

OLQ8

Practice and Review

OLQ8 is over `this`, `friends` and operator overloading.

Can you explain what `this` represents?

Can you explain why we would make a function a `friend` instead of a member function? What makes a `friend` function different from a regular non member function? How do you make a function a `friend`?

Given a class containing an operator overload function as a member function, can you make the necessary changes to move that operator overload function out of the class? The first piece of code here shows the operator overload function as a member function. The second section of code show the same operator overload function but OUTSIDE of the class, with different parameters and declared a `friend`.

Overloaded < defined as a member function with one parameter

```
class Quarterback
{
    public :
        Quarterback(std::string name, int att, int comp, int yds, int td)
        : qbName{name}, qbAtt{att}, qbComp{comp}, qbYds{yds}, qbTd{td}
        {
        }

        bool operator<(const Quarterback& QB)
        {
            if (qbAtt < QB.qbAtt &&
                qbComp < QB.qbComp &&
                qbYds < QB.qbYds &&
                qbTd < QB.qbTd)
                return true;
            else
                return false;
        }

    private :
        std::string qbName;
        int qbAtt;
        int qbComp;
        int qbYds;
        int qbTd;
};
```

Overloaded < defined a non member friend function with one parameter

```
class Quarterback
{
    friend bool operator<(const Quarterback& QB1, const Quarterback& QB2);

public :
    Quarterback(std::string name, int att, int comp, int yds, int td)
    : qbName{name}, qbAtt{att}, qbComp{comp}, qbYds{yds}, qbTd{td}
    {
    }

private :
    std::string qbName;
    int qbAtt;
    int qbComp;
    int qbYds;
    int qbTd;
};

bool operator<(const Quarterback& QB1, const Quarterback& QB2)
{
    if (QB1.qbAtt < QB2.qbAtt  &&
        QB1.qbComp < QB2.qbComp &&
        QB1.qbYds  < QB2.qbYds  &&
        QB1.qbTd   < QB2.qbTd)
        return true;
    else
        return false;
}
```