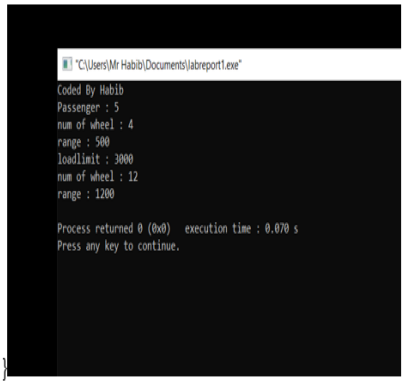
Here is other's 28 code of chapter 6 and chapter 7. Link->https://github.com/habib-e/cpp-code

1.

```
#include<bits/stdc++.h>
using namespace std;

class  vehicle

{

  int num_wheels;

  int range;

public:

  vehicle(int m,int n){num_wheels=m;range=n;}

  void show1()

  {

    cout<<"num of wheel : "<<num_wheels<<endl;

    cout<<"range : "<<range<<endl;

  }

};

class car:public vehicle

{

  int passenger;

public:

  car(int x,int y,int z):vehicle(y,z)

  {

    passenger=x;

  }

  void show()

  {

    cout<<"Passenger : "<<passenger<<endl;

    show1();

  }

};

class truck:public vehicle

{

  int loadlimit;

public:

  truck(int x,int y,int z):vehicle(y,z)

  {

    loadlimit=x;

  }

  void show()

  {

    cout<<"loadlimit : "<<loadlimit<<endl;

    show1();

  }

};

int main()

{

  car ob1(5,4,500);
```

```
  truck ob(3000,12,1200);
  ob1.show();
  ob.show();
  return 0;
```



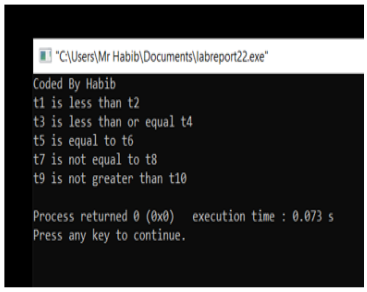

```
}
```

2.

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
class Circle
{
  int hr, min, sec;
public:
  Circle()
  {
    hr=0, min=0;
    sec=0;
  }
  Circle(int h, int m, int s)
  {
    hr=h, min=m;
    sec=s;
  }

  friend bool operator<(Circle &t1, Circle &t2);
  friend bool operator<=(Circle &t3, Circle &t4);
  friend bool operator!=(Circle &t5, Circle &t6);
  friend bool operator==(Circle &t7, Circle &t8);
  friend bool operator>(Circle &t9, Circle &t10);
};

bool operator< (Circle &t1, Circle &t2)
{
  return ( t1.hr < t2.hr && t1.min < t2.min && t1.sec < t2.sec );
}
bool operator<= (Circle &t3, Circle &t4)
{
  return ( t3.hr <= t4.hr && t3.min <= t4.min && t3.sec <= t4.sec );
}
```

```
bool operator!= (Circle &t5, Circle &t6)
{
   return ( t5.hr != t6.hr && t5.min != t6.min && t5.sec != t6.sec );
}

bool operator== (Circle &t7, Circle &t8)
{
   return ( t7.hr == t8.hr && t7.min == t8.min && t7.sec == t8.sec );
}

bool operator> (Circle &t9, Circle &t10)
{
   return ( t9.hr > t10.hr && t9.min > t10.min && t9.sec > t10.sec );
}
int main()
{
   Circle t1(3,15,45);
   Circle t2(4,15,45);
   if(t1 < t2)
      cout << "t1 is greater than t2"<<endl;
   else
      cout << "t1 is less than t2"<<endl;
   Circle t3(3,15,45);
   Circle t4(4,15,45);
   if(t3 <= t4)
      cout << "t3 is less than or equal t4"<<endl;
   else
      cout << "t3 is greater than or equal t4"<<endl;
   Circle t5(3,15,45);
   Circle t6(4,15,45);
   if(t5 != t6)
      cout << "t5 is not equal to t6"<<endl;
   else
      cout << "t5 is equal to t6"<<endl;
   Circle t7(3,15,45);
   Circle t8(4,15,45);
   if(t7 == t8)
      cout << "t7 is equal to t8"<<endl;
   else
      cout << "t7 is not equal to t8"<<endl;
   Circle t9(3,15,45);
   Circle t10(4,15,45);
   if(t9 > t10)
      cout << "t9 is greater than t10"<<endl;
   else
      cout << "t9 is not greater than t10"<<endl;
```

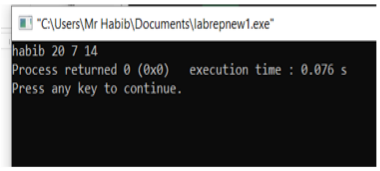
```
  return 0;
}
```



3.

