```cpp
#include<vector>
#include<iostream>
#include"PID.h"
#include"FlightCtrl.h"
#include<string>

int main(int argc, char **argv) {
 float trtime,dtime;//transition and dT time declaration
 float error=0.1;
 const float m_dT=1;
    // defining states using c++ "enum" and it take values 0,1,2,3,4,5 respectively
    enum State
    {   state1,
        state2,
        state3,
        state4,
        state5
    };
std::vector<PID_ctrl> m_ctrl_configs; //declaring vector
FlightCtr ob1;
// Taking int value for state because enum stores the value in form of numbers and we have to compare state with
enums
int currentstate;
int newstate;
cout<<"enter current state as any number from 0,1,2,3,4,5"<<endl;
std::cin >>currentstate;
cout<<"enter new state as any number from 0,1,2,3,4,5"<<endl;
std::cin >>newstate;
//checking if transition is required or not
if( currentstate!= newstate)
{
     currentstate=newstate;

     if(currentstate==state1){
      float trtime=2.0;
   float dtime=20.0;
       int i=0;

       ob1.fillvector(m_ctrl_configs);
       std::cout<<ob1.printvector(m_ctrl_configs,i,error)<<std::endl;
       std::cout<<ob1.waitvector(trtime,dtime)<<std::endl;
       std::cout<<ob1.sumvector(m_ctrl_configs,i,error)<<std::endl;
       cout<<"now you are in  state1 after continous transition"<<endl;
       }
    else if(currentstate==state2){
       trtime=2;
      dtime=30;
       int i=1;
       ob1.fillvector(m_ctrl_configs);
       std::cout<<ob1.printvector(m_ctrl_configs,i,error)<<std::endl;
       std::cout<<ob1.waitvector(trtime,dtime)<<std::endl;
       std::cout<<ob1.sumvector(m_ctrl_configs,i,error)<<std::endl;
       cout<<"now you are in  state2 after continous transition";
       }
      else if(currentstate==state3){
       trtime=0.5;
       dtime=10;
       int i=2;
       ob1.fillvector(m_ctrl_configs);
       std::cout<<ob1.printvector(m_ctrl_configs,i,error)<<std::endl;
       std::cout<<ob1.waitvector(trtime, dtime)<<std::endl;
       std::cout<<ob1.sumvector(m_ctrl_configs,i,error)<<std::endl;
       cout<<"now you are in  state3 after continous transition";
       }
       else if(currentstate==state4){
       trtime=1.5;
```

```
      dtime=15;
       int i=3;
       ob1.fillvector(m_ctrl_configs);
       std::cout<<ob1.printvector(m_ctrl_configs,i,error)<<std::endl;
       std::cout<<ob1.waitvector(trtime,dtime)<<std::endl;
       std::cout<<ob1.sumvector(m_ctrl_configs,i,error)<<std::endl;
       cout<<"now you are in  state4 after continous transition";
       }
       else if(currentstate==state5){
      trtime=10;
       dtime=100;
       int i=4;
       ob1.fillvector(m_ctrl_configs);
       ob1.printvector(m_ctrl_configs,i,error);
       std::cout<<ob1.waitvector(trtime,dtime)<<std::endl;
       std::cout<<ob1.sumvector(m_ctrl_configs,i,error)<<std::endl;
       cout<<"now you are in state5 after continous transition";
       }
}
//if transition is not required then following line will execute
else{
    cout<<"you are in your current state with same PID configurations and no transition is needed"<<endl;
}

}
```