Distance (parts 1 & 2):

Distance.h:

```
#ifndef DISTANCE_H
#define DISTANCE_H
#include <iostream>
class Distance //English Distance class
{
   friend Distance operator-(const Distance &lhs, const Distance &rhs);
   friend std::ostream &operator<<(std::ostream &os, const Distance &obj);
   friend std::istream &operator>>(std::istream &in, Distance &obj);
   friend Distance operator+(Distance &obj, int);
   private:
       int feet;
       float inches;

   public: //constructor (no args)
       Distance() : feet(0), inches(0.0)
       {
       } //constructor (two args)
       Distance(int ft, float in) : feet(ft), inches(in)
       {
       }

       Distance operator+(const Distance &rhs) const;
       Distance &operator++(int);
       // Distance &operator=(const Distance &rhs);
       bool operator>(const Distance &rhs);
};
#endif //DISTANCE_H
```

Distance.cpp:

```
#include "Distance.h"

Distance Distance::operator+(const Distance &rhs) const{
   int newFeet {this->feet + rhs.feet};
   float newInches {this->inches + rhs.inches};
   while(newInches >= 12){
       newFeet++;
       newInches -= 12;
   }
   return Distance {newFeet, newInches};
```

```
}

Distance &Distance::operator++(int){
   this->feet++;
   return *this;
}

// Distance &Distance::operator=(const Distance &rhs){
//     this->feet = rhs.feet;
//     this->inches = rhs.inches;
//     return *this;
// }

bool Distance::operator>(const Distance &rhs){
   return (this->feet > rhs.feet) || (this->feet > rhs.feet &&
this->inches > rhs.inches);
}

Distance operator-(const Distance &lhs, const Distance &rhs){
   int newFeet {lhs.feet - rhs.feet};
   float newInches {lhs.inches - rhs.inches};
   while(newInches < 0){
       newFeet--;
       newInches += 12;
   }
   return Distance {newFeet, newInches};
}

std::ostream &operator<<(std::ostream &os, const Distance &obj){
   os << "{Feet: " << obj.feet << ", " << "Inches: " << obj.inches << "}";
   return os;
}

std::istream &operator>>(std::istream &in, Distance &obj){
   std::cout << "Enter Feet: ";
   in >> obj.feet;
   std::cout << "Enter Inches: ";
   in >> obj.inches;
   return in;
}
```

```
Distance operator+(Distance &obj, int y){
   Distance d;
   d.feet = obj.feet + y;
   d.inches = obj.inches;
   return d;
}
```

Main.cpp:

```
#include <iostream>
#include "Distance.cpp"

int main(){
   Distance dist1, dist3, dist4; //define distances
   std::cin >> dist1;
   Distance dist2(11, 6.25); //define, initialize dist2
   dist3 = dist1 + dist2;    //single '+' operator
   dist4 = dist1 - dist2;    //friend '-' operators
   //display all lengths
   std::cout << "dist1 = ";
   std::cout << dist1 << std::endl;
   std::cout << "dist2 = ";
   std::cout << dist2 << std::endl;
   std::cout << "dist3 = ";
   std::cout << dist3 << std::endl;
   std::cout << "dist4 = ";
   std::cout << dist4 << std::endl;

   dist2 = dist1++;
   dist3 = dist2 + 10;
   std::cout << "dist2 = ";
   std::cout << dist2 << std::endl;
   std::cout << "dist3 = ";
   std::cout << dist3 << std::endl;
   if(dist4 > dist1){
       std::cout<< "Yes" << std::endl;
   }
   else{
       std::cout << "No" << std::endl;
   }
   return 0;
```

```
}
```

Runtime output:
Enter Feet: 13
Enter Inches: 8
dist1 = {Feet: 13, Inches: 8}
dist2 = {Feet: 11, Inches: 6.25}
dist3 = {Feet: 25, Inches: 2.25}
dist4 = {Feet: 2, Inches: 1.75}
dist2 = {Feet: 14, Inches: 8}
dist3 = {Feet: 24, Inches: 8}
No

Time part 3:

Time12.h:

```
#ifndef TIME_12_H
#define TIME_12_H
#include <iostream>
using namespace std;

class time12
{
private:
   bool pm;  //true = pm, false = am
   int hrs;  //1 to 12
   int mins; //0 to 59
public:       //no-arg constructor
   time12() : pm(true), hrs(0), mins(0)
   {
   }
   //3-arg constructor
   time12(bool ap, int h, int m) : pm(ap), hrs(h), mins(m)
   {
   }
   void display() const //format: 11:59 p.m.
   {
       cout << hrs << ':';
       if (mins < 10)
           cout << '0'; //extra zero for "01"
       cout << mins << ' ';
       string am_pm = pm ? "p.m." : "a.m.";
```

```
        cout << am_pm;
    }

};

#endif
```

Time24.h:

```
#ifndef TIME_24_H
#define TIME_24_H
#include <iostream>
#include "time12.h"
using namespace std;

class time24
{
private:
    int hours;   //0 to 23
    int minutes; //0 to 59
    int seconds; //0 to 59
public:          //no-arg constructor
    time24() : hours(0), minutes(0), seconds(0)
    {
    }
    time24(int h, int m, int s) : //3-arg constructor
                                  hours(h), minutes(m), seconds(s)
    {
    }
    void display() const //format: 23:15:01
    {
        if (hours < 10)
            cout << '0';
        cout << hours << ':';
        if (minutes < 10)
            cout << '0';
        cout << minutes << ':';
        if (seconds < 10)
            cout << '0';
        cout << seconds;
    }
```

```
    operator time12(){
        int hrs {this->hours % 12};
        if (hrs == 0)
            hrs = 12;
        int mins {this->minutes};
        bool pm {false};
        if (this->hours >= 12)
            pm = true;
        return time12 {pm, hrs, mins};
    }
};
#endif //TIME_24_H
```

Main.cpp:

```
#include "time12.h"
#include "time24.h"
#include <iostream>
using namespace std;

int main(){
    int h, m, s;

    while (true){ //get 24-hr time from user
        cout << "Enter 24-hour time: \n";
        cout << " Hours (0 to 23): ";
        cin >> h;
        if (h > 23) //quit if hours > 23
            return (1);
        cout << " Minutes: ";
        cin >> m;
        cout << " Seconds: ";
        cin >> s;

        time24 t24(h, m, s);     //make a time24
        cout << "You entered: "; //display the time24
        t24.display();

        time12 t12 = t24; //convert time24 to time12

        cout << "\n12-hour time: "; //display equivalent time12
```

```
        t12.display();
        cout << "\n\n";
    }
    return 0;
}
```

Runtime output:
Enter 24-hour time:
 Hours (0 to 23): 12
 Minutes: 00
 Seconds: 00
You entered: 12:00:00
12-hour time: 12:00 p.m.

Enter 24-hour time:
 Hours (0 to 23): 23
 Minutes: 59
 Seconds: 00
You entered: 23:59:00
12-hour time: 11:59 p.m.

Enter 24-hour time:
 Hours (0 to 23): 1
 Minutes: 00
 Seconds: 00
You entered: 01:00:00
12-hour time: 1:00 a.m.

Enter 24-hour time:
 Hours (0 to 23): 24