```
#include<iostream>
using namespace std;
class university
{
public:
  int num_of_dept;
  int rank;
  university(int a,int b)
  {
    num_of_dept=a;
    rank=b;
  }
  virtual  void show()
  {
    cout<<num_of_dept<< endl;
    cout<<rank<<endl;
  }
};
class department: public university
{
public:
  string name;
  int student_num;
  department(string x,int y,int m,int n):university(m,n)
  {
    name=x;
    student_num=y;
  }
  void show()
  {
    cout<<name<<" ";
    cout<<student_num<<" ";
    cout<<num_of_dept<<" ";
    cout<<rank<<" ";
  }
};
int main()
```

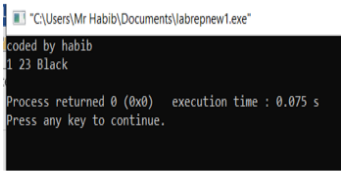
```
{
   department ob("habib", 20,7,14);
   ob.show();
   return 0;
}
```



4.

```
#include<iostream>
using namespace std;
class triangle
{
public:
   double side1,side2,side3;
   string color;
   double A,P;
   trianle()
   {
      side1=1.0;
      side2=1.0;
      side3=1.0;
   }
triangle(double a, double b,double c)
{
   side1=a;
   side2=b;
   side3=c;
}
accessor(double x, double y)
{
   A=x;
   P=y;

}

double getArea()
{
   A =0.5*side1*side2;
   return A;
}
double getPerimeter()
{
   P=side1+side2+side3;
   return P;
```

```
    }
    void show()
    {
        color="Black";
        cout<<color<<endl;
    }
};

int main()
{
    cout<<"coded by habib"<<endl;
    triangle ob(8,6,9);
    double ans=ob.getArea();
    if(ans>0)
        cout<<"1"<<endl;
    else
        cout<<"0"<<endl;
    double ans2=ob.getPerimeter();
    cout<<ans2<<endl;
    ob.show();
    return 0;
}
```



5.

```
#include<iostream>
using namespace std;

class division
{
public:
    int num_cities;
    string most_populous_city;
    division(int a, string b)
    {
        num_cities=a;
        most_populous_city=b;
    }
    virtual void show()
    {
        cout<<num_cities<<endl;
        cout<<most_populous_city<<endl;
    }
```

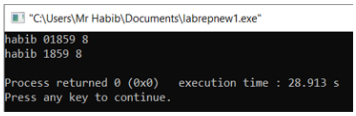
```
};
class city:public division
{
public:
  string name;
  int population;
  city(string x, int y,int n,string m):division(n,m)
  {
    name=x;
    population=y;
  }
  void show()
  {
    cout<<name<<" ";
    cout<<population<<" ";
    cout<<num_cities<<" ";
    cout<<most_populous_city<<" ";
  }
};
int main()
{
  cout<<"coded by habib"<<endl;
  city ob("dhaka", 10982, 8,"mirpur");
  ob.show();
}
```

```
"C:\Users\Mr Habib\Documents\labrepnew1.exe"
coded by habib
dhaka 10982 8 mirpur
Process returned 0 (0x0)   execution time : 0.073 s
Press any key to continue.
```

6.

