

**Lab1: Inheritance**

```
#include<iostream.h>
class Shape
{
public:
void setWidth(int w)
{
    width =w;
}
void setHight(int h)
{
    height =h;
}
protected:
    int width;
    int height;
};

class paintCost
{
public:
    int getCost(int area)
    {      return area * 70;      }
};

class Rectangle:public Shape,public paintCost
{
public:
    int getArea()
    {      return (width * height);      }
};

int main()
{
    Rectangle Rect;
    int area;
    Rect.setHight(7);
    Rect.setWidth(5);
    area=Rect.getArea();
    clrscr();
    cout<<"Total Area      :"<<Rect.getArea();
    cout<<"\nTotal paint cost :"<<Rect.getCost(area);
}
```

**Lab 2: Polymorphism**

```
#include<iostream.h>
class Poligon
{
protected:
    int width,height;
public:
    void setValues(int a,int b)
    {
        width=a;
        height=b;
    }
};

class Rectangle:public Poligon
{
public:
    int Area()
    {
        return (width * height);
    }
};

class Triangle:public Poligon
{
public:
    int Area()
    {
        return (width * height/2);
    }
};

int main()
{
    Rectangle Rect;
    Triangle Tri;
    Poligon * poli1 = &Rect;
    Poligon * poli2 = &Tri;
    poli1->setValues(4,5);
    poli2->setValues(4,5);
    cout<<"Rectangle Area     :"<<Rect.Area();
    cout<<"\nTriangle Area      :"<<Tri.Area();
}
```

**Lab 3: Overloading**

```
#include<iostream.h>
class print
{
public:
    void print(int i) {
        cout << "Printing int: " << i << endl;
    }

    void print(double f) {
        cout << "Printing float: " << f << endl;
    }

    void print(char* c) {
        cout << "Printing character: " << c << endl;
    }
};

int main(void)
{
    print pd;

    // Call print to print integer
    pd.print(5);
    // Call print to print float
    pd.print(500.263);
    // Call print to print character
    pd.print("Hello C++");
    return 0;
}
```

**Lab 4: Overriding**

```
#include<iostream.h>
class Poligon
{
protected:
    int width,height;
public:
    void PrintName()
    {
        Cout<<"This is a Poligon";
    }
};

class Rectangle:public Poligon
{
public:
    int PrintName (int a,int b)
    {
        Cout<<"This is a Poligon of four sieds";
    }
};

class Triangle:public Poligon
{
public:
    int PrintName (int a,int b)
    {
        Cout<<"This is a Poligon of three sieds";
    }
};

int main()
{
    Rectangle Rect;
    Triangle Tri;
    cout<<Rect.PrintName ();
    cout<<Tri.PrintName ();
    cout<<Rect.PrintName (2,3);
    cout<<Tri.PrintName (2,3);
}
```