Objective:

- Introducing flexibility/usability in the CString class by adding defining different operators for it.

Task-1:

Implement your own String class. It should not be difficult for you, as you have implemented most of the functions as global functions in your PF class. The idiom "Use const wherever possible" is also applied on the member functions.

-> Adding assignment operator
```
CString & operator = ( const CString & ref )
```

➔ Replacing
```
char & at( const int index);
```
With
```
char & operator []( const int index);
```

➔ Replacing
```
const char & at( const int index) const;
```
With
```
const char & operator []( const int index) const;
```

➔ Replacing
```
void concatEqual( const CString& s2 );
```
With
```
void operator +=( const CString& s2 );
```

➔ Replacing
```
void concatEqual( const char *const s2 );
```
With
```
void operator +=( const CString& s2 );
```

➔ Replacing
```
int isEqual(const CString & s2 )const  ;
```
With
```
int operator ==( const CString& s2 );
```

➔ Replacing
```
int isEqual(const char *const s2 )  ;
```
With
```
int operator ==(const char *const s2 )  ;
```

➔ Replacing
```
CString concat( const CString& s2 ) const  ;
```
With
```
CString operator + ( const CString& s2 ) const  ;
```

➔ Replacing
```
CString concat( const char * const s2 )  const;
```
With
```
CString operator +( const char * const s2 )  const;
```

Will do the following in next lecture

➔ Replacing
```
CString tokenzie( const char * const delim );
```
With
```
CString operator () ( const char * const delim );
```

➔ Replacing

```
        void input();
```
With
```
        istream & operator >> (istream &, const CString &);
```

→   Replacing
```
        void display()const;
```
With
```
        ostream & operator << (ostream &, const CString &);
```

–>   Adding type cast operators
```
        operator int()
```

Task-2:
Although it is not necessary to write copy constructor and assignment operator for every class but for practice it is advised to write assignment and copy constructors for all the classes involved in Scheduler Application.