```
#include<bits/stdc++.h>
using namespace std;
long long ll;
class person
{
public:
  string name;
  long long n,w_hour;
  person()
  {
    cin>>name>>n>>w_hour;
  }
  void display(){
  cout<<name<<" "<<n<<" "<<w_hour<<endl;
  }
};
class academic:virtual public person
```

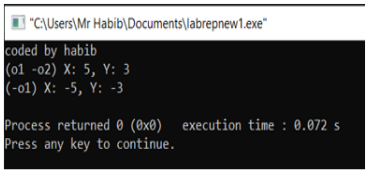
```
{
   public:

};
class non_academic: virtual public person
{
   public:


};
class supporting_stuf:public academic,public non_academic
{
   public:

};
int main()
{
   supporting_stuf ob1;
   ob1.display();
   return 0;
}
```



7.

```
# include <iostream >
using namespace std;
class coord
{
   int x, y;
public :
   coord ()
   {
      x=0;
      y=0;
   };
   coord (int i, int j)
   {
      x=i;
      y=j;
   }
   void get_xy (int &i, int &j)
   {
      i=x;
```

```
    j=y;
  }
  coord operator -( coord ob2);
  coord operator -();
};



coord coord :: operator -( coord ob2)
{
  coord temp ;
  temp .x = x - ob2 .x;
  temp .y = y - ob2 .y;
  return temp ;
}

coord coord :: operator -()
{
  x = -x;
  y = -y;
  return * this ;
}
int main ()
{
  cout<<"coded by habib"<<endl;
  coord o1 (10, 10), o2 (5, 7);
  int x, y;
  o1 = o1 - o2;
  o1. get_xy (x, y);
  cout << "(o1 -o2) X: " << x << ", Y: " << y << "\n";
  o1 = -o1;
  o1. get_xy (x, y);
  cout << "(-o1) X: " << x << ", Y: " << y << "\n";
  return 0;
}
```



8.

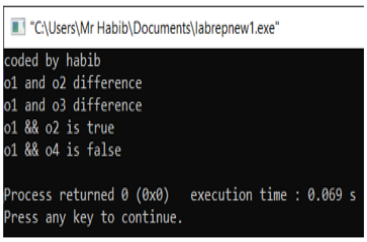
```
# include <iostream >
using namespace std;
class coord
{
  int x, y;
public :
  coord ()
```

```
  {
    x=0;
    y=0;
  };
  coord (int i, int j)
  {
    x=i;
    y=j;
  }
  void get_xy (int &i, int &j)
  {
    i=x;
    j=y;
  }
  coord operator ++() ;
};
coord coord :: operator ++()
{
  x++;
  y++;
  return * this ;
}
int main ()
{
  cout<<"coded by Habib"<<endl;
  coord o1 (10, 10);
  int x, y;
  ++ o1;
  o1.get_xy(x,y);
  cout << "(++ o1) X: " << x << ", Y: " << y << "\n";
  return 0;
}
```

9.

```
# include <iostream >
using namespace std;
class coord
{
  int x, y;
public :
  coord ()
  {
    x=0;
    y=0;
```

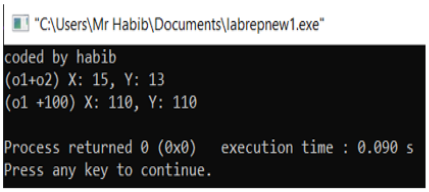
```
  };
  coord (int i, int j)
  {
    x=i;
    y=j;
  }
  void get_xy (int &i, int &j)
  {
    i=x;
    j=y;
  }
  int operator ==( coord ob2);
  int operator &&( coord ob2);
};

int coord :: operator ==( coord ob2)
{
  return x== ob2.x && y== ob2.y;
}

int coord :: operator &&( coord ob2)
{
  return (x && ob2.x) && (y && ob2.y);
}
int main ()
{
  cout<<"coded by habib"<<endl;
  coord o1 (5, 5), o2 (4, 3), o3 (9, 9), o4 (0, 0);
  if(o1 == o2)
    cout << "o1 same as o2\n";
  else
    cout << "o1 and o2 difference \n";
  if(o1 == o3)
    cout << "o1 same as o3\n";
  else
    cout << "o1 and o3 difference\n";
  if(o1 && o2)
    cout << "o1 && o2 is true \n";
  else
    cout << "o1 && o2 is false \n";
  if(o1 && o4)
    cout << "o1 && o4 is true \n";
  else
    cout << "o1 && o4 is false \n";
  return 0;
}
```



10.

