**ENGR 102 Sect 508 Lab 3b**

**100 +5 points**

**Reading assignment:**

|  |  |
| --- | --- |
| **Lecture Slides** | **L03** |
| **Python pdf handouts,**  **zyBook chapters 3** | **Complete all participation and challenge activities** |

***Attention!!***

***For submission: pdf/word file and all py-files as asked in the assignment. No pictures by the phone – it is impossible to read. You will be allowed to resubmit and reupload HW as many times as you want to within the due date/time, only last submission will be graded. No late submissions.***

***For submission you may use this file as template: rename file including your name. Do not forget to put your name inside of this file as well. If it is a team work, include the team number and all team members.***

**Activity To Do individually, in lab or outside**

***For this submission use Individual Header. Submit word file and py-files on eCampus.***

You may talk with others in lab about how to go about doing each of assignments. However, submission should be individual

**Problem 1 [10 pts]** Find the maximum and minimum values that can be stored in a 2-byte integer variable.

The minimum value would be -2^16-1= -2^15=-32768

The maximum value would be 2^(16-1)-1=2^(15)-1=32767

**Problem 2 [10]**

Can a 4-byte variable of the real data type be used to store larger number than 4-byte variable of the integer data type? Why or why not?

Yes you can, because float numbers can store larger numbers than integers, because they include exponents.

**Problem 3 [10]**

A second type of floating point data – is *Double precision*. A double precision number usually occupies 8 bytes (64 bits), instead of the 4 bytes occupied by real number. If 53 bits are used for mantissa and 11 bits are used for exponent. How many significant digits does double –precision value have? What is the range of double precision numbers?

A double precision number has a maximum of 15 significant digits.

The lowest number in double precision floating point data is: 10^-308

The highest number in double precision floating point data is: 10^308

**Program 1: [20 points]**

This assignment is meant to give you practice with the request information-gather input-perform computation-output result program structure. Your code should read all needed data from the user, and output the answer.

1. Your input statements should provide a clear prompt asking the user for information.
2. Your output should be descriptive of what the result is.
3. You can make these ad separate programs, (e.g. labeled as 1a, 1b, 1c, 1d).
4. If you know how make 1 program with options to calculate 1 or 2 or 3

Calculate

**[10 points]**

1. The period of an oscillating pendulum

 The equation for period of an oscillating pendulum

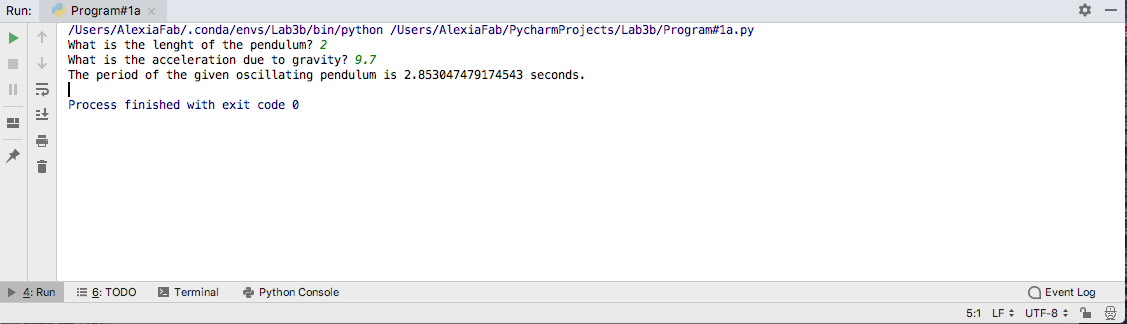
Where T is period of an oscillating pendulum

L –is the length of the pendulum (given)

g is the acceleration due to gravity

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: Alexia Perez  
# Section: 508  
# Assignment: Lab 3b  
# Date: 13-09-2018***import** numpy  
**from** math **import** \*  
  
*#This program solves a specific physics equation using the user's input.  
  
