

**This part of the assignment is due by the 20/12.**

**In general always write your name and date of creation as comments in the files and try to be “user friendly” when you write the code:**

- **Add comments on the code to remember yourself and the other what the code does**
- **Manage the possible errors in the input given by the user**

**The 70% of this assignment is everything a part from point 6**

Now we are going to simulate a very simple model of the solar system in which we ignore the interactions among the planets and we assume the planets are just rotating on circumferences around the sun.

1. Create a class Planet that has the data members of name (type string), mass and coordinates on a planet(type double) in km. Create this in its own separate files (.cc and .h) and define member setter and getter functions for each of the data members.
  - a. In your main program, read the file Planets\_data.txt line by line and instantiate a new object of your class for each planet. Test your setter and getter functions to ensure they work as expected.
  - b. Store all your planets in a vector and try to add elements to the vector while you read the file.
  - c. To check that the setter/getter functions work properly iterate on the vector and print the information you stored on each planet
2. Create a class SolarPlanet (.cc and .h) that inherits from the previous one all the functions, constructors and members but contains additional members: time, G (constant), mass of the sun, radius from the sun, angular\_speed.
  - a. Write the constructors and all the functions of this class making sure you use as much as possible of the class previously defined.
3. Add a new function to the class, Evolve(double dt), which will change the coordinates of your planets using the interval of time dt. To calculate the new coordinates just consider that the trajectory of the planets depends only on the distance from the sun, the sun mass ( $1.9890 \times 10^{30}$ ) and  $G(6.673 \times 10^{-20})$ .
4. Now you can evolve all your planets in main. For each planet evaluate the new x and y coordinates of the planet after a fixed interval of time dt till a certain time total\_t, these two quantities need to be provided from the user as a command line arguments
5. The output should be a new file for each Planet (named after the planet) that contains a line for each step in the evolution detailing the time from the beginning and the x and y coordinates of the Planet. Check that your program works in producing circular orbits for the planets (if you save the file as .csv you can use excel)
6. Use your program to find the orbit times of all the planets and print the result on the terminal for each planet.