Create a class **Circle** with the following **private** attributes,

|  |
| --- |
| **Id:** an integer , start from 1  **X:** abscissa of the center  **Y:** ordinate of the center  **Radius:** radius of the circle |

Implement the following methods in your class (You might need to add other methods if necessary):

* Constructor(s): as needed.
* setX(float x) , setY(float y) , setRadius(float radius): these functions would set the abscissa of the center , ordinate of the center and the radius of the circle respectively.
* getArea() : this would return the area of the circle.
* isInside(circle c1) : this would return a Boolean value. It would check whether circle c1 is completely inside of this circle.
* printInfo(): prints the id , x, y and radius of the circle.

You have to create a utility menu in the console to run these operations (in infinite loop). For example:

* i 1 3 5 : Create a circle with the current unused id , center (1,3) and radius 5. After each creation print the information of the circle. The first created circle will have id 1. The circles created after this will have id 2, 3, 4 and so on.
* s y 2 1 : Set the ordinate of the center of the circle with id 2 to 1.
* p 2 : print the information of the circle with id 2.
* s x 1 2 : Set the abscissa of the center of the circle with id 1 to 2.
* s r 2 4 : Set the radius of the circle with id 2 to 4.
* q 2 1 : Output whether the circle with id 1 is completely inside of the circle with id 2.
* a 1 : Output the area of the circle with id 1.

|  |  |
| --- | --- |
| **Sample input** | **Corresponding output** |
| i 1 3 5  i 1 0 8  s y 2 1  p 2  s x 1 2  p 1  q 2 1  q 1 2  s r 2 4  p 2  q 2 1  a 1  a 2  t | Circle ID: 1 X: 1 Y: 3 Radius: 5  Circle ID: 2 X: 1 Y: 0 Radius: 8  Circle ID: 2 X: 1 Y: 1 Radius: 8  Circle ID: 1 X: 2 Y: 3 Radius: 5  Yes  No  Circle ID: 2 X: 1 Y: 1 Radius: 4  No  78.54  50.2656 |