A.

```
//
//  create on 04/13/2015
//  version 1.01
//  revision history
//  update Apr.18.2015: Add a bool function createNewFile
//
//

#ifndef yz523_Header_h
#define yz523_Header_h

#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
using namespace std;

class yz523_Library{
public:

//this function displays a prompt on the output console and returns a
string.
string promptForString(string prompt);

//this function check if the file already exists, if exists ask user
to cancel or overwrite it. Return false if cancels, otherwise return
true.
bool createNewFile(ofstream& out, string filename);

};

static string promptForString(string prompt){ //the string promt here
is the actual information display on the output console
    cout << prompt; //display a prompt
    return prompt; //return a string
}


static bool createNewFile(ofstream &out, string filename){
    out.open(filename);
    if(out.is_open()){
        char r;
        cout << "The file already exists, Do you want to overwrite it?
(y/n)";
        cin >> r;
        if(r=='y'){
            out.open(filename,ios::out);
            out.close();
```

```
            return true; //overwrite the file and return true
        }
        else{
            return false; //cancel the action and return false
        }
    }
    else{
        out.open(filename,ios::out);
        out.close();
        return true; //create a new file and return true
    }
}
#endif
```

B.

```
//
//  Header.h
//  HW3
//
//  Created by Benny on 4/18/15.
//  Copyright (c) 2015 Benny. All rights reserved.
//

#ifndef TIMEB_Header_h
#define TIMEB_Header_h

#include<iostream>
using namespace std;

class Time
{
public:
    Time& operator++(int unused);
private:
    int seconds_;
};

Time& Time::operator++(){
    Time clone(seconds_,1)
    seconds_+=1;
    return clone;
}
#endif
```

C.
Header file:

```
ostream& operator<<(ostream &os, Socialite &s);
```

Implement file:

```
ostream& operator<<(ostream &os, Socialite &s){
    s.out(os);
    return os;
}
```

D.
Header file:

```
ofstream& operator<<(ofstream &of, Socialite &s);
```

Implement file:

```
ofstream& operator<<(ofstream &of, Socialite &s){
    s.HTML_Output(of);
    return of;
}
```

E.

```
#ifndef TIMEE_Header_h
#define TIMEE_Header_h

#include<iostream>
using namespace std;

class Time
{
public:
    Time& operator++();
    Time& operator--();
private:
    int hour_;
};

Time& Time::operator++(){
    hour_+=1;
    return *this;
}

Time& Time::operator--(){
    hour_-=1;
    return *this;
}
#endif
```