**\_\_\_\_\_**

In C++, two or more functions can share the same name as long as their parameter declarations are different.

• In this situation, the functions that share the same name are said to be overloaded, and the process is referred to as function overloading

Function overloading is the process of using the same name for two or more

functions.

• The secret to overloading is that each redefinition of the function must use

either-

• different types of parameters

• different number of parameters.

**Ambiguity**

void f(int x);

void f(int &x); // error

• two functions cannot be overloaded when the only difference is that one takes a

reference parameter and the other takes a normal, call-by-value parameter.

// This program contains an error.

#include <iostream>

using namespace std;

//function prototype

void f(int x);

void f(int &x); // error

int main()

{

int a=10;

f(a); // error, which f()?

return 0;

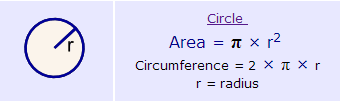
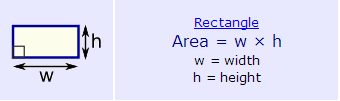
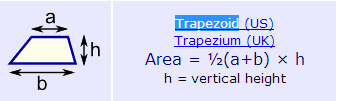
}

void f(int x)

{cout << "In f(int)\n";}

void f(int &x)

{cout << "In f(int &)\n";}

1. Write a function **AREA** to calculate the area of **circle**, **rectangle** and **Trapezoid** using function overloading and **PRINT** the result .

|  |  |
| --- | --- |
|  |  |
|  | HINT :  pi = 3.14  **void area(int);  //circle**  **void area(int,int);  //rectangle**  **void area(float ,int,int);  //trapezoid** |

const int pi=3.14;

void area(int r); //circle

void area(int,int); //rectangle

void area(float ,int,int); //trapezoid

//circle

void area(int r )

{

cout<<" circle area is : "<< pi\*r\*r;

}

//rectangle

void area(int h ,int w )

{

cout<<" rectangle area is : "<< h\*w;

}

//trapezoid

void area(float h ,int a ,int b )

{

cout<<" trapezoid area is : "<< 1/2\*(a+b)\*h;

}