PROGRAM TO IMPLEMENT POINT, IMPLEMENT LINE SEGMENT AND TEST 5-DIMENSION OF POINT LINE CLASSIFICATION

```
#include <iostream>
using namespace std;
class Point {
   public: int x_cor,y_cor;
   void enterPointCoordinate(){
      cout<<"Enter the X-coordinate: ";
      cin>>x_cor;
      cout<<"Enter the Y-coordinate: ";
      cin>>y_cor;
   }
   void displayPoint(){
       cout<<"("<<x_cor<<" ,"<<y_cor<<")";
   }
};
class Line{
   public: template <class A, class B>
   void line(A& a, B& b){
       cout<<"For Starting Point: "<<endl;
       a.enterPointCoordinate();
       cout<<"For End Point: "<<endl;
       b.enterPointCoordinate();
       cout<<" The line segment is from: ";
       a.displayPoint();
       cout<<" to ";
       b.displayPoint();
   }
};
```

```
class PLA{
   public: template <class A, class B, class C>
   void pla5D(A& stPt, B& edPt, C& nwPt){
      cout<<"For Starting Point: "<<endl; stPt.enterPointCoordinate();
      cout<<"For End Point: "<<endl;.enterPointCoordinate();
      cout<<"For New Point: "<<endl;.enterPointCoordinate();
      if(nwPt.x_cor==stPt.x_cor && nwPt.y_cor==stPt.y_cor){
         cout<<" The new point ("<<nwPt.x_cor<<","<<nwPt.y_cor<<") is the starting point.";
      }else if(nwPt.x_cor==edPt.x_cor && nwPt.y_cor==edPt.y_cor){
         cout<<" The new point ("<<nwPt.x_cor<<","<<nwPt.y_cor<<") is the terminal point.";
      }
      else if( ((edPt.y_cor-stPt.y_cor)/(edPt.x_cor-stPt.x_cor)) != ((nwPt.y_cor-stPt.y_cor) /
      (nwPt.x_cor-stPt.x_cor)) ){
         cout<<"The given three points:";
         cout<<" ("<<stPt.x_cor<<","<<stPt.y_cor<<"), ("<<edPt.x_cor<<","<<edPt.y_cor<<")";
         cout<<"and ("<<nwPt.x_cor<<","<<nwPt.y_cor<<") does not lie in the same line.";
      }else{
         if((nwPt.x_cor>stPt.x_cor || nwPt.y_cor>stPt.y_cor) && (nwPt.x_cor<edPt.x_cor ||
        nwPt.y_cor<edPt.y_cor)){
            cout<<" The new point ("<<nwPt.x_cor<<","<<nwPt.y_cor<<")";
            cout<<" lies between in the given line segment.";
         }else if(nwPt.x_cor<stPt.x_cor || nwPt.y_cor<stPt.y_cor){
            cout<<" The new point ("<<nwPt.x_cor<<","<<nwPt.y_cor<<")";
            cout<<" is behind the given line segment.";
         }else {
            cout<<" The new point ("<<nwPt.x_cor<<","<<nwPt.y_cor<<")";
            cout<<" is beyond the given line segment.";
         }
      }
   }
};
```

```
int main() {
   int choice;
   char cont;
   cout<<"   1. Implementation of Point."<<endl;
   cout<<"   2. Implementation of Line Segment."<<endl;
   cout<<"   3. Test for 5-Dimension of PLA."<<endl;
   cout<<" Enter the choice(1/2/3): ";
   cin>>choice;
   switch(choice){
      case 1: Point p1;
            p1.enterPointCoordinate();
            cout<<" The point is: ";
            p1.displayPoint();
         break;
      case 2: Point point1, point2;
            Line line1;
            line1.line<Point, Point>(point1,point2);
         break;
      case 3: Point startPt, endPt, newPt;
            PLA pla;
            pla.pla5D<Point,Point,Point>(startPt,endPt,newPt);
         break;
      default: cout<<"Invalid choice.";
   }
   return 0;
}
```