The purpose of this lab is to gain experience with basic Python data types, variables, operators, expressions, statements, basic I/O operations and built-in functions.

The completed lab must be submitted via Blackboard NLT (no later than) the beginning time of the lab session immediately subsequent to the lab section in which it is assigned. If you have questions, use the Blackboard discussion forums (and instructor/TA office hours) to obtain assistance.

Lab Requirements:

- The source code for this lab must be submitted in a file named using the following format:

  Firstname + underscore + lastname +underscore + labSectionNumber + underscore + lab + PANumber dot py

  Example: Josephine_Student_2xx_PA2.py

- The source code file must contain a **file header** formatted as specified below. The header information must be complete and accurate.

- The source code file should use **self-documenting code** and additional comments (as required) to improve code readability.

Source File Header:

```
#------------------------------------------------------------------------------
# source_file_name.py
# Student Name: Joe Student
# Assignment: Lab #N
# Submission Date: 09/09/9999
#------------------------------------------------------------------------------
# Honor Code Statement: I received no assistance on this assignment that
#                       violates the ethical guidelines as set forth by the
#                       instructor and the class syllabus.
#------------------------------------------------------------------------------
# References: (list of web sites, texts, and any other resources used)
#------------------------------------------------------------------------------
# Comments: (a note to the grader as to any problems or uncompleted aspects of
#            of the assignment)
#------------------------------------------------------------------------------
# Pseudocode: Provided in spec
#------------------------------------------------------------------------------
# NOTE: width of source code should be < 80 characters to facilitate printing
#------------------------------------------------------------------------------
```

Background:

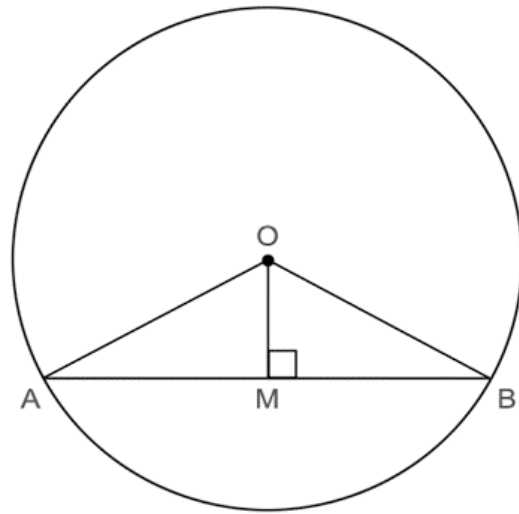### *Chord of a circle and angle subtended at the centre -*

      A **chord** is a line segment joining two points on the circumference of a circle. In the figure shown below, **AB** is a chord. The length of the longest chord of a circle or radius r is 2r (diameter of the circle). When you join the end points of the chord to the centre of the circle, it forms an isosceles triangle ($\triangle$**AOB** in the figure). The interior angle at the vertex at the centre of the circle of the isosceles triangle is the angle **subtended** by the chord ($\angle$AOB in figure). Since $\triangle$**AOB** is an isosceles triangle the perpendicular of the chord to the centre bisects the chord as well as the angle subtended at the centre.

$\angle$AOM = $\angle$MOB = ½ * ($\angle$AOB)
AM = MB = ½ * (AB)

Major and Minor Arcs –

      Any chord bisects the circumference of a circle to form two arcs, the longer of these two arcs is called the Major Arc and the smaller is known as the Minor Arc.

Lab Scenario:

      For this lab assignment, you are being asked to create a Python program which requests the user to input the length of the chord of a circle of unit radius (radius of length 1 unit) and calculates and outputs the angle subtended by this chord at the centre (in degrees) and the lengths of Major Arc and Minor Arcs formed by the intersection of this chord with the circle.

Lab Procedure:

      As you can see, this problem involves only basic trigonometry and most of you would have no trouble solving this problem, on paper. But, creating a program that solves it for any input might seem a little challenging. Here is one way to approach it
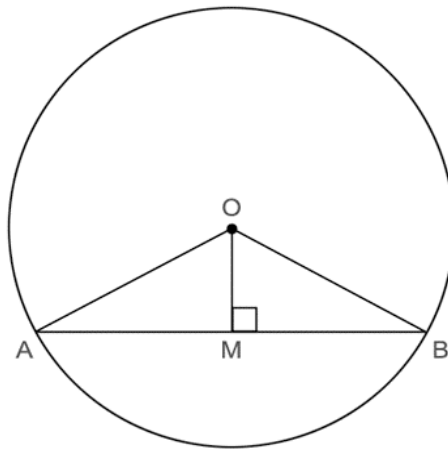
- Carefully read the problem statement and make sure that you understand the problem. List down all the inputs and outputs, along with the units (e.g. degrees, radians, meters, cm, kg. etc.)
- Solve the problem for one set of inputs, on paper and analyze each step. Note down the sequence of steps and think about how would you code these steps in Python. But, don't start coding yet. Try to get the design right first. As you by now know, this sequence of steps is called the pseudo-code.

- Go through each step in your pseudo-code and think of a way to translate it into a statement/expression in Python. You will have to be a little creative here. Here are some tips:

    o If a step in the pseudo-code does not map to a single python statement/expression, then break it down into smaller steps.

    o Also think about the data-types of your variables. Do you remember what happens to the fractional part of the result, when you divide two integers?

    o Read the documentation of the functions that you will be using.

    o Go through the tips provided in Programming Assignment #1

- Test and verify your solution.

    Since this is not a math class, an appendix at the end provides the formulas that you will need for solving this problem. Unless you are having trouble understanding the problem, we recommend that develop your own solution without using the appendix.

Grading Rubric:

	Excellent	Average	Needs Improving	Points
Submission Details	Both file name and file header meet stated spec.	Either file name is incorrect or file header is missing sections/details	Both file name and file header is missing or are incorrectly implemented.	__/2
Pseudo code	Pseudocode is present and provides sufficient detail for a developer to accurately implement the process.	Pseudocode is present, but provides insufficient detail for a developer to completely implement the process.	Pseudocode is missing or is so general that it provides little assistance in implementing the process.	__/3
Comments & Self Documenting Code	Comments clearly demonstrate which sections of code are related to specific steps in the pseudocode and variable names clearly reflect what they represent.	Comments generally demonstrate which sections of code are related to specific steps in the pseudocode and variable names generally reflect what they represent.	Comments are missing or provide little assistance to the understanding of the code and/or variable naming conventions provide no insight into the referenced data.	__/3
Formula Implementation	Formula is implemented correctly, including any required data transformations.	Formula is generally implemented correctly, but contains minor flaws or required data transformations are not implemented.	Formula is significantly flawed, and required data transformations are not implemented.	__/5
I/O	All specified input and output are implemented.	All specified standard input and output are generally implemented with minor flaws.	Specified standard input and output are not implemented.	__/2
Overall				__/15

Appendix:

AM = ½ * (AB)
AO = r
$\angle AOM = \sin^{-1} (AM / AO)$
$\angle AOB = 2 * \angle AOM$

Circumference = C = 2 * pi * r
Minor Arc = $(\angle AOB/360) * C$
Major Arc = C - Minor Arc

NOTE:
-   For your program, the circle has **unit radius.**
-   Read the documentation for $\sin^{-1}$(math.asin), what are the units of its return value?