#1. The period of an oscillating pendulum  
# The equation for period of an oscillating pendulum: 2pi\*sqrt(L/g)  
# Where T is period of an oscillating pendulum  
# L is the length of the pendulum (given)  
# g is the acceleration due to gravity*L = float(input(**"What is the lenght of the pendulum? "**))  
g = float(input(**"What is the acceleration due to gravity? "**))  
T = 2\*pi\*sqrt(L/g)  
print(**"%s %s %s"** % (**"The period of the given oscillating pendulum is"**, T, **"seconds."**))

Output:



**[10 points]**

1. Calculate the damped sinusoidal oscillation

 The equation for damped sinusoidal oscillation

Where Vm is the maximum value of oscillation

α is the exponential damping factor

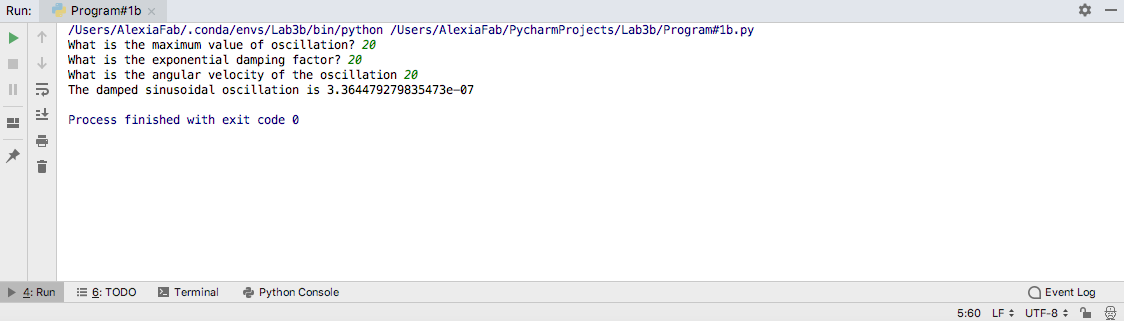
ω is angular velocity of the oscillation

In the word file put a couple of sentences, explaining what is damped sinusoidal oscillation? Sine waves describe many oscillating phenomena. When the wave is damped, each successive peak decreases as time goes on. A damped sine wave is a sinusoidal function whose amplitude approaches zero as time increases.

(https://en.wikipedia.org/wiki/Damped\_sine\_wave)

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: Alexia Perez  
# Section: 508  
# Assignment: Lab 3b  
# Date: 13-09-2018***import** numpy  
**from** math **import** \*  
  
*# This program calculates the damped sinusoidal oscillation  
# The equation for damped sinusoidal oscillation: v(t) = Vm\*e^(-α)\*cos(ω)  
# Where Vm is the maximum value of oscillation  
# α is the exponential damping factor  
# ω is angular velocity of the oscillation*Vm = float(input(**"What is the maximum value of oscillation? "**))  
a = float(input(**"What is the exponential damping factor? "**))  
w = float(input(**"What is the angular velocity of the oscillation "**))  
vt = Vm\*a\*cos(w)\*exp(-a)  
print(**"%s %s "** % (**"The damped sinusoidal oscillation is"**, vt))

Output:



**Program 2:[50 points]**

This program is meant to help you get practice with writing programs to perform more complex numerical calculations, and specifically to give you practice with vector calculations.

Write a program that calculates the angle between two points, as seen by an observer.

Your program should read in:

* Point 1 (x1,y1,z1): The 3D position of an observer
* Point 2 (x2,y2,z2): The 3D position of the first observed point
* Point 3 (x3,y3,z3): The 3D position of the second observed point

Then, it should calculate and output, *in degrees*, the angle between the points from the viewer’s perspective. The steps for doing this are:

