```
#include "pdd.h"
#include<iostream>
#include<math.h>

PI_ctrl::PI_ctrl(float P,float I, float dT)
{
    m_P = P;
    m_I = I;
    m_integrator = 0.0f; // set inital error accumulation to 0
    m_dT = dT;
}
//calculating p value
float PI_ctrl::calc_ctrl_P(float error){
    return m_P * error;
}
//calculating I value
float PI_ctrl::calc_ctrl_I (float error){
    m_integrator += m_I * error * m_dT; // sum
    return m_integrator;
}


PID_ctrl::PID_ctrl(float P, float I, float dT,float D)
    : PI_ctrl(P,I,dT)
{
    m_d=D;
    m_differniator = 0.0f;
    m_dT = dT;
}
//calculating D value
float PID_ctrl::calc_ctrl_D (float error,float perror){
    m_differniator += m_d * (error-perror)/ m_dT;
    return m_differniator;

}
//calculating PID value by summing value of P,I and D

float PID_ctrl::calc_ctrl_PID(float error,float perror){

    return PID_ctrl::calc_ctrl_P(error) + PID_ctrl::calc_ctrl_I(error)+ PID_ctrl::calc_ctrl_D(error,perror);

}
```