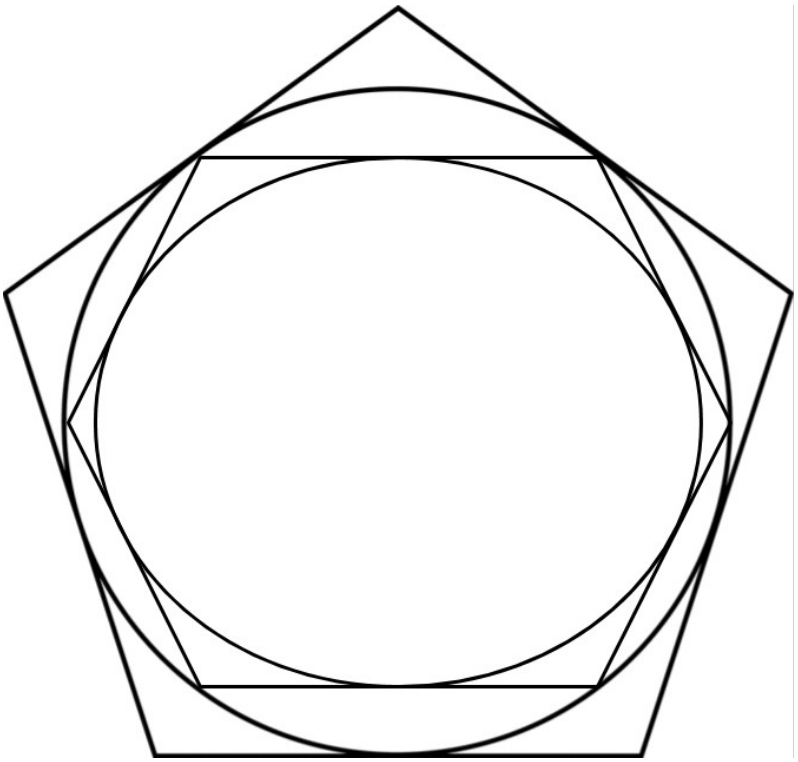
Lab #2 – Programming Assignment
Due: Friday February 5, 2021 at 11:00pm (time local to West Lafayette, IN)
10 Points Possible

Problem: In the image given there is a smaller circle inscribed inside of a regular hexagon. The hexagon is inscribed inside of a larger circle. And finally the larger circle is inscribed inside of a regular pentagon.

Given the radius of the smaller circle, calculate the following:

1. Area of the small circle.
2. Area and side length of the hexagon.
3. Area of the large circle.
4. Area and side length of the pentagon.

Accept input and display ALL output exactly as seen in the example executions that follow.



Example Execution #1:

```
Enter the radius of the small circle -> 10

-==-==========-==========-==-
Area of small circle:        314.16
Hexagon side length:          11.55
Area of hexagon:             346.41
Area of large circle:        418.88
Pentagon side length:         16.78
Area of pentagon:            484.36
-==-==========-==========-==-
```

Example Execution #2:

```
Enter the radius of the small circle -> 17.35

-==-==========-==========-==-
Area of small circle:        945.69
Hexagon side length:          20.03
Area of hexagon:            1042.77
Area of large circle:       1260.92
Pentagon side length:         29.11
Area of pentagon:           1458.04
-==-==========-==========-==-
```

All course programming and documentation standards are in effect for this and each assignment this semester. Please review this document!

Example Execution #3:

```
Enter the radius of the small circle -> 55.79

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Area of small circle:       9778.28
Hexagon side length:          64.42
Area of hexagon:           10782.10
Area of large circle:      13037.71
Pentagon side length:         93.61
Area of pentagon:          15075.87
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Example Execution #4:

```
Enter the radius of the small circle -> 121.21

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Area of small circle:      46155.85
Hexagon side length:         139.96
Area of hexagon:           50894.11
Area of large circle:      61541.14
Pentagon side length:        203.38
Area of pentagon:          71161.76
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Additional Requirements:

1. Add the **lab assignment header** (`vi` shortcut `hlb` while in command mode) to the top of your program. An appropriate description of your program must be included in the assignment header.

> Include the Purdue University e-mail addresses of **each contributing group member** in the assignment header!

2. **Each of the example executions provided for your reference represents a single execution of the program.** Your program must accept input and produce output **exactly** as demonstrated in the example executions, do not add any "bonus" features not demonstrated in the example executions. Your program will be tested with the data seen in the example executions and an unknown number of additional tests making use of **reasonable data.**
   - All floating-point variables must be of the `doubele` type.
   - Use the constant value `M_PI` where needed for the constant pi.

3. Course standards **prohibit** the use of programming concepts beyond the material found in the first three chapters of the book, notes, and lectures to be acceptable for use.
   - The use of the `math.h` library is expected for access to the `sqrt`, `pow`, and trigonometric functions.

4. A program **MUST** compile, be submitted through the `guru` server prior to the posted due date to be considered for partial credit. The C-file you submit must be named exactly: `lab02.c`

Course Programming and Documentation Standards Reminders:

- Indent all code found within the `main` function **exactly** two spaces.
- Place a **single space** between all operators and operands.
- Comment **all** variables to the right of each declaration. Declare only one variable per line.
- Notice that several programs (see program 2-9 on pages 74-75) in the programming text use a single line comment to indicate the start of the local declaration and executable statement sections of the `main` function.
  - At no point during the semester should these two sections ever overlap. You might consider adopting this habit of commenting the start of each section to help you avoid this mistake.
- Select **meaningful identifiers** (names) for all variables in your program.
- Do not single (or double) space the entire program, **use blank lines when appropriate**.
- Maximize your use of symbolic/defined constants and minimize your use of literal constants.