NAME: SATVIK DANDALE  
ROLL NO: 13

DIV: TY E

BATCH: B1

GR NO: 1710797

**1. FRIEND FUNCTION:**

/\*

Write a CPP program to implement the concept of friend function and function overloading.

\*/

#include<bits/stdc++.h>

using namespace std;

class Demo{

private:

int \*array;

int size;

public:

Demo(){

array = NULL;

size = 0;

}

Demo(int \*array, int size){

// Allocate memory for this->array

this->array = new int [size];

for(int i=0; i < size; i++)

this->array[i] = array[i];

this->size = size;

}

void input(){

// Ask for size then take the array

cout<<"Enter the size of the array\n";

cin>>this->size;

this->array = new int [size];

cout<<"Enter the array:\n";

for(int i=0; i < size; i++){

cout<<"Enter element no. "<<i+1<<": ";

cin>>this->array[i];

}

}

void display(){

for(int i=0; i < size; i++)

cout<<this->array[i]<<"\t";

}

friend int mean(Demo&);

};

int mean(Demo& d){

// This will calculate the avg of the elements in the array of d

int sum = 0;

for(int i=0; i < d.size; i++) // We can access the private data members because it is a friend function.

sum += d.array[i];

return sum/d.size;

}

int main(){

Demo d;

d.input();

cout<<"The mean of the entered array is:\n";

cout<<mean(d);

}

OUTPUT:

Enter the size of the array

5

Enter the array:

Enter element no. 1: 2

Enter element no. 2: 3

Enter element no. 3: 5

Enter element no. 4: 4

Enter element no. 5: 1

The mean of the entered array is:

3

**2. FUNCTION OVERLOADING:**

/\*

Write a cpp program to implement the concept of function overloading.

\*/

#include<bits/stdc++.h>

using namespace std;

class Demo{

private:

float x, y;

public:

Demo(){

cout<<"Default Constructor is called.\n";

this->x = 0;

this->y = 0;

}

Demo(int x, int y){

cout<<"Constructor 1 is called\n";

this->x = x;

this->y = y;

}

Demo(float x, int y){

cout<<"Constructor 2 is called\n";

this->x = x;

this->y = y;

}

Demo(float x, float y){

cout<<"Constructor 3 is called\n";

this->x = x;

this->y = y;

}

Demo(double x, double y){

cout<<"Constructor 4 is called\n";

this->x = x;

this->y = y;

}

Demo(int x, double y){

cout<<"Constructor 5 is called\n";

this->x = x;

this->y = y;

}

void display(){

cout<<this->x<<"\t"<<this->y<<endl;

}

};

int main(){

Demo d;

float a = 1.5, b = 4.5;

Demo d1(5, 5.4); // This will be d1(int, double)

Demo d2(1, 2); // This will be int, int

Demo d3(a, b); // This will be float, float

}

OUTPUT:

Default Constructor is called.

Constructor 5 is called

Constructor 1 is called

Constructor 3 is called