ASSIGNMENT: 1 Advanced Programing Jaswanth Sai K

Subject Code: 19CSE201 Roll No:CH.EN.U4CSE20130

Question:

Create a Circle class with a centre (x,y) and radius r. Use parameterized constructor to initialize the radius and centre of a circle object. Write a copy constructor that copies the values of one circle object to another one. Use a member function to display the values x,y,r of a circle object. Allow a friend function to update the centre and radius of the circle. There should a static variable 'number_of_circles' and a static member function 'update_number_of_circles' which will update the 'number_of_circles' whenever an instance of a circle is created.

Solution:

Aim:

To Create a C++ program For above problem and check whether it is working or not.

Algorithm:

Step 1: Create a class with name Circle

Step 2: Create member variables which are private by default.

Step 3: Create a default constructor to count the number of objects created.

Step 4: Create a parameterized constructor which will have the same name as class which takes arguments and updates the value of the member variables

Step 5: Create a copy constructor which takes object of the class as an argument.

Step 6: Create a void display function which is a member function that prints the values of Circle object.

Step 7: Declare a friend function within the class which will be able to access the private member variables.

Step 8: Update the number_of_circles value in the constructor which will calculate the number of objects created.

Step 9: Update the value of member variables using the friend function.

Step 10: Create objects in the main function.

Step 11: Call the display function to print the values

Step 12: Call the friend function to update the value.

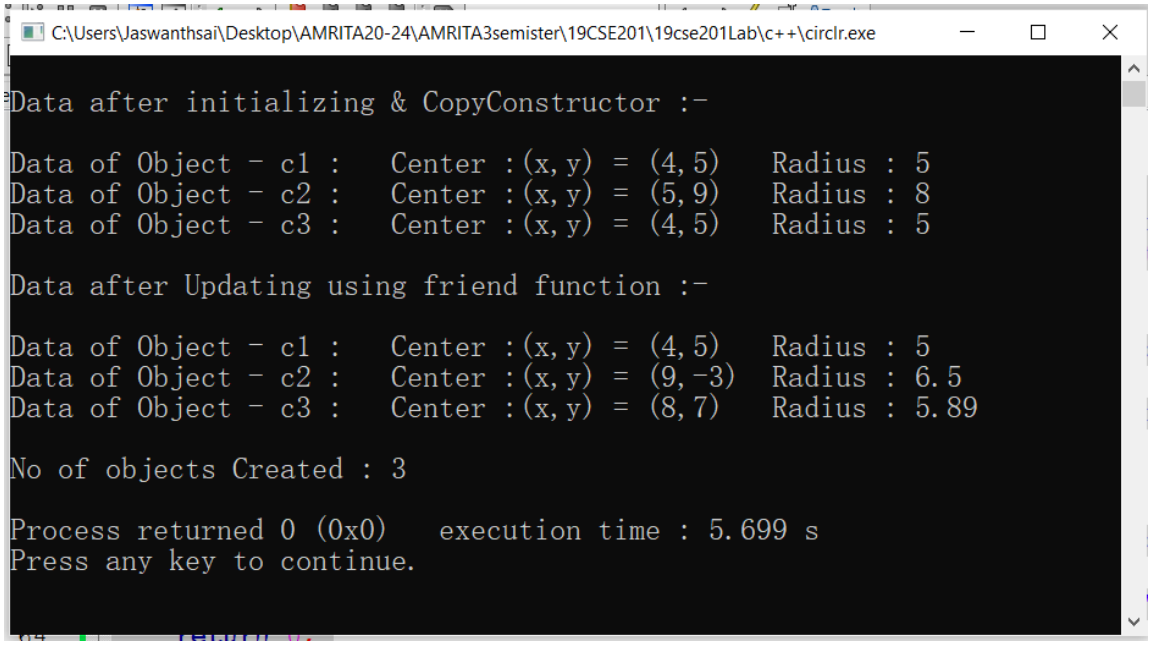
Step 13: Invoke the static members using the class name and the scope resolution operator.

Code:

```
#include<iostream>
using namespace std;
class circle{
  private:
     //private data members.
     float x_corr,y_corr;
     float radius;

  public:
    static int number_of_circles ;  // static data member.
    circle(){
      update_number_of_circle();
    }
    circle(int a,int b,float rad)  // Declaring parameterized Constructor.
    {
       x_corr = a;
       y_corr = b;
       radius = rad;
    }
    circle(circle &copy_circle)  // Declaring Copy Constructor.
    {
       x_corr = copy_circle.x_corr;
       y_corr = copy_circle.y_corr;
       radius = copy_circle.radius;
    }
    void display_circle(string obj_name) // Declaring member function for displaying data.
    {
       cout<<"Data of Object - "<<obj_name<<" :\t";
       cout<<"Center :(x,y) = ("<<x_corr<<","<<y_corr<<")\t";
       cout<<"Radius : "<<radius<<endl;
    }
    friend void update_circle(circle &,float , float , double); // defining friend function.
    static void update_number_of_circle() // static member function for update_number_of_circle
    {
       number_of_circles += 1;
    }
};

// initializing static varible
int circle::number_of_circles = 0;

// Declaring friend function.
void update_circle(circle &object_name,float x_corrdinate,float y_corrdinate,double radi)
{
   object_name.x_corr = x_corrdinate;
   object_name.y_corr = y_corrdinate;
   object_name.radius = radi;
}

int main()
{
   circle c1,c2,c3; // Declaring  objects c1,c2,c3.
```

```
    c1 = circle(4,5,5.0);    // Assigning values to private Data members
    c2 = circle(5,9,8.0);    // using parameterized constructor for c1,c2 circle objects.
    c3 = circle(c1);         // Copying C1 to C2 using Copy Constructor.
    cout<<"\nData after initializing & CopyConstructor :-\n\n";
    c1.display_circle("c1");
    c2.display_circle("c2");         // displaying data members of class using member function.
    c3.display_circle("c3");
    update_circle(c2 ,9,-3,6.5 );    // updating object c2,c3 with new values using friend function
    update_circle(c3 ,8,7,5.89);
    cout<<"\nData after Updating using friend function :-\n\n";
    c1.display_circle("c1");
    c2.display_circle("c2");         // displaying data members of class after updating.
    c3.display_circle("c3");
    cout<<"\nNo of objects Created : "<<c2.number_of_circles<<"\n";
    return 0;
}
```

Output:



Result:

Thus, the C++ program created for the above problem is successfully executed.