```
# include <iostream >
using namespace std;
class coord
{
  int x, y;
public :
  coord ()
  {
    x=0;
    y=0;
  };
  coord (int i, int j)
  {
    x=i;
    y=j;
  }
  void get_xy (int &i, int &j)
  {
    i=x;
    j=y;
  }
  coord operator +( coord ob2);
  coord operator +( int i);
};

coord coord :: operator +( coord ob2)
{
  coord temp ;
  temp .x = x + ob2 .x;
  temp .y = y + ob2 .y;
  return temp ;
}
coord coord :: operator +( int i)
{
  coord temp ;
  temp .x = x + i;
  temp .y = y + i;
  return temp ;
}
int main ()
{
```

```
  cout<<"coded by habib"<<endl;
  coord o1 (10, 10), o2 (5, 3), o3;
  int x, y;
  o3 = o1 + o2;
  o3. get_xy (x, y);
  cout << "(o1+o2) X: " << x << ", Y: " << y << "\n";
  o3 = o1 + 100;
  o3. get_xy (x, y);
  cout << "(o1 +100) X: " << x << ", Y: " << y << "\n";
  return 0;
}
```



11.

```
#include<iostream >
using namespace std;
class coord
{
  int x, y;
public :
  coord ()
  {
    x=0;
    y=0;
  };
  coord (int i, int j)
  {
    x=i;
    y=j;
  }
  void get_xy (int &i, int &j)
  {
    i=x;
    j=y;
  }
  coord operator +( coord ob2);
  coord operator -( coord ob2);
  coord operator =( coord ob2);
};
coord coord :: operator +( coord ob2)
{
  coord temp ;
  temp .x = x + ob2 .x;
  temp .y = y + ob2 .y;
```

```
    return temp ;
}
coord coord :: operator -( coord ob2)
{
    coord temp ;
    temp .x = x - ob2 .x;
    temp .y = y - ob2 .y;
    return temp ;
}
coord coord :: operator =( coord ob2)
{
    x = ob2.x;
    y = ob2.y;
    return * this ;
}
int main ()
{
    cout<<"Coded by habib"<<endl;
    coord o1 (5, 5), o2 (4, 3), o3;
    int x, y;
    o3 = o1 + o2;
    o3. get_xy (x, y);
    cout << "(o1+o2) X: " << x << ", Y: " << y << "\n";
    o3 = o1 - o2;
    o3. get_xy (x, y);
    cout << "(o1 -o2) X: " << x << ", Y: " << y << "\n";
    o3 = o1;
    o3. get_xy (x, y);
    cout << "(o3=o1) X: " << x << ", Y: " << y << "\n";
    return 0;
}
```

```
"C:\Users\Mr Habib\Documents\labrepnew1.exe"
Coded by habib
(o1+o2) X: 9, Y: 8
(o1 -o2) X: 1, Y: 2
(o3=o1) X: 5, Y: 5

Process returned 0 (0x0)   execution time : 0.069 s
Press any key to continue.
```

12.

```
#include<bits/stdc++.h>
using namespace std;
class intake41
{
    int nofcouse;
    int intake;
    string s;
```

```
public:
  double tution_fee;
  intake41()
  {
    nofcouse=5;
    intake=0;
    tution_fee=0.0;
    s="second";
  }
  intake41(int r,double t)
  {
    intake =r;
    tution_fee=t;
  }
  void show()
  {
    cout<<"intake : "<<intake<<endl;
    cout<<"section : "<<s<<endl;
  }
  int caltuition()
  {
    return tution_fee*(nofcouse*1000);
  }
};
int main()
{
  cout<<"coded by habib"<<endl;
  int r;
  double f;
  intake41 ob(2,19.0);
  ob.show();
  cout<<ob.caltuition()<<endl;
  return 0;
}
```

Select "C:\Users\Mr Habib\Documents\labrep6.exe"

```
coded by habib
intake : 2
section :
-1822618624

Process returned 0 (0x0)   execution time : 0.077 s
Press any key to continue.
```