SOLAR SYSTEM

Team Members:

|  |  |
| --- | --- |
| **Name** | **ID** |
| Ansar Ausama A. A. Yousif | 42110161 |
| Rawan ElLsayed Hashim Ibrahim | 42110142 |
| Salma Abdrabo Abdelbary Abdrabo | 42110089 |
| Alaa Hussein Ibrahim Saad | 42210367 |
| Reem Ayman Abdelfattah Issa | 42110099 |
| Esraa Ahmed Saeed Abdulsamad | 42110123 |

Libraries:

1. Iostream :

* The main library ,it’s responsible for taking inputs from the user and printing outputs for them.

1. fstream:

* Responsible for file operations such as reading, and writing.

1. string:

* Responsible for operations on strings.

1. cmath:

* Responsible for some complex mathematical calculations.

Interface:

The menus are as follows: 1- The rocky planets, 2- The gaseous planets, and 3- The sun.

If the user chose any number other than the previously mentioned numbers, there will be an error message demanding another input. This will continue until the user enters a number from the specified ones.

If the user entered number 1, the menu will switch to the rocky planets’ menu. Then they are asked to enter a rocky planet’s name.

Next, there will be a for loop that checks every index in the rocky planets’ array for the name of the entered planet. If found, all info will be displayed. Then, inside the loop, there will be a function that checks whether the entered planet has moons or not, if yes, the user will be asked if they wanted these info displayed, if yes, the info about all the moons that exist will be displayed (keep in mind that the array that contains info about the moons of a planet contains only 2 moons as a maximum number for both the rocky planets and the gaseous ones). Then exit the loop and the menu.

If the user entered number 2, the menu will switch to the gaseous planets’ menu. Then they are asked to enter a gaseous planet’s name.

Then again, there will be a loop that checks every index for the entered planet’s name. If found, all information about the planet will be displayed. And unlike the rocky planets, there will not be any checking for whether these planets have moons, because they all have a lot of moons. The user will be asked if they want to display info about the moons. If yes, there will be a loop to display the 2 moons of the planet, except if it was for planet Neptune, because we entered the info about only one moon (we couldn’t find much info about the other moons). Then exit the loop and the menu.

If the user entered number 3, the menu will switch to the sun menu, and all info will be displayed, including the distance between the sun and all the planets. Then exit the menu.

Afterwards, the user will be asked whether they wanted to exit the program or not, if no, the system will repeat everything again. If no, the program will close with a thanking messag.

Classes:

1. Planet class: (Inherit class from moon class)

* This class contain 17 functions:

1. I made a default constructor that have no parameters that assign attributes (0).
2. I mad set\_planet\_name \_ method (it’s tybe is void it has string parameter ) To set value to the planet\_name attribute.
3. I made get\_palnet\_name \_ method ( it’s tybe is string it has no parameter )To return the variable (name ) value
4. I mad set\_num\_of moons method (it’s tybe is void it has int parameter ) To set value to the num\_of\_moons\_ attribute.
5. I made get\_num\_of\_moons\_ method ( it’s tybe is int it has no parameter )To return the variable (num\_moons) value
6. I mad set\_planet\_num\_ method (it’s tybe is void it has int parameter ) To set value to the\_ planet \_num \_attribute.
7. I made get\_planet\_num\_ method ( it’s tybe is int it has no parameter)To return the variable (num) value
8. I mad set\_Tybe\_method (it’s tybe is void it has string parameter ) To set value to the\_ tybe\_ attribute.
9. I made get\_Tybe\_ method ( it’s tybe is string it has no parameter )To return the variable (tybe) value
10. I mad set\_time\_ method (it’s tybe is void it has int parameter ) To set value to the time attribute.
11. I made get\_time\_ method ( it’s tybe is int it has no parameter )To return the variable (Time) value
12. Then I made an array called moon [2].
13. I made search method (it’s tybe is bool that have string name parameter) to check If planet namet == the name that user enter it will return true.
14. I made search \_planet \_num method ( it’s tybe is int that have string name parameter)
15. To check if planet name == the name that user enter it will return the num of the palnet in solar system .
16. Moon class:

* This class contain 15 functions:

1. set\_around\_planet method (it’s type is void it has string parameter ) To set value to the around\_planet attribute.
2. get\_around\_planet method ( it’s type is string it has no parameter )

To return th variable (ap) value

1. set\_moon\_type method (it’s type is void it has string parameter )

To set value to the moon\_tybe attribute.

1. get\_moon\_tybe method ( it’s type is string it has no parameter )

To return th variable (mt) value

1. set\_moon\_name method (it’s type is void it has string parameter ) To set value to the moon\_name\_ attribute.
2. get\_moon\_name\_method ( it’s type is string it has no parameter )To return the variable (mn) value
3. set\_radius\_method (it’s type is void it has float parameter ) To set value to the radius \_attribute.
4. get\_radius\_method ( it’s type is float it has no parameter )

To return the variable (r) value

1. set\_mass\_method (it’s type is void it has float parameter ) To set value to the radius \_attribute.
2. set\_mass\_method (it’s type is void it has float parameter ) To set value to the radius \_attribute.
3. get\_mass\_method ( it’s type is float it has no parameter )To return the variable (m) value
4. set\_denisty\_method (it’s type is void it has float parameter ) To set value to the radious \_attribute.
5. get\_denisty\_method ( it’s type is float it has no parameter )To return the variable (d) value
6. set\_gravity\_method (it’s tybe is void it has float parameter ) To set value to the radious \_attribute.
7. get\_gravity\_method ( it’s tybe is float it has no parameter )To return the variable  (g) value
8. Sun class:

* This class contain 9 functions:

1. distance\_from\_sun method (it's type is float and it has

parameter) to set the general equation of the distance from the Sun

1. Distance\_of\_planets\_from\_Sun method (it's type is void and it has no parameters) to \*calculate distance between the sun and each planet in Km. \*
2. get\_Temperature\_on\_surface method (it's type is integer and it has no parameters) to retrieve the variable (temp\_surface) value.
3. get\_Temperature\_of\_core method :

(it's type is integer and it has no parameters) to retrieve the variable (temp\_core) value.

1. Mass\_of\_the\_Sun method :

(it's type is double and it has no parameters) to \*calculate the mass of the Sun in Kg by using the mass of the Earth. \*

1. Radious\_of\_the\_Sun method :

(it's type is integer and it has no parameters ) to \*calculate the mass of the Sun in Km by using the radius of the Earth .\*

1. overloading method that's Called Elements\_of\_Sun :

(it's type is void and has 2 parameters at once and 4 parameters at other once) to provide different semantics for the function,

depending on number of its arguments (as I made different number of arguments each time ).

* The Files functionality in the project:

Since Files functionality is quite limited and there isn’t much for it to do.

Files here works with the parent and child functions it takes their output and

print it into a text document named “planets.txt”

* Main method:

It is where the interface that will appear to the user is written using the classes and functions that were created before, so that the user can deal with the program easily. First off, we make an array of objects for rocky planets, from the planet class, then we fill in all the required information for every planet along with its moons. Next, we make another one for the gaseous planets, with the same steps as the rocky planets.

Then we make an object for Sun. There is no need for filling in any info, as the sun class already has the info within it.

Then we start with the user interface.

First the welcoming to the system, then ask the user to choose a menu from three ones, other than the exit.