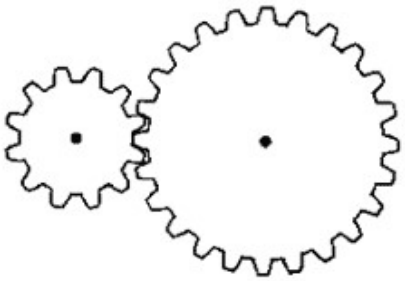


24-780 B—Engineering Computation

Assigned: Wed. Sept. 20, 2021
Due: Tues. Sept. 26, 2023, 11:59pm

Problem Set 4: Basic Gears

In this assignment, you will build a *simple* gear simulator. Don't worry, we will expand the capabilities of the system in future assignments. Note, in order to decrease your time commitment, Gear.h is attached.

Task 1: Develop data model

Gear

First, we need to define a *Gear* class, with the following member variables, all private (by tradition):

```
      std::string partID; // used for identification
      float pitch;        // given in teeth per inch
      int numbTeeth;
      Point2D loc;        // current location of gear {x, y}
      float angle;        // current rotation of gear in degrees
```

Provide the following functions for *Gear*. (all public)

```
      // reads data from a file over-riding any existing data. We'll work on
      // this function in class together.
void load(std::ifstream &inFile);

      // outputs all the data for a gear in format matching attached example
void print(std::ostream &output = std::cout);

      // calculates and return the pitch diameter
float pitchDiam() { return numbTeeth / pitch; };

      // draws the gear on graphics window using simplified geometry as attached
void draw();

      // rotate the gear by given amount
void rotate(float rotAngle) { angle += rotAngle; };
```

You should also include get and set functions as needed like this:

```
std::string getID() { return partID; }
void setID(const std::string &newID) { partID = newID; }
```

The printing should reproduce the input format data like this example:

```
Gear:
      partID: A1_temp
      pitch: 2.0
      numbTeeth: 30
      startX: 27.5
      startY: 46.7
      startAngle: 0.0
Gear End:
```

Task 2: Develop the Viewer

We are taking it a bit slower this week, so I don't want you to get bogged down with little details of user interface. Thus, I am providing you with a draft version of the main() and auxiliary functions that allow you to control the interface. Use this as the starting point for your solution.

We will discuss all the parts in the lecture. I don't want you to think that I can somehow program things that you cannot.

Task 3: Add one more feature

Add **one** more thing that is not otherwise required in the assignment. For example:

- Gears spin all the time
- Gears are filled in, with fancy coloring
- User can move the gears
- Etc.

Just try to come up with **one** more cool thing

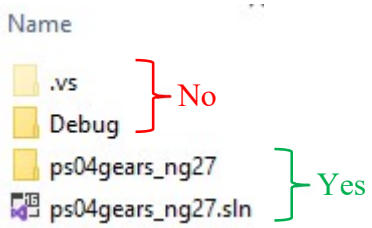
Deliverables

3 files (zipped together):

Gear.h, Gear.cpp << the files for your new class: *Gear* (Gear.h is given, but please edit)
GearViewer.cpp << start with mine and edit to your liking

Upload the zip file to the class Canvas page before the deadline (Tuesday, Sep.26, 11:59pm).

Alternatively, if you are using Visual Studio, it may be easier to submit your entire solution rather than a collection of files. To do this, create a *zip file* of the whole project (the .sln file and the associated folder), being careful NOT to include the hidden folder called ".vs". This folder is used only to manage the IDE and is typically huge (100MB). Erasing or omitting it will just force Visual Studio to rebuild it when needed. The Debug folder should be kept out of the zip file too to avoid including executable files that some firewalls may disallow. *The name of the project should include your AndrewID*



Learning Objectives

Use of classes and objects in C++.

More OpenGL, including transformations

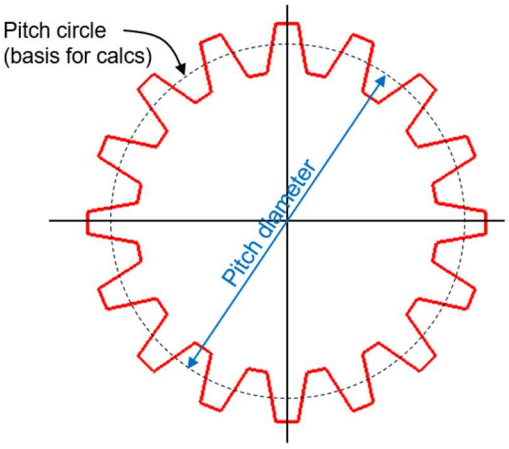
Making use of console input/output

File input and output

Using functions effectively

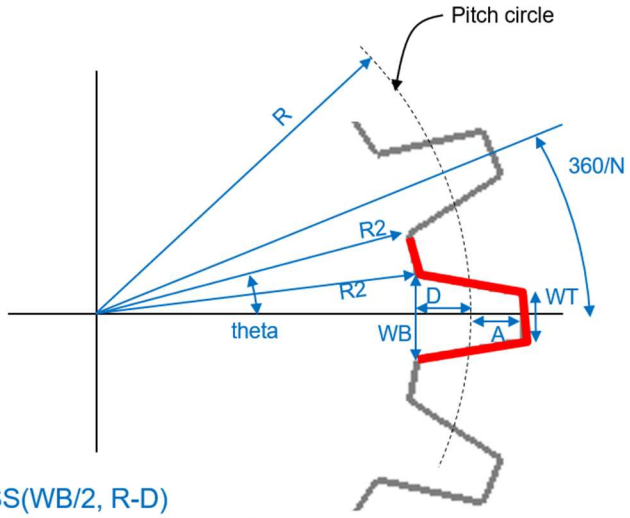
Searching references (online and/or textbook) for C++ library functions.

Gear Geometry



N = number of teeth
P = pitch = teeth per inch of diameter
Pitch diameter = N / P

M = module = 1 / P
D = dedendum = 1.25 * M
A = addendum = 1.0 * M
WB = width at tooth bottom = 1.8 * M
WT = width at tooth top = 1.0 * M



R2 = SRSS(WB/2, R-D)
theta = 360/N – atan(WB/2 / (R-D))