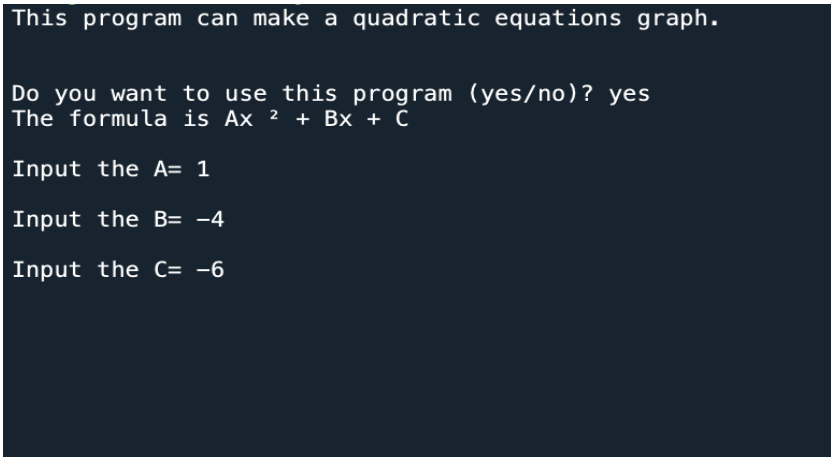


Figure 1.2

If the answer is yes, the user has to input A,B, and C

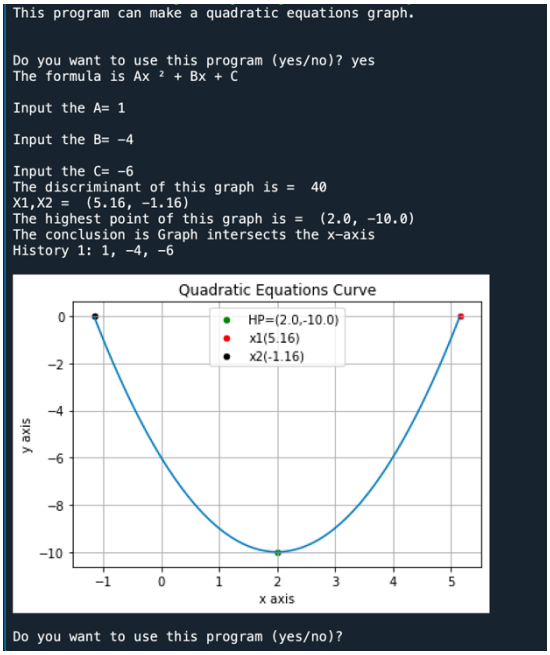


Figure 1.3

Then it will show the graph, the discriminant, the coordinate of the highest point, value of the start and the stop point, conclusion and also the history. It will ask the user, do they want to use this program again or not.

```
Do you want to use this program (yes/no)? yes
The formula is Ax ² + Bx + C

Input the A= 0

Input the B= 3

Input the C= 5
You can't assign 0 as A. Please try again.

Do you want to use this program (yes/no)?
```

Figure 1.4

If the value of A is 0, it will turn into an error. It will loop to the beginning.

```
Do you want to use this program (yes/no)? no
Thankyou for using this program. See you next time!
```

Figure 1.5

If the answer is no, then this program will be closed.

E. Conclusion

Even though the code for this program is very simple, but this program is very useful for students. Next time, i have to make some updates for this program. In other way, we can use this program too to count for the straight line equation.

Source :

http://dl.uncw.edu/digilib/Mathematics/Algebra/mat111hb/PandR/quadratic/quadratic.html (for the math things)