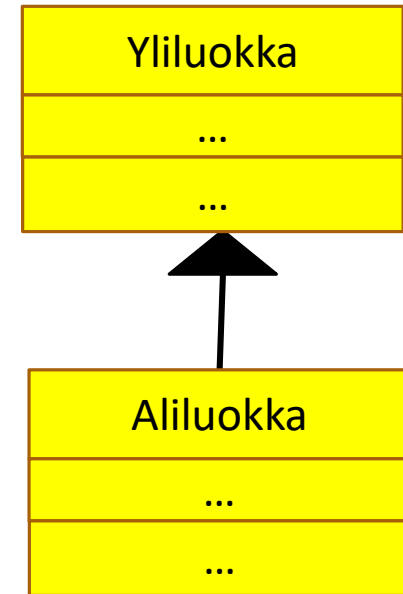


## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Yksityisessä periytymisessä lapsiluokan oliota **ei voida** käyttää kuin se olisi kantaluokan olio.
- Yksityisessä periytymisessä kantaluokan **julkiset** ja **suojatut** jäsenet ovat lapsiluokan yksityisiä jäseniä. Lapsiluokka voi käyttää omassa toteutuksessa kantaluokan julkisia sekä suojattuja jäseniä.
- Jos yliluokan kaikkia tietoja tai toimintoja ei haluta antaa aliluokan käyttöön, niin täytyy käyttää yksityistä perintää julkisen perinnän sijaan.
- Yksityisen perinnän suurin vaikutus on se, että yliluokan jäsenet näkyvät aliluokassa aivan kuin ne määritelty aliluokkaan luokan **private** osaan, riippumatta siitä, miten ne on suojattu kantaluokassa.
- Luo uusi projekti nimeltä **YksityinenPerinta (Non-Qt Project->Plain C++ Application)**
- Lisää projektiin luokat **Yliluokka** ja **Aliluokka**



## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Avaa yliluokan määrittely (yliluokka.h) kirjoita siitä alla olevan mukainen

```
#include <iostream>
using namespace std;
```

```
class Yliluokka
{
public:
    Yliluokka();
    ~Yliluokka();
    void yliluokkaPublicFunktio();

protected:
    void yliluokkaProtectedFunktio();

private:
    void yliluokkaPrivateFunktio();
};
```

## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Avaa aliluokan määrittely (aliluokka.h) kirjoita siitä alla olevan mukainen

```
#include "yliluokka.h"
```

```
class Aliluokka : private Yliluokka // yksityinen periminen  
{  
public:  
    Aliluokka();  
    ~Aliluokka();  
};
```

## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Lisää yliluokan jäsenfunktioiden toteutus alla olevan mukaisesti yliluokan toteutukseen (yliluokka.cpp)

```
#include "yliluokka.h"
```

```
Yliluokka::Yliluokka()  
{  
    cout << "Yliluokka muodostin" << endl;  
}
```

```
Yliluokka::~Yliluokka()  
{  
    cout << "Yliluokka tuhoaja" << endl;  
}
```

```
void Yliluokka::yliluokkaPublicFunktio()  
{  
    cout << "yliluokkaPublicFunktio()" << endl;  
}
```

```
void Yliluokka::yliluokkaProtectedFunktio()  
{  
    cout << "yliluokkaProtectedFunktio()" << endl;  
}
```

```
void Yliluokka::yliluokkaPrivateFunktio()  
{  
    cout << "yliluokkaPrivateFunktio()" << endl;  
}
```

## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Lisää aliluokan jäsenfunktioden toteutus alla olevan mukaisesti aliluokan toteutukseen (aliluokka.cpp)

```
#include "aliluokka.h"
```

```
Aliluokka::Aliluokka()  
{  
    cout << "Aliluokka muodostinfunktio" << endl;  
}
```

```
Aliluokka::~~Aliluokka()  
{  
    cout << "Aliluokka tuhoajafunktio" << endl;  
}
```

## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Lisää **main()** –funktioon alla oleva koodi

```
#include "aliluokka.h"
```

```
int main()  
{  
    Aliluokka olioAliluokka;  
  
    return 0;  
}
```

- Suorita build ja aja ohjelmaa.
- Seuraavalla sivulla yritetään aliluokan oliolla kutsua ylliluokan jäsenfunktioita.

## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Lisää **main()** funktioon olion luonnin jälkeen rivi **olioAliluokka.yliluokkaPublicFunktio();**
- Nyt kun yrität kääntää koodia, niin kääntäjä antaa virheilmoituksen  
... **error: 'void Yliluokka::yliluokkaPublicFunktio()' is inaccessible ...**
- Tämä johtuu siitä, että yksityisessä periytymisessä kantaluokan **julkiset** ja **suojatut** jäsenet ovat lapsiluokan yksityisiä jäseniä.
- ***Eli aliluokan olio ei voi suoraan kutsua yliluokan jäsenfunktioita yksityisessä perinnässä!***
- Lisää aliluokan muodostinfunktioon rivi **yliluokkaPublicFunktio();** ,suorita build ja aja ohjelmaa.
- Nyt yliluokan jäsenfunktion kutsuminen onnistuu.
- Lisää aliluokan muodostinfunktioon rivi **yliluokkaProtectedFunktio();** ,suorita build ja aja ohjelmaa.
- Protected osassakin olevan jäsenfunktion kutsuminen onnistuu, mutta jos lisäät vielä aliluokan muodostinfunktioon rivin **yliluokkaPrivateFunktio();** ,niin siitä kääntäjä antaa virheilmoituksen. Tämä siksi, koska **private** osan jäseniin ei pääse käsiksi.

## IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

- Yleinen nyrkkisääntö on, että olio-ohjelmoinnissa käytetään luokkien välillä julkista (public) perintää.
- Joissakin erikoistapauksissa yksityistä perintää voi käyttää, mutta silloin täytyy erityisesti olla tiedossa se, että yliluokan kaikkia tietoja tai toimintoja ei haluta antaa aliluokan käyttöön.