

---

Subject: Re: again... use if deleted function :?  
Posted by [Oblivion](#) on Thu, 18 Jan 2018 10:26:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quote:

```
// constructor
QTFStr() = default;
```

it shouldn't be a solution?

Matteo

Hello Matteo,

There is a problem here:

```
QTFStr Tensoflessione::Instabilitaqtf(QTFStr &qtf)
{
    // -----
    // ^
    // You are passing a reference (which can be modified, as you do below.

String risultato;
<= 1 &";
<= 1 ";
risultato << GetVerificalnst();

qtf.StartTable(3,2,2,10).CharSize(12);
qtf.TableSubTitle("Instabilità flesso-torsionale");
qtf.LeftCellMargin(75)("[l, eff]").LeftCellMargin(15).AlignRight()
(Format("%.2f", GetInstab().leff)).AlignLeft()("mm")("Lunghezza libera di inflessione");

(Format("%.2f", GetInstab().sigmam_critico)).AlignLeft()("")("Tensione critica di svergolamento");

(Format("%.2f", GetInstab().lambda_relm)).AlignLeft()("")("Snellezza relativa di svergolamento");
qtf.LeftCellMargin(75)("[k, crit]").LeftCellMargin(15).AlignRight()
(Format("%.2f", GetInstab().kcrit)).AlignLeft()("N/mm[<sup>2</sup>]")("Coefficiente di sbandamento
laterale");
qtf.TableSubTitle(risultato);
qtf.EndTable();

qtf.TableSubTitle(risultato);
qtf.EndTable();

return qtf;
```

```
//-----
// ^
// Then, compiler attempts (by default) to copy qtf (But you need to explicitly define a copy
constructor, which is missing, and in it, explicitly copy the members of your class (using clone() ),
They are causing the error.)
// By the way, returning a QTFStr from this function shouldn't be necessary at all, since you
are already modifying the referenced instance! (it is non-const), unless you really need a copy of
it. It is a much cheaper solution.
}
```

OR, if you don't use the referenced variable qtf after calling the method but you need your method to return a QTFStr, you can move it instead.

[

```
QTFStr(QTFStr&&) = default;
QTFStr& operator=(QTFStr&&) = default;
```

Then,

```
return pick(qtf);
```

And please read about copy and move semantics of both C++11 (StackOverFlow has plenty of good articles on it), and UPP.

Best regards,  
Oblivion

---