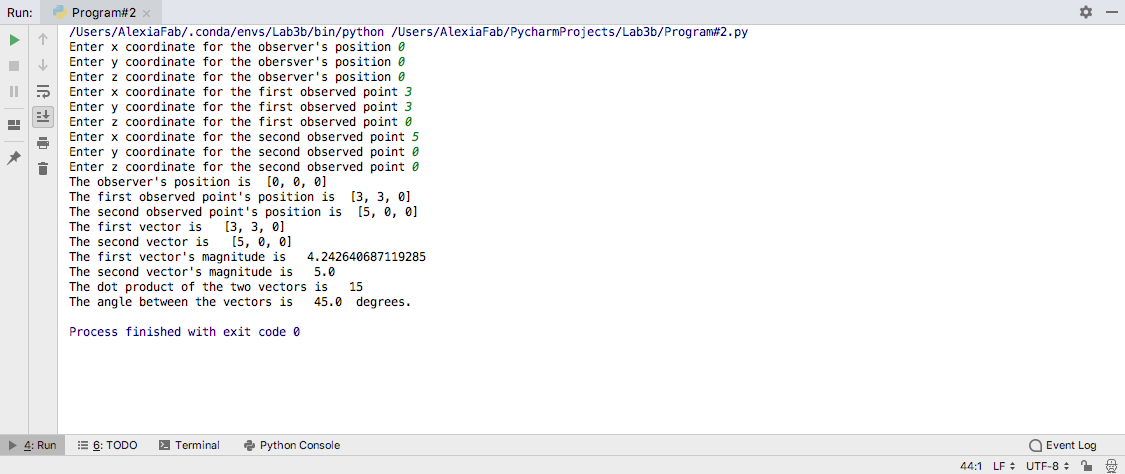
* [5] Read in the points from the user. Assume the points will be some (x, y, z) coordinates.
* [10] Calculate the two vectors from the observer to each of the observed points
* [10] Normalize the vectors
* [10] Calculate the dot product between the vectors
* [10] Use that to calculate the angle between the two observed points. Note that the dot product of two normalized vectors gives the cosine of the angle between those vectors.
* [5] Outputs the answer in degrees

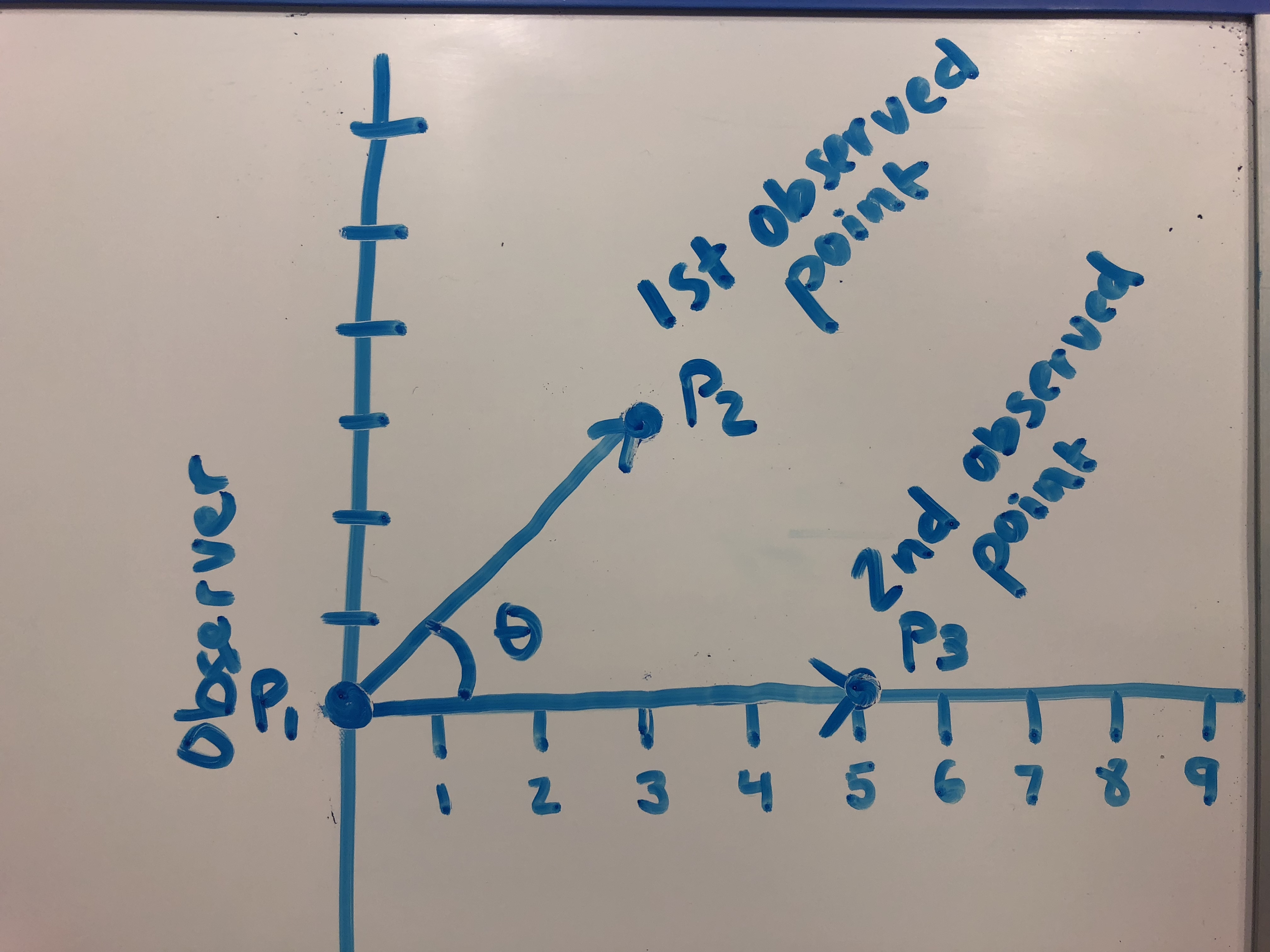
Before beginning coding, you should stop and think briefly about the variables you will need for your program.

