**Heaven’s Light is Our Guide**



**Rajshahi University of Engineering and Technology**

**Department of Computer Science and Engineering**

**Course No:** CSE.1204

**Course Title:** Sessional based on CSE.1203 (Object Oriented Programming)

**Lab Report No:** 05

**Lab Report On:** Friend Function and Friend Class.

**Submitted By** **Submitted To**

Md. Ariful Islam Md. Asifur Rahman

Roll No: 1803046 Lecturer

Section: A Dept. of CSE,RUET

Department: CSE

**Problem No:** 01

**Problem Statement:** Implementation of **Friend Function** and **Friend Class**.

**circle**

|  |
| --- |
| float redious;  float area; |
| circle(float);  friend void print\_c(circle&);  friend void print\_ta(circle&,square&);  friend class square; |

**square**

|  |
| --- |
| float side;  float area; |
| square(float);  friend void print\_s(square&);  friend void print\_ta(circle&,square&);  void trfc(circle&,square&); |

**Theory :** A **Friend Function** is a function which is **not a member of a class** but can **access** the **private** and **protected** members of a class in which the function is declared as friend.

Similarly a **Friend Class** is a class which can **access** the **private** and **protected** members of **other class** in which it is declared as friend.

Following are some important points about friend functions and classes :

1. Friends should be used only for limited purpose. Too many functions or external classes are declared as friends of a class with protected or private data, it lessens the value of encapsulation of separate classes in object-oriented programming.
2. Friendship is not mutual. If class A is a friend of B, then B doesn’t become a friend of A automatically.
3. As a matter of Object-Oriented Programming, the concept of friends is not there in Java.

**Source Code:**

1. **main.cpp :**

|  |
| --- |
| #include <iostream>  #include "circle.h"  using namespace std;  int main()  {  circle A(5.0);  print\_c(A);  square B(10.0);  print\_s(B);  print\_ta(A,B);  B.trfc(A,B);  print\_c(A);  print\_s(B);  print\_ta(A,B);  return 0;  } |

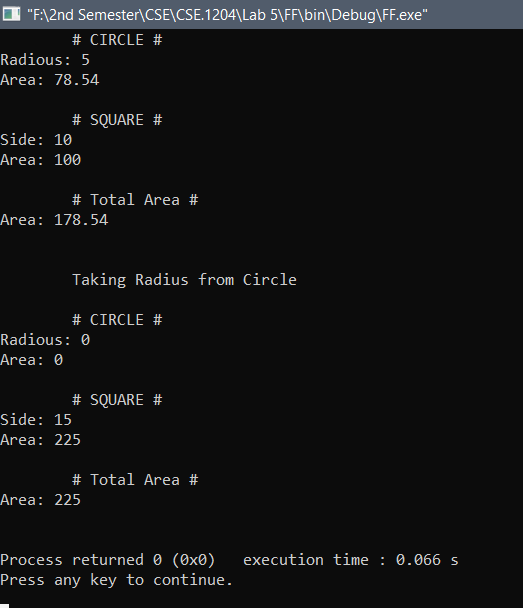
1. **circle.h :**

|  |
| --- |
| #ifndef CIRCLE\_H  #define CIRCLE\_H  class square;  class circle{  private:  float radious;  float area;  public:  circle(float);  friend void print\_c(circle&);  friend void print\_ta(circle&,square&);  friend class square;  };  class square{  private:  float side;  float area;  public:  square(float);  friend void print\_s(square&);  friend void print\_ta(circle&,square&);  void trfc(circle&,square&);  };  #endif // CIRCLE\_H |

1. **circle.cpp :**

|  |
| --- |
| #include <iostream>  #include "circle.h"  using namespace std;  circle::circle(float r){  radious=r;  }  square::square(float s){  side=s;  }  void square::trfc(circle& ob1,square& ob2){  cout<<"\n\tTaking Radius from Circle\n"<<endl;  ob2.side=ob2.side+ob1.radious;  ob1.radious=0;  }  void print\_c(circle& ob){  ob.area=3.1416\*ob.radious\*ob.radious;  cout<<"\t# CIRCLE #"<<endl;  cout<<"Radious: "<<ob.radious<<endl;  cout<<"Area: "<<ob.area<<"\n"<<endl;  }  void print\_s(square& ob){  ob.area=ob.side\*ob.side;  cout<<"\t# SQUARE #"<<endl;  cout<<"Side: "<<ob.side<<endl;  cout<<"Area: "<<ob.area<<"\n"<<endl;  }  void print\_ta(circle& ob1,square& ob2){  cout<<"\t# Total Area #"<<endl;  cout<<"Area: "<<ob1.area+ob2.area<<"\n"<<endl;  } |

**Output :**

****

**Conclusion :** By our Course Teachers help and my knowledge about C and C++, I completed the program.

**# The End #**