As output for this problem I also need you to include sketch in the word file. You may do it by hand and attach it as picture. No plotting in the program

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: Alexia Perez  
# Section: 508  
# Assignment: Lab 3b  
# Date: 13-09-2018***import** numpy  
**from** math **import** \*  
  
*#Get observer's coordinates:*obsX= int(input(**"Enter x coordinate for the observer's position "**))  
obsY= int(input(**"Enter y coordinate for the obersver's position "**))  
obsZ =int(input(**"Enter z coordinate for the observer's position "**))  
Point1 = [obsX,obsY,obsZ]  
  
*#Get first observed point's coordinates*x1= int(input(**"Enter x coordinate for the first observed point "**))  
y1= int(input(**"Enter y coordinate for the first observed point "**))  
z1 =int(input(**"Enter z coordinate for the first observed point "**))  
Point2 = [x1,y1,z1]  
  
*#Get second observed point's coordinates*x2= int(input(**"Enter x coordinate for the second observed point "**))  
y2= int(input(**"Enter y coordinate for the second observed point "**))  
z2 =int(input(**"Enter z coordinate for the second observed point "**))  
Point3 = [x2,y2,z2]  
  
*# Get vectors between the observer's position and the observed points*vector1 = [x1-obsX, y1-obsY, z1-obsZ]  
vector2 = [x2-obsX, y2-obsY, z2-obsZ]  
v1x= vector1[0]  
v1y= vector1[1]  
v1z= vector1[2]  
v2x= vector2[0]  
v2y= vector2[1]  
v2z= vector2[2]  
  
*#Get the magnitude of each vector*v1\_magnitude=float(sqrt(v1x\*\*2+v1y\*\*2+v1z\*\*2))  
v2\_magnitude=float(sqrt(v2x\*\*2+v2y\*\*2+v2z\*\*2))  
  
*# dot product between the 2 vectors*dot\_prod=(v1x\*v2x+v1y\*v2y+v1z\*v2z)  
  
*#angle between the 2 vectors (in degrees)*angle= degrees(acos(dot\_prod/(v1\_magnitude\*v2\_magnitude)))  
  
*#Output*print(**"%s %s"** % (**"The observer's position is "**, Point1))  
print(**"%s %s"** % (**"The first observed point's position is "**, Point2))  
print(**"%s %s"** % (**"The second observed point's position is "**, Point3))  
print(**"%s %s"** % (**"The first vector is "**, vector1))  
print(**"%s %s"** % (**"The second vector is "**, vector2))  
print(**"%s %s"** % (**"The first vector's magnitude is "**, v1\_magnitude))  
print(**"%s %s"** % (**"The second vector's magnitude is "**, v2\_magnitude))  
print(**"%s %s"** % (**"The dot product of the two vectors is "**, dot\_prod))  
print(**"%s %s %s"** % (**"The angle between the vectors is "**, angle, **" degrees."**))

Output:





**[5] Challenge Program (Optional):**

Using *only the commands we have covered in class*, write a program that asks a user for a number of digits of precision, and prints the value 1/7 rounded to that many digits of precision.